SSE/14789/IFS/0002: CMS Integration Services

The Integration Services interface for Cicero LMS is designed to be used by the end-user interfaces of a library, for users to be able to check up on loans, bills and more. A prerequisite for using this interface is that the client application must be user-faced, and have an interactive flow with the user that only fetches information from the interface when it is needed.

This interface must not be used for other purposes than what is described here. Examples of detrimental usage:

- Fetching large amounts of information, e.g. to harvest information from patrons or libraries
- Fetching patron information when the patron is not using the system
- Fetching library information when it is not used in conjunction with the user using the system
- Other activities which do not adhere to the described purpose

Detrimental usage of the CMS interface can lead to the client application getting its access to the interface closed down.

It is generally allowed to have a local cache with information. Updating the local cache must however be part of the normal user scenarios of the system - information harvesting, which for a shorter or a longer period of time makes many requests to the CMS interface, will not be tolerated.

Some operations are suited to having information cached across patrons in the application. This information is limited to:

- Configurations
- LoanerGroups
- Operations for fetching the placements of the library (Branches, Departments, Locations and Sublocations)

Show/Hide List Operations Expand Operations Raw

/external/{agencyid}/catalog/availability/v3 GET

Get availability of bibliographical records.

Implementation Notes

Returns an array of availability for each bibliographical record.

Parameters

Parameter	Description	Parameter Type	Data Type		
agencyid	ISIL of the agency (e.g. DK-761500)	path	string		
recordid	list of record ids	query	array[string]		
exclude	Identifies the branchIds which are excluded from the result	query	array[string]		
Response Class					
Model Model Schem	ia				
AvailabilityV3 {					

recordId (string): The FAUST number of the Bibliographic record, reservations (integer): Total number of current active reservations of the Bibliographic record,

reservable (boolean): True if materials can be reserved,

available (boolean): True if materials is available on-shelf at some placement, false if all materials are lent out

}

Response Content Type application/json 🗸

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

GET /external/{agencyid}/catalog/bookingInformation/{recordid}/v3

Retrieves all booking information for each branch of an agency for bookings that ends after the current date.

Implementation Notes

Returns an array of BookingBranchInfo which contains:

- the branch ID
- gross number of available materials for that branch
- array of BookingInfo for the given bibliographic record, containing the booking Period and the number of preferred materials

This is to highlight when materials are available for a new potential booking.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
recordid	identifies the bibliographical record, i.e. the FAUST number	path	string

Response Class

```
Model Model Schema
```

BookingBranchInfoV3 {

branchId (string): The branch ID, **grossNumberAvailable** (integer): The gross number of available materials, **bookingInfo** (array[BookingInfo]): Details about requested booking information

}

BookingInfo {

preferredMaterials (integer): The preferred number of materials, **period** (Period): The booking period information containing the start and the end date

}

Period {

```
from (string, optional): Open-ended if not set,
to (string, optional): Open-ended if not set
```

}

Response Content Type application/json V			
Response Message	25		
HTTP Status Code	Reason	Response Model	
400	bad request com.dantek.dl.rest.Re	stException	
401	client unauthorized		
404	patron not found		

GET	/external/{agencyid}/catalog/holdings/v3	Get placement holdings for bibliographical records (DEPRECATED).
-----	--	--

Implementation Notes

This method has been deprecated. Consider using /external/{agencyid}/catalog/holdings/v5 instead Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available on-shelf or lent out.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
recordid	Identifies the bibliographical records - The FAUST number.	query	array[string]
exclude	Identifies the branchIds which are excluded from the result	query	array[string]

Response Class

Model Model Schema

HoldingsForBibliographicalRecordV3 {

recordId (string): Identifies the bibliographical record for the available materials, The FAUST number,

reservations (integer): Total number of current active reservations for the bibliographical record,

reservable (boolean): True if there is any reservable materials,

holdings (array[HoldingsV3]): An array of holdings for the materials matching the bibliographical record, as distributed across branches, departments and locations

5/5/25, 1:11 PM

Swagger UI

HoldingsV3 {

materials (array[MaterialV3]): Materials that belongs to this placement, location (AgencyLocation, optional): Placement location, sublocation (AgencySublocation, optional): Placement sublocation, department (AgencyDepartment, optional): Placement department, branch (AgencyBranch): Placement branch

}

}

MaterialV3 {

itemNumber (string): Identifies the material,

materialGroup (MaterialGroup): Name of the material group that the material belongs to,

periodical (Periodical, *optional*): Present if material is a periodical,

available (boolean): True if material is available on-shelf, false if lent out

}

MaterialGroup {

name (string): Name of the material group, description (string, optional): Description of the material group

}

Periodical {

volume (string, optional),
volumeYear (string, optional),
displayText (string): A representation of the periodica volume information that is suitable for display,
volumeNumber (string, optional)

}

AgencyLocation {

locationId (string): Location identifier, **title** (string): Name of the location

}

AgencySublocation {

title (string): Name of the sub-location, **sublocationId** (string): Sub-location identifier

}

AgencyDepartment {

departmentId (string): Department identifier, **title** (string): Name of the department

<pre>} AgencyBranch { branchId (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch }</pre>				
Response Content Ty	pe application/json 🗸			
Response Messages	5			
HTTP Status Code	Reason	Response Model		
400	bad request			
401	client unauthorized			

GET	/external/{agencyid}/catalog/holdings/v4

Implementation Notes

Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available on-shelf or lent out.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
recordid	Identifies the bibliographical records - The FAUST number.	query	array[string]
exclude	Identifies the branchIds which are excluded from the result	query	array[string]

Response Class

Model Model Schema

HoldingsForBibliographicalRecordV4 {

recordId (string): Identifies the bibliographical record for the available materials, The FAUST number,

Get placement holdings for bibliographical records.

reservations (integer): Total number of current active reservations for the bibliographical record,

reservable (boolean): True if there is any reservable materials,

holdings (array[HoldingsV4]): An array of holdings for the materials matching the bibliographical record, as distributed across branches, departments and locations

}

HoldingsV4 {

materials (array[MaterialV3]): Materials that belongs to this placement,

location (AgencyLocation, optional): Placement location,

sublocation (AgencySublocation, optional): Placement sublocation,

section (AgencySection, optional): Placement section,

department (AgencyDepartment, optional): Placement department,

branch (AgencyBranch): Placement branch

}

MaterialV3 {

itemNumber (string): Identifies the material,

materialGroup (MaterialGroup): Name of the material group that the material belongs to,

periodical (Periodical, optional): Present if material is a periodical,

available (boolean): True if material is available on-shelf, false if lent out

}

MaterialGroup {

name (string): Name of the material group, description (string, optional): Description of the material group

}

Periodical {

volume (string, optional),

volumeYear (string, optional),

displayText (string): A representation of the periodica volume information that is suitable for display, **volumeNumber** (string, *optional*)

}

AgencyLocation {

locationId (string): Location identifier, **title** (string): Name of the location

}

AgencySublocation {

title (string): Name of the sub-location,

AgencySection { section identifier, title (string): Section identifier, title (string): Name of the section AgencyDepartment{ departmentId (string): Department identifier, title (string): Name of the department AgencyBranch { branchId (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch Response Content Type [application/json > Response Messages HTTP Status Code Reason Response Model 400 bad request denumber of the department of the departm	sublocationId (str	ing): Sub-location identifier	
sectionId (string): Section identifier, title (string): Name of the section AgencyDepartment { departmentId (string): Department identifier, title (string): Name of the department AgencyBranch { branchId (string): ISIL of branch (e.g. DK-761501), title (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch Response Content Type [application/json Response Messages HTTP Status Code Reason Response Model 400 bad request 401 client unauthorized GET /external/{agencyid}/catalog/holdings/v5 Get placement holdings for bibliographical records Implementation Notes Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available	}		
title (string): Name of the section AgencyDepartment { departmentId (string): Department identifier, title (string): Name of the department AgencyBranch { branchId (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch gesponse Content Type application/json v Response Messages HTTP Status Code Reason Response Messages HTTP Status Code Reason deat deat <	AgencySection {		
AgencyDepartment { departmentId (string): Department identifier, title (string): Name of the department AgencyBranch { branchId (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch esponse Oontent Type application/json ~ Response Messages HTTP Status Code Reason Response Messages HTTP Status Code Response Messages HTTP Status Code Reason Response Messages HTTP Status Code Response Messages HTTP Status Code Response Messages Response Messages HTTP Status Code Response Messages HTTP Status Code Response Messages Response Messages <t< th=""><th>sectionId (string):</th><th>Section identifier,</th><th></th></t<>	sectionId (string):	Section identifier,	
AgencyDepartment { departmentId (string): Department identifier, title (string): Name of the department AgencyBranch { branchId (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch (e.g. DK-761501), title (string): Name of branch Response Content Type application/json ~ Response Messages: HTTP Status Code Response Messages: HTTP Status Code de0 bad request Get /external/{agencyid}/catalog/holdings/v5 Get placement holdings for bibliographical records. The holdings lists the materials on each placement, and whether they are available	title (string): Name	e of the section	
departmentId (string): Department identifier, title (string): Name of the department AgencyBranch { branchId (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch esponse Content Type [application/json v] Response Messages HTTP Status Code Response Messages HTTP Status Code keason de0 bad request de1 client unauthorized Get placement holdings for bibliographical records. The holdings lists the materials on each placement, and whether they are available	}		
title (string): Name of the department AgencyBranch { branchId (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch Response Content Type application/json ~ Response Messages HTTP Status Code Reason Response Messages HTTP Status Code Request 401 client unauthorized			
AgencyBranch { branchId (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch Response Content Type application/json v Response Messages HTTP Status Code Reason Response Model 400 bad request 401 client unauthorized GET /external/{agencyid}/catalog/holdings/v5 GET /external/{agencyid}/catalog/holdings/v5 Cet placement holdings for bibliographical record. The holdings lists the materials on each placement, and whether they are available	-		
AgencyBranch Id (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch Response Content Type [application/json v] Response Messages: HTTP Status Code Reason AgencyBranch Id (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch AgencyBranch Id (string): Name of branch Response Messages: HTTP Status Code Reason AgencyBranch Id (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch Response Message: HTTP Status Code Reason AgencyBranch Id (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch (e.g. D	title (string): Name	e of the department	
branchId (string): ISIL of branch (e.g. DK-761501), title (string): Name of branch Response Content Type application/json v Response Messages HTTP Status Code Reason Response Messages Unterpotentiation Votes Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available	}		
title (string): Name of branch Response Content Type [application/json v] Response Messages HTTP Status Code Reason v A00 bad request 400 client unauthorized GET /external/{agencyid}/catalog/holdings/v5 Get placement holdings for bibliographical records Implementation Notes Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available			
Response Content Type application/json Response Messages HTTP Status Code Reason A00 bad request 400 client unauthorized	-	_	
Response Content Type application/json Response Messages HTTP Status Code Reason 400 bad request 401 client unauthorized GET /external/{agencyid}/catalog/holdings/v5 Get placement holdings for bibliographical records		e of branch	
Response Messages HTTP Status Code Reason 400 bad request 401 client unauthorized	}		
400 bad request 401 client unauthorized			Response Model
401 client unauthorized GET /external/{agencyid}/catalog/holdings/v5 Get placement holdings for bibliographical records Implementation Notes Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available			
GET /external/{agencyid}/catalog/holdings/v5 Get placement holdings for bibliographical records Implementation Notes Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available		·	
Implementation Notes Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available	401	client unauthorized	
Implementation Notes Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available			
Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available	GET /external	'{agencyid}/catalog/holdings/v5	Get placement holdings for bibliographical records.
Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available			
Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available	Implementation	Notes	
	· · · · · · · · · · · · · · · · · · ·		The holdings lists the materials on each placement, and whether they are available
		0	

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string

5/5/25, 1:1	1 PM		Swagger UI	
	Parameter	Description	Parameter Type	Data Type
	recordid	Identifies the bibliographical records - The FAUST number.	query	array[string]
	exclude	Identifies the branchIds which are excluded from the result	query	array[string]

Response Class

Model Model Schema

HoldingsForBibliographicalRecordV5 {

recordId (string): Identifies the bibliographical record for the available materials, The FAUST number,

reservations (integer): Total number of current active reservations for the bibliographical record,

reservable (boolean): True if there is any reservable materials,

holdings (array[HoldingsV5]): An array of holdings for the materials matching the bibliographical record, as distributed across placements

}

HoldingsV5 {

materials (array[MaterialV4]): Materials that belongs to this placement,

location (AgencyLocation, optional): Placement location,

sublocation (AgencySublocation, optional): Placement sublocation,

section (AgencySection, optional): Placement section,

department (AgencyDepartment, optional): Placement department,

branch (AgencyBranch): Placement branch

}

MaterialV4 {

itemNumber (string): Identifies the material,

returnDate (string): Return date of the material if it is lendout. Format: yyyy-mm-dd,

materialGroup (MaterialGroup): Name of the material group that the material belongs to,

periodical (Periodical, optional): Present if material is a periodical,

available (boolean): True if material is available on-shelf, false if lent out

}

MaterialGroup {

name (string): Name of the material group,description (string, *optional*): Description of the material group

}

Periodical {

volume (string, optional),
volumeYear (string, optional),
displayText (string): A representation of the periodica volume information that is suitable for display,
volumeNumber (string, optional)

}

AgencyLocation {

locationId (string): Location identifier, **title** (string): Name of the location

}

AgencySublocation {

title (string): Name of the sub-location, **sublocationId** (string): Sub-location identifier

}

AgencySection {

sectionId (string): Section identifier, title (string): Name of the section

}

AgencyDepartment {

departmentId (string): Department identifier, **title** (string): Name of the department

}

AgencyBranch {

branchId (string): ISIL of branch (e.g. DK-761501), **title** (string): Name of branch

}

Response Content Type application/json

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

GET /external/{agencyid}/catalog/holdingsLogistics/v1

Implementation Notes

Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available on-shelf or lent out.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
recordid	Identifies the bibliographical records - The FAUST number.	query	array[string]
exclude	Identifies the branchIds which are excluded from the result	query	array[string]

Response Class

Model Model Schema

HoldingsForBibliographicalRecordLogisticsV1 {

recordId (string): Identifies the bibliographical record for the available materials, The FAUST number,

reservations (integer): Total number of current active reservations for the bibliographical record,

reservable (boolean): True if there is any reservable materials,

holdings (array[HoldingsLogisticsV1]): An array of holdings for the materials matching the bibliographical record, as distributed across branches, departments and locations

}

HoldingsLogisticsV1 {

materials (array[MaterialV3]): Materials that belongs to this placement,

ImsPlacement (PlacementV1, optional): LMS placement,

branch (AgencyBranch): Placement branch,

logisticsPlacement (array[string], optional): Logistics placement

}

MaterialV3 {

itemNumber (string): Identifies the material, **materialGroup** (MaterialGroup): Name of the material group that the material belongs to, **periodical** (Periodical, *optional*): Present if material is a periodical, available (boolean): True if material is available on-shelf, false if lent out

}

MaterialGroup {

name (string): Name of the material group, description (string, optional): Description of the material group

}

Periodical {

volume (string, optional),

volumeYear (string, optional),

displayText (string): A representation of the periodica volume information that is suitable for display, **volumeNumber** (string, *optional*)

}

PlacementV1 {

section (AgencySection, optional): Placement section, location (AgencyLocation, optional): Placement location, sublocation (AgencySublocation, optional): Placement sublocation, department (AgencyDepartment, optional): Placement department

}

AgencySection {

sectionId (string): Section identifier, title (string): Name of the section

}

AgencyLocation {

locationId (string): Location identifier, **title** (string): Name of the location

}

AgencySublocation {

title (string): Name of the sub-location, **sublocationId** (string): Sub-location identifier

}

AgencyDepartment {

departmentId (string): Department identifier, **title** (string): Name of the department

}

AgencyBranch {

401	client unauthorized	
400	bad request	
HTTP Status Code	Reason	Response Model
Response Message	es	
Response Content T	ype application/json V	
}		
branchId (string): IS title (string): Name	SIL of branch (e.g. DK-761501), of branch	
PM		Swagger UI

GET	/external/{agenc	yid}/catalog/holo	dingsLogistics/v2
-----	------------------	-------------------	-------------------

Get placement holdings for bibliographical records.

Implementation Notes

Returns an array of holdings for each bibliographical record. The holdings lists the materials on each placement, and whether they are available on-shelf or lent out. If the materials are lent out, a return date will also be available.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
recordid	Identifies the bibliographical records - The FAUST number.	query	array[string]
exclude	Identifies the branchIds which are excluded from the result	query	array[string]

Response Class

Model Model Schema

HoldingsForBibliographicalRecordLogisticsV2 {

recordId (string): Identifies the bibliographical record for the available materials, The FAUST number, reservations (integer): Total number of current active reservations for the bibliographical record, reservable (boolean): True if there is any reservable materials,

holdings (array[HoldingsLogisticsV2]): An array of holdings for the materials matching the bibliographical record, as distributed across placements

}

HoldingsLogisticsV2 {

materials (array[MaterialV4]): Materials that belongs to this placement,

ImsPlacement (PlacementV1, optional): LMS placement,

branch (AgencyBranch): Placement branch,

logisticsPlacement (array[string], *optional*): Logistics placement

}

MaterialV4 {

itemNumber (string): Identifies the material,

returnDate (string): Return date of the material if it is lendout. Format: yyyy-mm-dd,

materialGroup (MaterialGroup): Name of the material group that the material belongs to,

periodical (Periodical, optional): Present if material is a periodical,

available (boolean): True if material is available on-shelf, false if lent out

}

MaterialGroup {

name (string): Name of the material group, description (string, optional): Description of the material group

}

Periodical {

volume (string, optional),
volumeYear (string, optional),
displayText (string): A representation of the periodica volume information that is suitable for display,
volumeNumber (string, optional)

}

PlacementV1 {

section (AgencySection, optional): Placement section, location (AgencyLocation, optional): Placement location, sublocation (AgencySublocation, optional): Placement sublocation, department (AgencyDepartment, optional): Placement department

}

AgencySection {

sectionId (string): Section identifier, title (string): Name of the section 1

f Ageneral exertion (
AgencyLocation {	Le continue informatifie a	
locationId (string):		
title (string): Name	of the location	
}		
AgencySublocation {		
	of the sub-location,	
sublocationId (stri	ng): Sub-location identifier	
}		
AgencyDepartment	{	
departmentId (stri	ng): Department identifier,	
title (string): Name	of the department	
}		
AgencyBranch {		
branchId (string): I	SIL of branch (e.g. DK-761501),	
title (string): Name	of branch	
}		
Response Content 1	ype application/json 🗸	
Response Messag	es	
HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	
GET /external/	{agencyid}/catalog/number(fLoansPerRecord/v1 Get the number of loans for the given records in the given period.

Implementation Notes

Returns an array containing the amount of loans for each bibliographical record, created in the given period. If startDate and endDate is omitted, the number of loans for the given record is returned, without filtering on date. If a recordId is not found, it will not appear in the response array.

Parameters

5, 1:11 PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
recordid	Identifies the bibliographical records - The FAUST number.	query	array[string]
startDate	The start date of the time interval within which loans are counted for the specified records. Format: yyyy-mm-dd. If the parameter is not set, all loans until the endDate will be counted.	query	string
endDate	The end date of the time interval within which loans are counted for the specified records. Format: yyyy-mm-dd. if the parameter is not set, all loans since the startDate will be counted.	query	string
Response Class Model Model Schem NumberOfLoansPerRe recordId (string): The			
	reger): Number of loans on the record.		
Response Content Ty	pe application/json 🗸		
Response Messages	S		
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
Interlibrary Loai	ns		Show/Hide List Operations Expand Operations Raw
POST /external/{a	gencyid}/ill/requestitem/v1 Initiate a req	uest for an interlibrary	v loan for a given requesting agency, recordId, and enduserId.

Implementation Notes

The parameters for the requesting Agency, enduserId, and recordId are placed in the interlibrary loan request. If the record is a periodical, material periodical information also needs to be added in the request. The result of the request contains the field "successful", which can be true/false, and a result, which can be one of these values:

PATRON_NOT_FOUND: No library loaner exists for the requesting agency
 RECORD_NOT_FOUND: No record exists for the given recordId
 MATERIAL_NOT_RESERVABLE: None of the materials on the record can be reserved
 SENDING_FAILURE: We were not able to send the item requested message to the requesting agency

- OK: request was successful

The request is only successful if the result is OK, and successful is set to true.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. NO-2030000)	path	string
interlibraryLoanRequest	interlibrary loan request	body	Model Schema
			InterlibraryLoanRequestV1 { recordId (string): Cicero-ID of the Bibliographic Record being requested, periodical (PeriodicalInformation, optional): Mandatory if requesting a periodical, requestingAgency (string): AgencyISIL for the requesting agency initiating the InterlibraryLoan, enduserId (string): Patron identifier number (e.g. library card number or social security card number) for the patron requesting the InterlibraryLoan }
			PeriodicalInformation { volume (string, optional), number (string, optional), year (string, optional)
			}
Response Class			

Model Model Schema

11 PM		Swagger UI	
	NOT_FOUND' or 'RECORD_NOT_FOUND' or 'MAT if the operation was successful, False if not	'ERIAL_NOT_RESERVABLE'	or 'SENDING_FAILURE' or 'OK'],
Newsletter		S	how/Hide List Operations Expand Operations Raw
POST /external/{agenc	cyid}/newsletters/preferences/patron/v Fetch		with preferences that a patron subscribed to with an email.
Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
subscriptionRequest	patronId and email	body	Model Schema
			NewsletterPatronSubscriptionRequestV1 { emailAddress (string): The email address of the patron, patronId (integer): The id of the patron }
Response Class			
Model Model Schema			
		wsletter preferences	

}

NewsletterTagV1 {

name (string): The preference name stored in Mailchimp,

id (string): The preference id stored in Mailchimp

}		
NewsletterGroupCat	tegoryV1 {	
id (string): The prefe	erence id stored in Mailchimp,	
title (string): The pr	reference name stored in Mailchimp,	
	ewsletterGroupNameV1], <i>optional</i>): The preference name stored in Mailchimp	
}		
NewsletterGroupNa	ImeV1 {	
-	preference name stored in Mailchimp,	
	erence id stored in Mailchimp	
id (string): The prefe		
id (string): The prefe }	erence id stored in Mailchimp	
id (string): The prefe }		
id (string): The prefe }	erence id stored in Mailchimp Type application/json ❤	
id (string): The prefe } Response Content T	erence id stored in Mailchimp Type application/json ❤	
id (string): The prefe } Response Content T Response Messag	erence id stored in Mailchimp Type application/json ✔ ;es	
id (string): The prefe } Response Content T Response Message HTTP Status Code	erence id stored in Mailchimp Type application/json ✓ res Reason Response Model	
id (string): The prefe Response Content T Response Messag HTTP Status Code 400	Fype application/json ✓ res Reason Response Model bad request	

GET /external	/{agencyid}/newsletters/preferences/v1			
	Fetche	s the newsletters with prefere	ences from the Mailchimp platform	configured on the Agency
Parameters				
Parameter	Description	Parameter Type	Data Type	
agencyid	ISIL of the agency (e.g. DK-761500)	path	string	
Response Class Model Model Sch NewsletterV1 { listId (string): New name (string): The	vsletter contact list id,			

Parameters Parameter	Description ISIL of the agency (e.g. DK-761500)	r di di littere i type	Data Type
Parameters		Parameter Type	Data Type
	Subscribe to newslette	r preferences with an ema	il from the Mailchimp platform configured on the Age
PUT /external/	{agencyid}/newsletters/subscribe/v1		
404	Mailchimp configuration not found		
401	client unauthorized		
400	bad request		
HTTP Status Code	Reason	Response Model	
Response Messag	es		
Response Content 1	ype application/json 🗸		
-	preference name stored in Mailchimp, erence id stored in Mailchimp		
NewsletterGroupNa			
(F	
	reference name stored in Mailchimp, wsletterGroupNameV1], <i>optional</i>): The preference name	stored in Mailchimp	
	erence id stored in Mailchimp,		
NewsletterGroupCa	tegoryV1 {		
ia (string): The pref	erence id stored in Mailchimp		
	preference name stored in Mailchimp,		
NewsletterTagV1 {			
0		·	
	:terTagV1],	vsletter preferences	

5/5/25, 1:11	PM		Swagger UI	
	Parameter	Description	Parameter Type	Data Type
	subscriptionDetails		body	Model Schema
				NewsletterSubscriptionDetailsV1 {
				emailAddress (string): Patron email,
				subscriptions (array[NewsletterV1], optional):
				Subscription details
				}
				NewsletterV1 {
				listId (string): Newsletter contact list id,
				name (string): The newslatter name,
				tags (array[NewsletterTagV1], optional): The
				newsletter preferences,
				interestCategories
				(array[NewsletterGroupCategoryV1], optional): The
				newsletter preferences
				}
				NewsletterTagV1 {
				name (string): The preference name stored in
				Mailchimp, id (string): The preference id stored in Mailchimp
				}
				NewsletterGroupCategoryV1 { id (string): The preference id stored in Mailchimp,
				title (string): The preference in stored in Malichimp,
				Mailchimp,
				interests (array[NewsletterGroupNameV1], optional):
				The preference name stored in Mailchimp
				}
				NewsletterGroupNameV1 {
				name (string): The preference name stored in
				Mailchimp,
				id (string): The preference id stored in Mailchimp
				}
	Response Messages			

1.1			
	HTTP Status Code	Reason	Response Model
	400	bad request	
	401	client unauthorized	
	404	Mailchimp configuration not found	

PUT /external/{agencyid}/newsletters/unsubscribe/v1

Unsubscribe from newsletter preferences with an email from the Mailchimp platform configured on the Agency

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
subscriptionDetails		body	Model Schema
			<pre>NewsletterSubscriptionDetailsV1 { emailAddress (string): Patron email, subscriptions (array[NewsletterV1], optional): Subscription details } NewsletterV1 { listId (string): Newsletter contact list id, name (string): The newslatter name, tags (array[NewsletterTagV1], optional): The newsletter preferences, interestCategories (array[NewsletterGroupCategoryV1], optional): The newsletter preferences } NewsletterTagV1 { name (string): The preference name stored in Mailchimp, id (string): The preference id stored in Mailchimp }</pre>
			NewsletterGroupCategoryV1 {

Parameter	Description	Parameter Type Data Type	
		id (string): The p title (string): The Mailchimp, interests (array[The preference r } NewsletterGroup name (string): Th Mailchimp,	reference id stored in Mailchin preference name stored in NewsletterGroupNameV1], <i>oj</i> name stored in Mailchimp NameV1 { ne preference name stored in reference id stored in Mailchin
Response Messag	zes		
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
404	Mailchimp configuration not found		
Balances	/{agencyid}/patron/{patronid}/fee/	Show/Hide List Operations/v1	tions Expand Operation
		List of transactions in Cicero for the fee with all available	information about the trai
Implementation I Returns array of tra Each transactions in The list of available creation reduction cancellation	ansactions. n the response includes a 'type', which	is used to distinguish between different types of transac	tions.

write_off write_off_reduction

While the type can be used by client systems to look up a suitable display message for the end user, it is important that unrecognized types are treated as 'other'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the fee with transactions	path	integer
feeId	the fee that the transactions belong to	path	integer

Response Class

Model Model Schema

FeeTransaction {

amount (number): The amount of the transaction, in the currency of the agency,

creationTime (string): The date and time the transaction was created,

type (string): Can be used to distinguish between different types of transactions,

feeId (integer): Identifies the fee,

transactionId (integer): Identifies the transaction,

parentTransactionId (integer, *optional*): The id of the parent transaction if it has one.

}

Response Content Type application/json

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

/external/{agencyid}/patron/{patronid}/fee/expected-fees/v1

GET

Swagger UI Returns a list of expected fees of a patron, if all overdue loans were to be returned today.

Implementation Notes

This endpoint is intended for use with event-based fees.

If Cicero is configured to use time-based fees, this operation will always return an empty list.

Note: Expected fees vary from normal fees in the following ways:

The fee id (bill number) is -1. The creation date is today. The due date is null. The payableByClient field is false.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the fees	path	integer

Response Class

Model Model Schema

FeeV2 {

payableByClient (boolean): true if the client system is allowed to offer payment for the fee, false if not allowed,

amount (number): The amount to pay, in the currency of the agency,

paidDate (string, optional): If the fee has been paid in full, this will be set to the date of the final payment, otherwise not set,

materials (array[FeeMaterialV2]): Set if fee covers materials,

reasonMessage (string): Human readable free text message about the reason for the fee, presentable to an end user (language is likely to be the mother tongue of the agency),

dueDate (string, *optional*): Expected payment due date,

type (string): Can be used to distinguish between different types of fees,

creationDate (string): The date the fee was created,

feeId (integer): Identifies the fee, used when registering a payment that covers the fee

}

FeeMaterialV2 {

recordId (string): The FAUST number of the bibliographic record,

11 PM		Swagger UI	
 materialGroup (MaterialGroup): The material group that the material periodical (Periodical, <i>optional</i>): Present if material is a periodical, materialItemNumber (string): Identifies the exact material covered 			
}			
MaterialGroup {			
name (string): Nam	e of the material group,		
description (string	optional): Description of the material group		
}			
Periodical {			
<pre>volume (string, optional), volumeYear (string, optional), displayText (string): A representation of the periodica volume information that is suitable for display,</pre>			
volumeNumber (s }	ning, optional)		
Response Content 1	ype application/json 🗸		
Response Messag	es		
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
GET /external/	{agencyid}/patron/{patronid}/fees/v2	List of fees in FBS for the patron with all available information about the fee.	

Implementation Notes

Returns array of fees.

If the fee covers loaned materials, information about the materials is returned. Each fee in the response includes a 'type', which is used to distinguish between different types of fees.

If the material exists no more, which is the case for fees that are related to closed interlibraryloans, then the fee is still returned, but without material information

The list of available types currently is fee

While the type can be used by client systems to look up a suitable display message for the end user, it is important that unrecognized types are treated as 'other'.

Parameters

compensation

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the fees	path	integer
includepaid	true if all paid/unpaid fees should be included, false if only unpaid fees should be included; default=false	query	boolean
includenonpayable	true if fees that are not payable through a CMS system should be included (for read only access); default=false	query	boolean

Response Class

Model Model Schema

FeeV2 {

payableByClient (boolean): true if the client system is allowed to offer payment for the fee, false if not allowed,

amount (number): The amount to pay, in the currency of the agency,

paidDate (string, optional): If the fee has been paid in full, this will be set to the date of the final payment, otherwise not set,

materials (array[FeeMaterialV2]): Set if fee covers materials,

reasonMessage (string): Human readable free text message about the reason for the fee, presentable to an end user (language is likely to be the mother tongue of the agency),

dueDate (string, optional): Expected payment due date,

type (string): Can be used to distinguish between different types of fees,

creationDate (string): The date the fee was created,

feeId (integer): Identifies the fee, used when registering a payment that covers the fee

FeeMaterialV2 {

recordId (string): The FAUST number of the bibliographic record, materialGroup (MaterialGroup): The material group that the material belongs to, periodical (Periodical, optional): Present if material is a periodical,

}

materialItemNum }	ber (string): Identifies the e	kact material covered by the fee
MaterialGroup {		
name (string): Nam	ne of the material group,	
description (string	, <i>optional</i>): Description of th	e material group
}		
Periodical {		
volume (string, opt	tional),	
volumeYear (string	s, optional),	
displayText (string volumeNumber (s		riodica volume information that is suitable for display,
}		
Response Content	Type application/json ❤	
Response Messag	jes	
HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

POST	/external/{agencyid}/patron/{patronid}/payment/v2	Pay fees.
------	---	-----------

Implementation Notes

Returns array of payment confirmations for each fee.

This call is used to inform FBS that a payment have been completed successful from the payment gateway through the CMS client system. The payment contain the order ID from the payment gateway (e.g. dibs) and the fee identifiers for fees covered by the payment. It is expected that a fee has been paid in full when covered by a payment order. The client system is not allowed to offer partial payment of individual fees.

The paymentStatus on the response can be any of these values:

- paymentRegistered
- paymentRegisteredByDifferentOrderId
- paymentNotAllowedByClient

If any other value is encountered, it should be treated as yet another reason for not registerering payment of a fee using the specified order id.

Multiple calls to pay a fee with the same order Id will return the same confirmationId, and the payment will have paymentStatus=='paymentRegistered'.

If a fee has already been paid using a different orderId then no confirmationId is provided, and the payment will have paymentStatus=='paymentRegisteredByDifferentOrderId'.

If the client system was not allowed to offer payment of a fee, then no confirmationId is provided, and the payment will have paymentStatus=='paymentNotAllowedByClient'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the fees	path	integer
paymentOrder	registration of fees covered by a payment order	body	Model Model Schema PaymentOrder { orderId (string): Order Id from payment gateway, feeIds (array[integer]): Array of fees fully covered by the order }

Response Class

Model Model Schema

PaymentConfirmationV2 {

orderId (string): Order Id from payment gateway,
 confirmationId (string, optional): set if fee was registered when using the orderId, unset otherwise (see paymentStatus for reason),
 feeId (integer),
 paymentStatus (string)

}

Response Content Type application/json

```
Response Messages
```

Swagger UI

•••			
	HTTP Status Code	Reason	Response Model
	400	bad request	
	401	client unauthorized	

POST	/external/{ag	encyid}/patron/{patronid}/service-fee/v1			Create a service fee for a patron.
Param	neters				
Param	eter	Description	Parameter Type	Data Type	
agenc	yid	ISIL of the agency (e.g. DK-761500)	path	string	
patro	nid	the patron that the fee is created for	path	integer	
amoun	t	the amount of the service fee. Must be between 0 and 99999999, with no more than 2 decimal places.	body	number	
Respo	nse Messages				
HTTP S	Status Code	Reason	Response Model		
400		bad request			
401		client unauthorized			
403		forbidden access to wrong agency			

patron not found

Show/Hide List Operations Expand Operations Raw /external/{agencyid}/patrons/{patronid}/loans/{bookingid}/v2 Retrieves material loans for the given booking ID. **Implementation Notes** Retrieves an array of BookingLoan corresponding to the given booking ID.

404

5/5/25, 1	1:11 PM
-----------	---------

• •			Swagger Of	
	Parameters			
	Parameter	Description	Parameter Type	Data Type
	agencyid	ISIL of the agency (e.g. DK-761500)	path	string
	patronid	the ID of the patron that owns the bookings	path	integer
	bookingid	the ID of the booking	path	string

Response Class

Model Model Schema

BookingLoanV2 {

patronName (string): The name of the patron that owns the booking, isRenewable (boolean): indicates whether this loan can be renewed, loanDetails (LoanDetailsV2): The loan that was attempted renewed, isLongtermLoan (boolean): indicates whether this loan is a long term loan, renewalStatusList (array[string]): if isRenewable == false then this states the reasons for denial

}

LoanDetailsV2 {

recordId (string): The FAUST number of the bibliographic record, loanType (string): The loan type, either loan or interLibraryLoan, materialGroup (MaterialGroup): Material group that the material belongs to, periodical (Periodical, optional): Present if material is a periodical, dueDate (string): The date when the material must be returned, ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans, loanDate (string): The date when the material was picked up, materialItemNumber (string): Identifies the exact material that has been loaned, loanId (integer): Identifies the loan for use when renewing the loan

}

MaterialGroup {

name (string): Name of the material group, description (string, optional): Description of the material group

}

Periodical {

volume (string, optional),

11	PM		Swagger UI
	<pre>volumeYear (string, displayText (string): volumeNumber (str }</pre>	A representation of the periodica volume information that	at is suitable for display,
	, ILLBibliographicRecor	rd I	
		<i>nal</i>): The author of the material, <i>I</i>): ISBN-information from the bibliographic record,	
		(string, <i>optional</i>): Issue number of a periodical,	
	-	<i>onal</i>): Edition-information from the bibliographic record,	
		<i>itional</i>): Language of the requested material.,	
		ory (string, <i>optional</i>): Bibliographic category from danMAF	RC2 008 *t
	• • •	<i>f</i>): The title of the material,	
	. .	Component (string, <i>optional</i>): Publication date of an item	component or article
	recordId (string): The		component, or untrelet,
): ISSN-information from the bibliographic record,	
	placeOfPublication	C .	
	•	g): Type of the requested material - from danMARC2 009 *	a+*g (general and specific),
		string, <i>optional</i>): Volume name of a periodical,	
		<i>ptional</i>): Publisher of the requested material.,	
		ring, optional): Publication date of the requested material	
	}		
	-		
	Response Content Ty	/pe application/json 🗸	
	Response Message	!S	
	HTTP Status Code	Reason	Response Model
	400	had request com dantak di rest DestEvcention	

400	bad request com.dantek.dl.rest.RestException
401	client unauthorized
404	patron not found

POST /external/{agencyid}/patrons/{patronid}/loans/renew/v2

Implementation Notes

Renew loans.

Returns an array of the updated loans.

If the materials could not be renewed, the return date will be unchanged.

The response field renewalStatus will contain a list of one or more of these values:

- renewed
- deniedReserved
- deniedMaxRenewalsReached
- deniedLoanerIsBlocked
- deniedMaterialIsNotLoanable
- deniedMaterialIsNotFound
- deniedLoanerNotFound
- deniedLoaningProfileNotFound
- deniedOtherReason

If any other value is encountered then it must be treated as 'deniedOtherReason'.

The response contains the field loanDetails.loanType, which can be any of these values:

- loan
- interLibraryLoan

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'other' .

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the loans	path	integer
materialLoanIds	a list of loanId to be renewed	body	array[integer]

Response Class

Model Model Schema

RenewedLoanV2 {

IoanDetails (LoanDetailsV2): The Ioan that was attempted renewed, **renewalStatus** (array[string]): indicates if renewal was succesful or denied - including the reas

renewalStatus (array[string]): indicates if renewal was succesful or denied - including the reason for denial.

}

LoanDetailsV2 {

recordId (string): The FAUST number of the bibliographic record, loanType (string): The loan type, either loan or interLibraryLoan, materialGroup (MaterialGroup): Material group that the material belongs to,

periodical (Periodical, optional): Present if material is a periodical,

dueDate (string): The date when the material must be returned,

ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans,

loanDate (string): The date when the material was picked up,

materialItemNumber (string): Identifies the exact material that has been loaned,

loanId (integer): Identifies the loan for use when renewing the loan

}

MaterialGroup {

name (string): Name of the material group, description (string, optional): Description of the material group

}

Periodical {

volume (string, *optional*),

volumeYear (string, optional),

displayText (string): A representation of the periodica volume information that is suitable for display,

volumeNumber (string, optional)

}

ILLBibliographicRecord {

author (string, optional): The author of the material, isbn (string, optional): ISBN-information from the bibliographic record, periodicalNumber (string, optional): Issue number of a periodical, edition (string, optional): Edition-information from the bibliographic record, language (string, optional): Language of the requested material., bibliographicCategory (string, optional): Bibliographic category from danMARC2 008 *t, title (string, optional): The title of the material, publicationDateOfComponent (string, optional): Publication date of an item component, or article., recordId (string): The FAUST number, issn (string, optional): ISSN-information from the bibliographic record, placeOfPublication (string, optional), mediumType (string): Type of the requested material - from danMARC2 009 *a+*g (general and specific), periodicalVolume (string, optional): Volume name of a periodical, publisher (string, optional): Publisher of the requested material., publicationDate (string, optional): Publication date of the requested material. }

localhost:8080/externalapidocs/

Swagor III

TT PM			Swagger UI	
Respon	nse Content Ty	/pe application/json ✓		
Respo	nse Message	S		
HTTP S	tatus Code	Reason	Response Model	
400		bad request		
401		client unauthorized		
404		patron not found		
GET	/external/{a	agencyid}/patrons/{patronid}/loans/v2		Get list of current loans by the patron.

Implementation Notes

Returns an array of loans.

If a loan is not renewable then the field renewalStatus will contain a list of one or more of these values:

- deniedReserved
- deniedMaxRenewalsReached
- deniedLoanerIsBlocked
- deniedMaterialIsNotLoanable
- deniedMaterialIsNotFound
- deniedLoanerNotFound
- deniedLoaningProfileNotFound deniedOtherReason

If any other value is encountered then it must be treated as 'deniedOtherReason'.

The response contains the field loanDetails.loanType, which can be any of these values:

- loan
- interLibraryLoan

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'other'.

NOTE: Cicero can decide to skip evaluation of the returned loans to minimize response time for loaners with many loans. In that case isRenewable will have the value true, as if it were a successful validation.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string

Parameter	Description	Parameter Type	Data Type
patronid	the patron that owns the loans	path	integer
Response Class			
Model Model Sch	nema		
LoanV2 {			
-	olean): indicates whether this loan can be renewed	,	
-	DetailsV2): The loan that was attempted renewed		
	(boolean): indicates whether this loan is a long ter		
	st (array[string]): if isRenewable == false then this		
}			
LoanDetailsV2 {			
recordId (string):	The FAUST number of the bibliographic record,		
loanType (string)	: The loan type, either loan or interLibraryLoan ,		
materialGroup (MaterialGroup): Material group that the material b	elongs to,	
periodical (Perio	dical, <i>optional</i>): Present if material is a periodical,		
dueDate (string):	The date when the material must be returned,		
	ecord (ILLBibliographicRecord, optional): Additiona	al bibliographic information fo	r inter-library loans,
	: The date when the material was picked up,		
	mber (string): Identifies the exact material that ha		
	dentifies the loan for use when renewing the loan		
}			
MaterialGroup {			
MaterialGroup { name (string): Na	me of the material group,		
MaterialGroup { name (string): Na	me of the material group, ng, <i>optional</i>): Description of the material group		

volumeYear (string, optional),

displayText (string): A representation of the periodica volume information that is suitable for display,

volumeNumber (string, optional)

}

ILLBibliographicRecord {

author (string, optional): The author of the material,

5/5/25, 1:11 PM		Swagger UI
isbn (string, optional periodicalNumber edition (string, opti- language (string, opti- bibliographicCateg title (string, optional publicationDateOf recordId (string): The issn (string, optional placeOfPublication	he FAUST number, a/): ISSN-information from the bibliog n (string, <i>optional</i>),	graphic record, periodical, pibliographic record, material., category from danMARC2 008 *t, cation date of an item component, or article.,
publisher (string, <i>o</i>	(string, optional): Volume name of a ptional): Publisher of the requested ptional): Publisher of the requested ptring, optional): Publication date of the string optional op	material.,
} Response Content T	ype application/json 🗸	
Response Messag	es	
HTTP Status Code	Reason	Response Model

	Keason	Kesponse model
400	bad request	
401	client unauthorized	
404	patron not found	

GET /external/{agencyid}/patrons/{patronid}/loans/withhistoricalloans/v1 Retrieves material loans and historical loans for the given patron.

Implementation Notes

Retrieves an object containg two arrays of loans and historical loans corresponding to the given patron ID. If the patron has not given consent to keep historical data, no historical loans will be retrieved.

5/5/25, 1:1	1 PM		Swagger UI	
	Parameter	Description	Parameter Type	Data Type
	agencyid	ISIL of the agency (e.g. DK-761500)	path	string
	patronid	the ID of the patron that owns the loans	path	integer

Response Class

Model Model Schema

LoansWithHistoryV1 {

loans (array[LoanV2]): List of loans, **historicalMaterialLoans** (array[HistoricalLoanV1]): List of historical material loans

}

LoanV2 {

isRenewable (boolean): indicates whether this loan can be renewed, loanDetails (LoanDetailsV2): The loan that was attempted renewed, isLongtermLoan (boolean): indicates whether this loan is a long term loan, renewalStatusList (array[string]): if isRenewable == false then this states the reasons for denial

}

LoanDetailsV2 {

recordId (string): The FAUST number of the bibliographic record, loanType (string): The loan type, either loan or interLibraryLoan, materialGroup (MaterialGroup): Material group that the material belongs to, periodical (Periodical, optional): Present if material is a periodical, dueDate (string): The date when the material must be returned, ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans, loanDate (string): The date when the material was picked up, materialItemNumber (string): Identifies the exact material that has been loaned, loanId (integer): Identifies the loan for use when renewing the loan

}

MaterialGroup {

name (string): Name of the material group, description (string, optional): Description of the material group

}

Periodical {

volume (string, *optional*), **volumeYear** (string, *optional*),

displayText (string): A representation of the periodica volume information that is suitable for display, volumeNumber (string, optional)

}

ILLBibliographicRecord {

author (string, optional): The author of the material, isbn (string, optional): ISBN-information from the bibliographic record, periodicalNumber (string, optional): Issue number of a periodical, edition (string, optional): Edition-information from the bibliographic record, language (string, optional): Language of the requested material., bibliographicCategory (string, optional): Bibliographic category from danMARC2 008 *t, title (string, optional): The title of the material, publicationDateOfComponent (string, optional): Publication date of an item component, or article., recordId (string): The FAUST number, issn (string, optional): ISSN-information from the bibliographic record, placeOfPublication (string, optional), mediumType (string): Type of the requested material - from danMARC2 009 *a+*g (general and specific), periodicalVolume (string, optional): Volume name of a periodical, publisher (string, optional): Publisher of the requested material., publicationDate (string, optional): Publication date of the requested material.

HistoricalLoanV1 {

recordId (string): The FAUST number of the bibliographic record, loanType (string): The loan type, either loan or interLibraryLoan, periodical (Periodical, optional): Present if material is a periodical, dueDate (string): The date when the material must be returned, ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans, loanDate (string): The date when the material was picked up, materialItemNumber (string): Identifies the exact material that has been loaned, loanId (integer): Identifies the loan for use when renewing the loan, returnedDate (string, optional): Date when the material was returned

}

}

Response Content Type | application/json ~

Response Messages

PM						
HTTP Status Code	Reason	Response Model				
400	bad request com.dantek.dl.rest.RestException					
401	client unauthorized					
404	patron not found					
eservations v	ersion 2		Show/Hide	List Operations	Expand Operations	Ra
PUT /external/	'{agencyid}/patrons/{patronid}/reservations	:/v2			Update existing reserv	atior
The response conta - reserved	Notes The updated reservation details. Ains reservation state, which can be any of these	e values:				
Returns an array of The response conta - reserved - readyForPickup - inTransit - other The values are sub	the updated reservation details.	untered, it should be tre	ated as 'othei	r'.		
Returns an array of The response conta - reserved - readyForPickup - inTransit - other The values are sub	f the updated reservation details. ains reservation state, which can be any of these lect to change. If an unrecognized value is encou	untered, it should be tre	ated as 'othei	r'.		
Returns an array of The response conta - reserved - readyForPickup - inTransit - other The values are sub The activation date	f the updated reservation details. ains reservation state, which can be any of these lect to change. If an unrecognized value is encou	untered, it should be tre	ated as 'othei Data Typ			
Returns an array of The response conta - reserved - readyForPickup - inTransit - other The values are sub The activation date	f the updated reservation details. ains reservation state, which can be any of these fect to change. If an unrecognized value is encou can only be updated for active or passive reserv	untered, it should be tre vations.				
Returns an array of The response conta - reserved - readyForPickup - inTransit - other The values are sub The activation date Parameters Parameter	f the updated reservation details. ains reservation state, which can be any of these fect to change. If an unrecognized value is encou can only be updated for active or passive reserv Description	untered, it should be tre vations. Parameter Type	Data Typ			

UpdateReservationBatchV2 {

reservations (array[UpdateReservationV2])

}

UpdateReservationV2 {

expiryDate (string, optional): The date where the patron is no longer interested in the reserved material. If not set,

5/5/25,

1 PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
			the reservation will keep the original date., pickupBranch (string, optional): ISIL-number of pickup branch. If not set, the reservation will keep the original pickup branch., reservationId (integer): Identifies the reservation, activationDate (string, optional): Sets the activation dat From this date, the reservation can be fulfilled.
			1
Response Class			
Model Model Sch	iema		
pickupDeadline dateOfReservati ilBibliographicRe numberInQueue pickupNumber (cord (ILLBibliographicRecord, <i>option</i> (integer, <i>optional</i>): The number in the string, <i>optional</i>): The reservation number in the num	erial is available for loan, <i>nal</i>): Additional bibliographic information for ne reservation queue., nber. Will be present if the reservation is reac	
recordId (string):	ring): Identifies the transaction of res The FAUST number,		
		nger interested in the reserved material,	
	0	use when updating or deleting the reservation	n,
reservationType	dical, <i>optional</i>): Present if material is	a periodical,	

state (string),

activationDate (string, optional): Sets the activation date. From this date, the reservation can be fulfilled.

}

ILLBibliographicRecord {

author (string, *optional*): The author of the material, **isbn** (string, *optional*): ISBN-information from the bibliographic record,

periodicalNumber (string, *optional*): Issue number of a periodical,

edition (string, optional): Edition-information from the bibliographic record,

language (string, optional): Language of the requested material.,

bibliographicCategory (string, optional): Bibliographic category from danMARC2 008 *t,

title (string, optional): The title of the material,

Swadder UI

IT PM		Swagger U		
<pre>publicationDateOfComponent (string, optional): Publication date of an item component, or article., recordId (string): The FAUST number, issn (string, optional): ISSN-information from the bibliographic record, placeOfPublication (string, optional), mediumType (string): Type of the requested material - from danMARC2 009 *a+*g (general and specific), periodicalVolume (string, optional): Volume name of a periodical, publisher (string, optional): Publisher of the requested material., publicationDate (string, optional): Publication date of the requested material.</pre>				
}				
Periodical {				
<pre>volume (string, optic volumeYear (string, optic displayText (string): volumeNumber (string);</pre>	<i>optional</i>), A representation of the periodica volume infor	nation that is suitable for display,		
Response Content Ty	vpe application/json ✓			
Response Messages	S			
HTTP Status Code	Reason	Response Model		
400	bad request			
401	client unauthorized			

GET	/external/{agencyid}/patrons/{patronid}/reservations/v3	Get all unfulfilled reservations made by the patron.
-----	---	--

Implementation Notes

Returns an array of reservation details.

When the patron picks up the reserved materials, the reservation will no longer be returned. Expired or deleted reservations will not be returned.

The response contains reservation state, which can be any of these values: - reserved - readyForPickup - inTransit

- other

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'other' .

The response contains reservationType, which can be any of these values:

- NORMAL
- PARALLEL
- SERIAL
- INTER_LIBRARY

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'normal'.

The response contains a transactionId, which links together parallel reservations.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the reservations	path	integer

Response Class

Model Model Schema

ReservationDetailsV3 {

pickupBranch (string): ISIL-number of pickup branch,

pickupDeadline (string, optional): Set if reserved material is available for loan,

dateOfReservation (string),

ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans,

numberInQueue (integer, optional): The number in the reservation queue.,

pickupNumber (string, optional): The reservation number. Will be present if the reservation is ready for pickup (the state is 'readyForPickup'),

transactionId (string): Identifies the transaction of reservations,

recordId (string): The FAUST number,

expiryDate (string): The date when the patron is no longer interested in the reserved material,

reservationId (integer): Identifies the reservation for use when updating or deleting the reservation,

periodical (Periodical, optional): Present if material is a periodical,

reservationType (string),

state (string),

activationDate (string, optional): Sets the activation date. From this date, the reservation can be fulfilled.

ILLBibliographicRecord {

}

author (string, optional): The author of the material, isbn (string, optional): ISBN-information from the bibliographic record, periodicalNumber (string, optional): Issue number of a periodical, edition (string, optional): Edition-information from the bibliographic record, language (string, optional): Language of the requested material., bibliographicCategory (string, optional): Bibliographic category from danMARC2 008 *t, title (string, optional): The title of the material, publicationDateOfComponent (string, optional): Publication date of an item component, or article., recordId (string): The FAUST number, issn (string, optional): ISSN-information from the bibliographic record, placeOfPublication (string, optional), **mediumType** (string): Type of the requested material - from danMARC2 009 *a+*g (general and specific), periodicalVolume (string, optional): Volume name of a periodical, publisher (string, optional): Publisher of the requested material., publicationDate (string, optional): Publication date of the requested material. Periodical { volume (string, optional), volumeYear (string, optional),

displayText (string): A representation of the periodica volume information that is suitable for display, volumeNumber (string, optional)

}

}

Response Content Type application/json 🗸

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

POST	/external/{	{agencyid	}/patrons/{	patronid	/reservations/v3

Create new reservations for the patron.

Implementation Notes

Given a CreateReservationBatch, it creates a list of reservations and returns a ReservationResponse.

The CreateReservationBatch.type indicates the reservation type of the request. If left out the request will be considered of type normal. The type can be any of the following values:

- normal
- parallel

The values are subject to change.

ReservationResponse.success indicates if the reservations were created sucessfully. If all reservations are successfully created ReservationResponse.success will be true. If the type is normal, and any of the reservations have failed then ReservationResponse.success will be false, and no reservations are created. If the type is parallel, and any of the reservations have failed then ReservationResponse.success will be false, but successful reservations are created.

ReservationResponse.reservationResults contains details about each reservation. A ReservationResult.result has the status of a reservation and can be any of the following values:

- success

- patron_is_blocked
- patron_not_found
- already_reserved
- already_loaned
- material_not_loanable
- material_not_reservable
- material_lost
- material_Discarded
- loaning_profile_not_found
- material_not_found
- material_part_of_collection
- not_reservable
- no_reservable_materials
- interlibrary_material_not_reservable
- previously_loaned_by_homebound_patron

- exceeds_max_reservations

The values are subject to change. If an unrecognized value is encountered, it should be treated as an error.

The reservation detail in the response contains a reservation state, which can be any of these values:

- reserved
- readyForPickup
- inTransit
- other

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'other'.

When making a reservation of a periodical, the values to put in the PeriodicalReservation structure can be obtained from the periodical information retrieved with the Catalog service.

Parameter	Description	Parameter Type	Data Type
	· · ·		
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that makes the reservations	path	integer
createReservationBatch	the reservations to be created	body	Model Schema
			CreateReservationBatchV3 {
			reservations (array[CreateReservationV2]):
			Reservations with duplicate record id's will be
			removed.,
			type (string, <i>optional</i>): Will be considered normal if not set
			}
			۲ CreateReservationV2 {
			recordId (string): Identifies the bibliographical
			record to reserve - The FAUST number,
			expiryDate (string, optional): The date where the
			patron is no longer interested in the reserved
			material. If not set, a date will be calculated from
			the agency default interest period,
			pickupBranch (string, <i>optional</i>): ISIL-number of
			pickup branch. If not set, will default to patrons preferred pickup branch,
			periodical (PeriodicalReservation, <i>optional</i>): Presen
			if making reservation on a periodical,
			activationDate (string, optional): Sets the activation
			date. From this date, the reservation can be fulfilled
			}
			PeriodicalReservation {
			volume (string, optional),
			volumeYear (string, optional),
			volumeNumber (string, optional)
			}

Response Class

Model Model Schema

ReservationResponseV3 {

success (boolean): True if all reservations were created successfully otherwise false, **reservationResults** (array[ReservationResultV3]): Result of each reservation

}

ReservationResultV3 {

recordId (string): Recordid of the record to reserve,

result (string): The reservation result,

periodical (PeriodicalReservation, optional): Periodical information of the reservation,

reservationDetails (ReservationDetailsV3, optional): The reservation data as returned by the create/update operation

}

PeriodicalReservation {

volume (string, optional),
volumeYear (string, optional),
volumeNumber (string, optional)

}

ReservationDetailsV3 {

pickupBranch (string): ISIL-number of pickup branch, pickupDeadline (string, optional): Set if reserved material is available for loan, dateOfReservation (string), ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans, numberInQueue (integer, optional): The number in the reservation queue., pickupNumber (string, optional): The reservation number. Will be present if the reservation is ready for pickup (the state is 'readyForPickup'), transactionId (string): Identifies the transaction of reservations, recordId (string): The FAUST number, expiryDate (string): The date when the patron is no longer interested in the reserved material, reservationId (integer): Identifies the reservation for use when updating or deleting the reservation, periodical (Periodical, optional): Present if material is a periodical, reservationType (string), state (string), activationDate (string, optional): Sets the activation date. From this date, the reservation can be fulfilled.

ILLBibliographicRecord {

}

<pre>isbn (string, optional periodicalNumber edition (string, opti- language (string, opti- bibliographicCateg title (string, optional publicationDateOf recordId (string): Th issn (string, optional placeOfPublication mediumType (string, periodicalVolume (publisher (string, op- publicationDate (st } Periodical { volume (string, opti- volumeYear (string) volumeYear (string) volumeNumber (st }</pre>	<i>ptional</i>): Language of the re gory (string, <i>optional</i>): Bibli a /): The title of the material Component (string, <i>option</i> the FAUST number, a /): ISSN-information from the c (string, <i>optional</i>), g): Type of the requested n (string, <i>optional</i>): Volume n <i>ptional</i>): Publisher of the re tring, <i>optional</i>): Publication ional), , <i>optional</i>), : A representation of the per ring , <i>optional</i>)	the bibliographic record, mber of a periodical, from the bibliographic record, equested material., ographic category from danMARC2 008 *t, , <i>nal</i>): Publication date of an item component, or article., the bibliographic record, haterial - from danMARC2 009 *a+*g (general and specific), ame of a periodical,
Response Content T	ype application/json ✔	
HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

PUT	/external/{agencyid}/patrons/{patronid}/reservations/v3
101	

patron not found

Update existing reservations.

404

Implementation Notes

Returns an array of the updated reservation details.

The response contains reservation state, which can be any of these values:

- reserved
- readyForPickup inTransit
- other

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'other'

The response contains estimatedDaysUntilReservation, which gives an estimated date for when the reservation can be fulfilled.

The activation date can only be updated for active or passive reservations.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the reservations	path	integer
reservations	the reservations to be updated	body	Model Schema
			<pre>UpdateReservationBatchV2 { reservations (array[UpdateReservationV2]) } UpdateReservationV2 { expiryDate (string, optional): The date where the patron is no longer interested in the reserved material. If not set, the reservation will keep the original date., pickupBranch (string, optional): ISIL-number of pickup branch. If not set, the reservation will keep the original pickup branch., reservationId (integer): Identifies the reservation, activationDate (string, optional): Sets the activation date. From this date, the reservation can be fulfilled. }</pre>



Response Class

Model Model Schema

ReservationDetailsV4 {

pickupBranch (string): ISIL-number of pickup branch,

estimatedDaysUntilReservation (integer, optional): Provides an estimated approximate queue time on reservation in days,

pickupDeadline (string, optional): Set if reserved material is available for loan,

dateOfReservation (string),

ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans,

numberInQueue (integer, optional): The number in the reservation queue.,

pickupNumber (string, *optional*): The reservation number. Will be present if the reservation is ready for pickup (the state is 'readyForPickup'), **transactionId** (string): Identifies the transaction of reservations,

recordId (string): The FAUST number,

expiryDate (string): The date when the patron is no longer interested in the reserved material,

reservationId (integer): Identifies the reservation for use when updating or deleting the reservation,

periodical (Periodical, optional): Present if material is a periodical,

reservationType (string),

state (string),

activationDate (string, optional): Sets the activation date. From this date, the reservation can be fulfilled.

}

ILLBibliographicRecord {

author (string, optional): The author of the material, isbn (string, optional): ISBN-information from the bibliographic record, periodicalNumber (string, optional): Issue number of a periodical, edition (string, optional): Edition-information from the bibliographic record, language (string, optional): Language of the requested material., bibliographicCategory (string, optional): Bibliographic category from danMARC2 008 *t, title (string, optional): The title of the material, publicationDateOfComponent (string, optional): Publication date of an item component, or article., recordId (string): The FAUST number, issn (string, optional): ISSN-information from the bibliographic record, placeOfPublication (string, optional), mediumType (string): Type of the requested material - from danMARC2 009 *a+*g (general and specific), periodicalVolume (string, optional): Volume name of a periodical, publisher (string, optional): Publisher of the requested material., publicationDate (string, optional): Publisher of the requested material.

localhost:8080/externalapidocs/

Periodical {

volume (string, optional),
volumeYear (string, optional),
displayText (string): A representation of the periodica volume information that is suitable for display,
volumeNumber (string, optional)

}

}

Response Content Type application/json

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

GET /external/{agencyid}/patrons/{patronid}/reservations/v4 Get all unfulfilled reservations made by the patron.

Implementation Notes

Returns an array of reservation details.

When the patron picks up the reserved materials, the reservation will no longer be returned. Expired or deleted reservations will not be returned.

The response contains reservation state, which can be any of these values:

- reserved
- readyForPickup
- inTransit
- other

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'other' .

The response contains estimatedDaysUntilReservation, which gives an estimated date for when the reservation can be fulfilled.

The response contains reservationType, which can be any of these values:

- NORMAL
- PARALLEL
- SERIAL
- INTER_LIBRARY

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'normal'.

The response contains a transactionId, which links together parallel reservations.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the reservations	path	integer

Response Class

Model Model Schema

ReservationDetailsV4 {

pickupBranch (string): ISIL-number of pickup branch,

estimatedDaysUntilReservation (integer, optional): Provides an estimated approximate queue time on reservation in days,

pickupDeadline (string, optional): Set if reserved material is available for loan,

dateOfReservation (string),

ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans,

numberInQueue (integer, optional): The number in the reservation queue.,

pickupNumber (string, *optional*): The reservation number. Will be present if the reservation is ready for pickup (the state is 'readyForPickup'), **transactionId** (string): Identifies the transaction of reservations,

recordId (string): The FAUST number,

expiryDate (string): The date when the patron is no longer interested in the reserved material,

reservationId (integer): Identifies the reservation for use when updating or deleting the reservation,

periodical (Periodical, optional): Present if material is a periodical,

reservationType (string),

state (string),

activationDate (string, optional): Sets the activation date. From this date, the reservation can be fulfilled.

ILLBibliographicRecord {

author (string, optional): The author of the material,
isbn (string, optional): ISBN-information from the bibliographic record,
periodicalNumber (string, optional): Issue number of a periodical,
edition (string, optional): Edition-information from the bibliographic record,
language (string, optional): Language of the requested material.,

}

	Swagger Of	
title (string, optional)	ory (string, <i>optional</i>): Bibliographic category from danMARC2 008 *t,): The title of the material,	
	Component (string, <i>optional</i>): Publication date of an item component, or article.,	
recordId (string): The	e FAUST number,): ISSN-information from the bibliographic record,	
placeOfPublication		
	(): Type of the requested material - from danMARC2 009 *a+*g (general and specific),	
periodicalVolume (s	string, <i>optional</i>): Volume name of a periodical,	
	<i>tional</i>): Publisher of the requested material.,	
•	ring, <i>optional</i>): Publication date of the requested material.	
}		
Periodical { volume (string, optio	onal).	
volumeYear (string, o		
displayText (string):	A representation of the periodica volume information that is suitable for display,	
volumeNumber (stri	ing, optional)	
}		
Response Content Ty	′pe_application/json ✓	
Response Messages	S	
HTTP Status Code	Reason Response Model	
400	bad request	
401	client unauthorized	

POST	/external/{agencyid}/patrons/{patronid}/reservations/v4	Create new reservations for the patron.
------	---	---

Implementation Notes

Given a CreateReservationBatch, it creates a list of reservations and returns a ReservationResponse.

The CreateReservationBatch.type indicates the reservation type of the request. If left out the request will be considered of type normal. The type can be any of the following values:

- normal

parallel The values are subject to change.

ReservationResponse.success indicates if the reservations were created sucessfully. If all reservations are successfully created ReservationResponse.success will be true. If the type is normal, and any of the reservations have failed then ReservationResponse.success will be false, and no reservations are created. If the type is parallel, and any of the reservations have failed then ReservationResponse.success will be false, but successful reservations are created.

ReservationResponse.reservationResults contains details about each reservation. A ReservationResult.result has the status of a reservation and can be any of the following values:

- success

- patron_is_blocked
- patron_not_found
- already_reserved
- already_loaned
- material_not_loanable
- material_not_reservable
- material_lost
- material_Discarded
- loaning_profile_not_found
- material_not_found
- material_part_of_collection
- not_reservable
- no_reservable_materials
- interlibrary_material_not_reservable
- previously_loaned_by_homebound_patron
- exceeds_max_reservations

The values are subject to change. If an unrecognized value is encountered, it should be treated as an error.

The response contains estimatedDaysUntilReservation, which gives an estimated date for when the reservation can be fulfilled.

The reservation detail in the response contains a reservation state, which can be any of these values:

- reserved
- readyForPickup
- inTransit
- other

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'other'.

When making a reservation of a periodical, the values to put in the PeriodicalReservation structure can be obtained from the periodical information retrieved with the Catalog service.

Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that makes the reservations	path	integer
createReservationBatch	the reservations to be created	body	Model Model Schema
			<pre>CreateReservationBatchV3 { reservations (array[CreateReservationV2]): Reservations with duplicate record id's will be removed., type (string, optional): Will be considered normal if not set recordId (string): Identifies the bibliographical recordId (string): Identifies the bibliographical recordId (string): Identifies the bibliographical recordId (string, optional): The date where the patron is no longer interested in the reserved material. If not set, a date will be calculated from the agency default interest period, pickupBranch (string, optional): ISIL-number of pickup branch. If not set, will default to patrons preferred pickup branch, periodical (PeriodicalReservation, optional): Present if making reservation on a periodical, activationDate (string, optional): Sets the activation date. From this date, the reservation can be fulfilled. } PeriodicalReservation{ volume(string, optional), volumeYear (string, optional), volumeNumber (string, optional); }</pre>
Response Class			

Model Model Schema

ReservationResponseV4 {

success (boolean): Result of each reservation,
reservationResults (array[ReservationResultV4]): Result of each reservation

}

ReservationResultV4 {

recordId (string): Recordid of the record to reserve, **result** (string): The reservation result, **periodical** (PeriodicalReservation, *optional*): Periodical information of the reservation,

reservationDetails (ReservationDetailsV4, optional): The reservation data as returned by the create/update operation

}

PeriodicalReservation {

volume (string, optional),
volumeYear (string, optional),
volumeNumber (string, optional)

}

ReservationDetailsV4 {

pickupBranch (string): ISIL-number of pickup branch, **estimatedDaysUntilReservation** (integer, *optional*): Provides an estimated approximate queue time on reservation in days, **pickupDeadline** (string, *optional*): Set if reserved material is available for loan,

dateOfReservation (string),

ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans,

numberInQueue (integer, optional): The number in the reservation queue.,

pickupNumber (string, optional): The reservation number. Will be present if the reservation is ready for pickup (the state is 'readyForPickup'),

transactionId (string): Identifies the transaction of reservations,

recordId (string): The FAUST number,

expiryDate (string): The date when the patron is no longer interested in the reserved material,

reservationId (integer): Identifies the reservation for use when updating or deleting the reservation,

periodical (Periodical, optional): Present if material is a periodical,

reservationType (string),

state (string),

activationDate (string, optional): Sets the activation date. From this date, the reservation can be fulfilled.

}

ILLBibliographicRecord {

author (string, optional): The author of the material,

periodicalNumber edition (string, opti	nal): ISBN-information from the bibliographic record, er (string, <i>optional</i>): Issue number of a periodical,	
edition (string, opti		
	stingely Edition information from the hibling working record	
language (string, o	<i>tional</i>): Edition-information from the bibliographic record,	
	<i>optional</i>): Language of the requested material.,	
	egory (string, optional): Bibliographic category from danMARC2 008 *t,	
	<i>nal</i>): The title of the material,	
	DfComponent (string, <i>optional</i>): Publication date of an item component, or article.,	
recordId (string): TI		
	nal): ISSN-information from the bibliographic record,	
-	on (string, <i>optional</i>),	
	ing): Type of the requested material - from danMARC2 009 *a+*g (general and specific), e (string, <i>optional</i>): Volume name of a periodical,	
-	<i>optional</i>): Publisher of the requested material.,	
	(string, <i>optional</i>): Publication date of the requested material.	
}		
Periodical {		
volume (string, opt	ntional	
volumeYear (string		
	g): A representation of the periodica volume information that is suitable for display,	
volumeNumber (st		
}		
Response Content T	Type application/json V	
Response Messag	ges	
HTTP Status Code	Reason Response Model	
400	bad request	
401	client unauthorized	
404	patron not found	

Patron

Show/Hide | List Operations | Expand Operations | Raw

DELETE /external/{agencyid}/patrons/{patronid}

Deletes a patron.

Parameters

T al al literes			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	Id of the patron to be deleted	path	integer

POST	/external/{agencyid}/	patrons/{patronid}	/addMailIdentifier/v1
------	-----------------------	--------------------	-----------------------

Add the email as a LoanerIdentifier of type LibraryCard for the patron.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron concerned	path	integer
emailAddress	emailAddress belonging to the patron	body	string

Response Class

Model Model Schema

EmailIdentifierV1 {

result (string) = ['TOO_MANY_LIBRARY_CARDS_ALREADY' or 'EMAIL_ADDRESS_NOT_VERIFIED' or 'EMAIL_ADDRESS_DOES_NOT_BELONG_TO_LOANER' or 'IDENTIFIER_EXISTS_ON_AGENCY' or 'UNKNOWN_ERROR' or 'EMAIL_IDENTIFIER_CREATED']: result of the attempt at creating a email identifier for the patron, **success** (boolean): true, if creation of email identifier was successful, false otherwise

Response Content Type application/json >



}

/external/{agencyid}/patrons/{patronid}/authenticationLevels/v1 Returns the authentication levels of a patron from an external source.

I1 PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer
Response Class			
Model Model Sch AuthenticationLev			
AuthenticationLev level (string),		YSICAL.	
AuthenticationLev level (string), name (string): Au }	vel {	YSICAL.	
AuthenticationLev level (string), name (string): Au }	vel { athentication level type. Possible values: ONLINE, PH t Type application/json ❤	YSICAL.	
AuthenticationLev level (string), name (string): Au } Response Content	vel { athentication level type. Possible values: ONLINE, PH t Type application/json ❤	YSICAL. Response Model	
AuthenticationLev level (string), name (string): Au } Response Content Response Messa	rel { athentication level type. Possible values: ONLINE, PH t Type application/json ∽ ages		

POST	/external/{agencyid}/patrons/{patronid}/authenticationLevels/v1	Sets the authentication levels of a patron from an external source.
------	---	---

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer
authenticationLevels	authentication levels to set	body	Model Model Schema
			AuthenticationLevel { level (string), name (string): Authentication level type. Possible values: ONLINE, PHYSICAL.

5/5/25, 1:1	/5/25, 1:11 PM		Swagger UI	
	Parameter	Description	Parameter Type	Data Type
				}
	_			
	Response Message	S		
	HTTP Status Code	Reason	Response Model	
	200	ok		
	404	loaner not found		
	400	invalid authentication level name		
	401	client unauthorized		

PUT /external/{agencyid}/patrons/{patronid}/changePassword/v1

Update patron password upon receiving the correct old password and a new password that satisfies the password requirements.

Implementation Notes

The response 400 bad request is sent back if the patron does not have a current password or if the new password is invalid. The response 404 not found is sent back if the patron is not found.

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron to be updated	path	integer
changePasswordRequestV1	the old (current) and new password. Both must be supplied	body	Model Model Schema ChangePasswordRequestV1 { oldPassword (string): The old password of the user, newPassword (string): The new password of the user

5/5/25, 1:11 PM		Swagger UI		
Parameter	Description	Parameter Type	Data	Туре
			}	
Response Class				
Model Model Sch	nema			
integer				
integer				
Response Content	Type application/json 🗸			
Response Messa	ges			
HTTP Status Code	Reason	Response Model		
400	bad request			
401	client unauthorized			
403	client forbidden			
404	patron not found			
200	Ok			
GET /externa	l/{agencyid}/patrons/{patronid}/favorites/v1			Retrieves a list of the patron's favorites lists
Parameters				
Parameter	Description	Parameter Type	Data Type	
agencyid	agencyId ISIL of the agency (e.g. DK-761500)	path	string	
patronid		path	integer	
Response Class				
Model Model Sch	iema			
FavoritesV1 {				
favorites (array[F				
patronId (integer),			
localhost 8080/externalapidocs/				

PUT	/external/{agencyid}/patrons/{patronid}/favorites/v1	Replaces the exis	sting lists of favorites of the patron with the supplied lists.			
Implementation Notes The patron's favorites lists can be deleted by supplying an empty list.						
Param	eters					
Param	eter Description	Parameter Type	Data Type			
agenc	vid agencyId ISIL of the agency (e.g. DK-761500)	path	string			
patro	nid	path	integer			
reque	tV1	body	array[UpdateFavoritesRequestV1]			

PUT /external/{agencyid}/patrons/{patronid}/guardian/v2

Update information about the patron as a guardian.

Implementation Notes

If the corresponding agency configuration through /external/v1/{agencyid}/configuration/loaner.guardianVisibility.enabled is not enabled or there is no patron found, that has the guardian visibility enabled, for the patronid parameter, then response message 403 will be sent back.

The name and address cannot be supplied by the client. If the configured person registry is not authorized to provide information about the patron, then the name and address will not be updated.

It is possible to either update just the pincode, update just some patron settings, or update both.

If a patron is blocked the reason is available as a code:

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton
 'W': self created at website

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Multiple email addresses and phone numbers can be stored for a patron.

If multiple email addresses are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If multiple phone numbers are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron to be updated	path	integer
updatePatronAsGuardian	updated information about the patron	body	Model Schema
			<pre>UpdatePatronAsGuardianRequestV2 { patron (PatronSettingsV6, optional): Set this if patron details are to be changed, pincodeChange (PincodeChange, optional): Set this if pincode is to be changed, guardianPersonId (string): Identifies the guardian by person identifier } PatronSettingsV6 { preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place If left empty default library language will be used,</pre>

M Swagger UI Parameter Description Parameter Type Data Type NotificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV1], optional): Existing email addresses are overwritten with these

values If left empty existing email addresses are

preferredPickupBranch (string): ISIL-number of

onHold (Period, optional): If not set then the patron

interests (array[string], optional): A list of interests

phoneNumbers (array[PhoneNumberV1], *optional*): Existing phonenumbers are overwritten with these values If left empty existing phonenumbers are

deleted,

is not on hold,

of the patron.,

EmailAddressV1 {

PhoneNumberV1 {

PincodeChange {

libraryCard,

emailAddress (string),

receiveNotification (boolean)

receiveNotification (boolean),

phoneNumber (string)

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

pincode (string): The new pincode for the

libraryCardNumber (string): Identifies the

libraryCard for which the pincode is to be changed.

deleted

}

}

}

}

Period {

preferred pickup branch,

guardianVisibility (boolean), receivePostalMail (boolean),

64/175

I PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
			This can be either a physical card or the CP
			number, that is used as a libraryCard }
			I
Response Class			
Model Model Schema			
AuthenticatedPatronV10) {		
	onal): Only available if patron exists in FBS a	nd was succesfully authenticated	
	ring) = ['VALID' or 'INVALID' or 'LOANER_LO	-	
- 'VALID': successfully at			
	ser is not known in the system, or an invalio	d combination of authentication para	meters has been used.
- 'LOANER_LOCKED_OUT	ר': user has been blocked temporary becaus	se of too many failed login attempts	
}			
PatronV9 {			
birthday (string, option	al),		
secondaryAddress (Ad	dressV3, <i>optional</i>),		
preferredLanguage (st	ring, optional): Language in which the patro	on prefers the communication with th	e library to take place,
preferredPickupBranc	h (string): ISIL of preferred pickup branch,		
address (AddressV3, <i>op</i>			
· ·	al): If not set then the patron is not on hold,		
	on identifier to be used in subsequent serv	ice calls involving the patron,	
guardianVisibility (boo			
receiveEmail (boolean)			
	kStatus], <i>optional</i>): A list of block statuses -	•	value is empty or null,
•	ta (boolean, <i>optional</i>): Patron consent to ke	eep historical loans,	
receiveSms (boolean),	onal): Tags associated with the patron. Can	he used to e.g. give permissions to pa	trop in ovtornal raccourses
	<pre>/nchronizationConsent (boolean),</pre>	be used to e.g. give permissions to pa	tron in external ressources.,
emailAddress (string, c			
	,	ols that the patron want to receive no	tification on. SMS and EMAIL are not included
phoneNumber (string,			
name (string, <i>optional</i>),			
receivePostalMail (boo			

defaultInterestPeriod (integer): Length of default interest period in days,
interests (array[string], optional),
resident (boolean): True if the user is resident in the same municipality as the library

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

Response Content Type application/json 🗸

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	
403	the guardian does not have access to update the patron details	

/external/{agencyid}/patrons/{patronid}/identifiers/block/v1

Block identifier for the given patron.

PUT

Implementation Notes

The identifier for the given patron is blocked

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	agencyId ISIL of the agency (e.g. DK-761500)	path	string
patronid	the id of a given patron	path	integer
blockIdentifier	Identifier to be blocked	body	Model Model Schema
			BlockIdentifierV1 { identifier (string)
			}

GET	/external/{agencyid}/patrons/{patronid}/identifiers/v1	Retrieve all non-expired identifiers for the given patron.

Implementation Notes

The identifiers are sorted by IdentifierType.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	agencyId ISIL of the agency (e.g. DK-761500)	path	string
patronid	the id of a given patron	path	integer

Response Class

Model Model Schema

LoanerIdentifierV1 {

expiryDate (string, optional),
identifier (string),
description (string, optional),

}

Swagger UI

identifierType (string), status (string)

Response Content Type application/json •

PUT /external/{agencyid}/patrons/{patronid}/setPassword/v1

Set patron password upon receiving a new password that satisfies the password requirements.

Implementation Notes

The response 400 bad request is sent back if the patron already has a password or if the new password is invalid. The response 404 not found is sent back if the patron is not found.

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron to be updated	path	integer
setPasswordRequestV1	the new password. Must be supplied and follow the password requirements	body	Model Model Schema SetPasswordRequestV1 { password (string): The new password of the user }

Response Class

Model Model Schema

integer

Response Content Type application/json >

Response Messages				
HTTP Status Code	Reason	Response Model		
400	bad request			
401	client unauthorized			
403	client forbidden			
404	patron not found			
200	Ok			

PUT	/external/{agencyid}/patrons/{patronid}/updateconsent/historicaldata/v2	Updates the patron's consent to keep historical loans.
-----	---	--

Implementation Notes

Withdrawing consent will permanently delete existing historical loans. Returns list of given consents for the patron.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer
updateConsent	payload containing true or false consent	body	Model Model Schema
			UpdateConsentV1 { consent (boolean): If consent is given or not.

}

Response Class

Model Model Schema

ConsentV1 {

consentType (string): If you encounter another type of consent please add it to this documentation.

ConsentTypes: - 'KEEP_HISTORICAL_LOAN_DATA' - 'SYNC_WITH_NATIONAL_REGISTRY' , consentDate (string): Date that consent was given. }				
Response Content T	ype application/json ►			
Response Message	25			
HTTP Status Code	Reason	Response Model		
200	ok			
400	bad request			
401	client unauthorized			
403	forbidden access to wrong agency			
404	no patron found for patronId			

PUT /external/{agencyid}/patrons/{patronid}/updateconsent/nationalregistry/v2

Updates the patron's consent to having their data synchronized with the National Patron Registry.

Implementation Notes

The data that is synchronized can be configured for the library. Returns list of given consents for the patron.

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer
updateConsent	payload containing true or false consent	body	Model Model Schema
			UpdateConsentV1 {

5/25, 1:11 PM		Swagger UI	
Parameter	Description		ta Туре
		}	consent (boolean): If consent is given or not.
Response Class			
Model Model Sche	ma		
ConsentV1 { consentType (strin	g): If you encounter another type of co	nsent please add it to this documentation.	
ConsentTypes: - 'KEEP_HISTORICAL - 'SYNC_WITH_NATION , consentDate (string) }			
Response Content T Response Messag HTTP Status Code	ype application/json ❤ es Reason	Response Model	
200	ok		
400	bad request		
401	client unauthorized		
403	forbidden access to wrong agency		
404	no patron found for patronId		
GET /external/-	{agencyid}/patrons/{patronid}/v	4	Returns the patron details (DEPRECATED).
Implementation N	lotes		
If a patron is blocke	d the reason is available as a code:		
calhost:8080/externalapidocs/			

71/175

- 'O': library card stolen

- 'U': exclusion

- 'F': extended exclusion

- 'S': blocked by self service automaton

- 'W': self created at website

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

AuthenticatedPatronV8 {

patron (PatronV7, *optional*): Only available if patron exists in FBS and was succesfully authenticated., **authenticateStatus** (string) = ['VALID' or 'INVALID' or 'LOANER_LOCKED_OUT']: AuthenticateStatus:

- 'VALID': successfully authenticated

- 'INVALID': either the user is not known in the system, or an invalid combination of authentication parameters has been used.

- 'LOANER_LOCKED_OUT': user has been blocked temporary because of too many failed login attempts

PatronV7 {

}

birthday (string, optional),

secondaryAddress (AddressV2, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV2, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

guardianVisibility (boolean),

receiveEmail (boolean),

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

receiveSms (boolean),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

emailAddress (string, optional),

notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included.,

phoneNumber (string, optional),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, optional): True if the user is allowed to create bookings.,

defaultInterestPeriod (integer): Length of default interest period in days,

interests (array[InterestV1], optional),

resident (boolean): True if the user is resident in the same municipality as the library

}

AddressV2 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

InterestV1 {

displayName (string): Display name of the interest, **name** (string): Name of the interest

}

Response Content Type application/json 🗸

Response Messages

5/5/25, 1:11 PM

Swagger UI

•••			
	HTTP Status Code	Reason	Response Model
	400	bad request	
	401	client unauthorized	

/external/{agencyid}/patrons/{patronid}/v8

Update information about the patron.

Implementation Notes

The name and address cannot be supplied by the client. If the configured person registry is not authorized to provide information about the patron, then the name and address will not be updated.

It is possible to either update just the pincode, update just some patron settings, or update both.

Multiple email addresses and phone numbers can be stored for a patron.

If multiple email addresses are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If multiple phone numbers are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If a patron is blocked the reason is available as a code:

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton 'W': self created at website

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron to be updated	path	integer
updatePatron	updated information about the patron	body	Model Model Schema

Parameter	Description	Parameter Type	Data Type
			UpdatePatronRequestV6 {
			patron (PatronSettingsV6, optional): Set this if patron
			details are to be changed,
			pincodeChange (PincodeChange, optional): Set this if
			pincode is to be changed
			}
			<pre>PatronSettingsV6 { preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place If left empty default library language will be used, notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV1], optional): Existing email addresses are overwritten with these values If left empty existing email addresses are deleted, preferredPickupBranch (string): ISIL-number of preferred pickup branch, onHold (Period, optional): If not set then the patron is not on hold, guardianVisibility (boolean), interests (array[string], optional): A list of interests of the patron., phoneNumbers (array[PhoneNumberV1], optional): Existing phonenumbers are overwritten with these value If left empty existing phonenumbers are deleted } EmailAddressV1 { emailAddress(string), receiveNotification (boolean) } Period { from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set to (string)</pre>
			PhoneNumberV1 {

5/5/25, 1:11 PM

5/5/25, 1:1 ⁻	1 PM		Swagger UI	
	Parameter	Description	Parameter Type	Data Type
				<pre>receiveNotification (boolean), phoneNumber (string) } PincodeChange { pincode (string): The new pincode for the libraryCard, libraryCardNumber (string): Identifies the libraryCard for which the pincode is to be changed. This can be either a physical card or the CPR number, that is used as a libraryCard }</pre>
	Response Class Model Model Scher	na		
	Response			
	Response Content Ty	ype application/json 🗸		
	Response Message	25		
	HTTP Status Code	Reason	Response Model	
	400	bad request		
	401	client unauthorized		
	404	patron not found		
	204	no content		

ut /external/{agencyid}/patrons/{patronid}/withoutIntegration/v1

Update information about the patron, as well as information about the patron's guardian (e.g., a financially responsible person).

Implementation Notes

Only the details provided in the payload are used for the update; no integration is used to retrieve patron data (e.g., from a person registry).

If the patron does not have a guardian, a new one will be created using the information provided.

Multiple email addresses and phone numbers can be stored for a patron.

If multiple email addresses are provided with "receiveNotification" set to true, only the first one in the list with this setting ("receiveNotification" = true) will be stored as the preferred email. The rest will be stored as non-preferred.

If multiple phone numbers are provided with "receiveNotification" set to true, only the first one in the list with this setting ("receiveNotification" = true) will be stored as the preferred phone number. The rest will be stored as non-preferred.

An email confirming the update will be sent if the patron has a guardian. If the update fails and the patron has a guardian or attempted to add one, an email will be sent to the guardian stating that the update failed.

If a patron is blocked the reason is available as a code:

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton 'W': self created at website

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron to be updated	path	integer
patronWithoutIntegrationRequest	The payload with information for the patron update	body	ModelModel SchemaUpdatePatronWithoutIntegrationRequestV1 {addressInfo (PatronAddressInfoV1, optional):Patron address information,bankInfo (PatronBankInfoV1, optional): Patronbank information,notificationProtocols (array[string], optional):Notification protocols that the patron want toreceive notification on. SMS and EMAIL are notincluded,contactInfo (PatronContactInfoV1, optional): Patroncontact information,guardianInfo (PatronGuardianInfoV1, optional):Patron guardian information,

Parameter	Description	Parameter Type	Data Type
			guardianVisibility (boolean, <i>optional</i>): Patron guardian visibility, generalInfo (PatronGeneralInfoV1): Patron general information, libraryInfo (PatronLibraryInfoV1): Patron library information
			}
			PatronAddressInfoV1 { secondaryAddress (AddressV3, optional): Secondary address of the person. The country code must be specified in a two-letter format (e.g. DK, SE, NO, DE), primaryAddress (AddressV3, optional): Primary address of the person. The country code must be specified in a two-letter format (e.g. DK, SE, NO, DE) }
			AddressV3 { country (string): Country,
			city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code,
			district (string): Dsitrict, subDistrict (string): Subdistrict
			}
			<pre>PatronBankInfoV1 { accountHolder (string, optional): The name of the account holder, iban (string, optional): The International Bank Account Number (IBAN) of the account, bic (string, optional): The Bank Identifier Code (BIC) of the account }</pre>
			PatronContactInfoV1 { emailAddresses (array[EmailAddressV1], optional): Patron's email addresses. If multiple email addresses are provided and have

Parameter	Description	Parameter Type	Data Type
Parameter	Description	Parameter Type	<pre>'receiveNotification': true, the first one is considered the primary email address, phoneNumbers (array[PhoneNumberV1], optional): Patron's phone numbers. If multiple phone numbers are provided and have 'receiveNotification': true, the first one is considered the primary phone number } EmailAddressV1 { emailAddress (string), receiveNotification (boolean) } PhoneNumberV1 { receiveNotification (boolean), phoneNumber (string) } PatronGuardianInfoV1 { country (string, optional): Country of the guardian. The country code must be specified in a two-letter format (e.g. DK, SE, NO, DE), address (string, optional): Gender of the guardian. Allowed values are 'MALE', 'FEMALE', 'OTHER', 'UNKNOWN', mobilePhone (string, optional): Mobile phone number of the guardian, city (string, optional): City of the guardian, homePhone (string, optional): Home phone number of the guardian, postalCode (string, optional): Postal code of the guardian, name (string): Name of the guardian,</pre>
			 name (string): Name of the guardian, personIdentifier (string, optional): Person identifier of the guardian, workPhone (string, optional): Work phone number of the guardian, birthDate (string, optional): Birthdate of the guardian,
30/externalapidocs/			

Swaq	aer	ш	
Oway	yei	U.	

PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
			email (string): Email of the guardian. Must be valid
			}
			PatronGeneralInfoV1 {
			<pre>gender (string, optional): Gender of the person. Allowed values are 'MALE', 'FEMALE', 'OTHER', 'UNKNOWN', name (string): Name of the person, personIdentifier (string, optional): Person identifier, birthDate (string, optional): Birth date of the perso } PatronLibraryInfoV1 { preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place. If left empty default library language will be used, preferredPickupBranch (string): Preferred pickup branch of the patron }</pre>
Response Class			
Model Model Sche	ema		
Response			
Response Content Response Messag	Type application/json V		
		Deers and Marchal	
HTTP Status Code	Reason	Response Model	
204	No content		
400	Bad Request - The request is malformed or contains invalid data.		

5/5/25, 1:11 PM

Swagger UI

1.11	PIVI		Swagger Or
	HTTP Status Code	Reason	Response Model
	403	Client Forbidden - The client does not have access to the specified agency or the patron's branch.	
	404	Patron not found	

POST /external/{agencyid}/patrons/authenticate/v10

Authenticates a patron and returns a patron id, patron type and authentication status.

Implementation Notes

The returned patronId must be used by all subsequent service calls made on behalf of that patron. The full patron can be fetched with the operation corresponding to the patron type using the patron id and patron type.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
authenticationRequest	credentials for patron to be authenticated	body	Model Model Schema AuthenticationRequestV2 { pincode (string, optional): The pincode that belongs to the libraryCardNumber in plain text. Either pincode or password must be supplied. Supplying both is not allowed., password (string, optional): The password that belongs to the libraryCardNumber in plain text. Either pincode or password must be supplied. Supplying both is not allowed., libraryCardNumber (string): Identifies a libraryCard. This can be either a physical card or a CPR number that is used as a libraryCard
Response Class			
Model Model Schema			

PatronAuthenticationResponse {

5/25, 1:11 PM		Swagger UI
authenticateStatu - 'VALID': successfu		D' or 'LOANER_LOCKED_OUT']: AuthenticateStatus:
	•	stem, or an invalid combination of authentication parameters has been used. temporary because of too many failed login attempts
, patronId (integer),		
patronType (string	;) = ['PERSON' or 'COMPANY' or	'LIBRARY' or 'GROUP']
}		
Response Content	Type application/json 🗸	
Response Messag	jes	
HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

POST	/external/{agencyid}/patrons/authenticate/v5	Authenticates a patron and returns the patron details (DEPRECATED).
------	--	---

Implementation Notes

The returned patron details includes a patronId that has to be used by all subsequent service calls made on behalf of that patron.

If a patron is blocked the reason is available as a code: - 'O': library card stolen - 'U': exclusion

- 'F': extended exclusion
- 'S': blocked by self service automaton 'W': self created at website

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string

Darameter	Description	Daramater Tuna	Data Tura
Parameter	Description	Parameter Type	Data Type
authenticationRequest	credentials for patron to be authenticated	body	Model Model Schema AuthenticationRequest { pincode (string): The pincode that belongs to libraryCardNumber in plain text, libraryCardNumber (string): Identifies a libre This can be either a physical card or a CPR nu that is used as a libraryCard
			}
Response Class			
Model Model Schema			
AuthenticatedPatronV5 {			
- 'VALID': successfully authors - 'INVALID': either the user	g) = ['VALID' or 'INVALID' or 'LOANER_LOCKED_C enticated is not known in the system, or an invalid combi iser has been blocked temporary because of too	nation of authentication para	meters has been used.
- 'VALID': successfully authors - 'INVALID': either the user	enticated is not known in the system, or an invalid combi	nation of authentication para	meters has been used.
 'VALID': successfully authors 'INVALID': either the user 'LOANER_LOCKED_OUT': u 	enticated is not known in the system, or an invalid combi iser has been blocked temporary because of too	nation of authentication para	meters has been used.
 'VALID': successfully authors 'INVALID': either the user 'LOANER_LOCKED_OUT': u PatronV4 { 	enticated is not known in the system, or an invalid combi iser has been blocked temporary because of too	nation of authentication para	meters has been used.
 'VALID': successfully authors 'INVALID': either the user 'LOANER_LOCKED_OUT': u PatronV4 { birthday (string, optional), secondaryAddress (Address) preferredLanguage (string) 	enticated is not known in the system, or an invalid combi iser has been blocked temporary because of too	nation of authentication para	
 'VALID': successfully authors 'INVALID': either the user 'LOANER_LOCKED_OUT': u PatronV4 { birthday (string, optional), secondaryAddress (Address) preferredLanguage (string) 	enticated is not known in the system, or an invalid combi iser has been blocked temporary because of too ssV2, <i>optional</i>), g, <i>optional</i>): Language in which the patron prefe string): ISIL of preferred pickup branch,	nation of authentication para	
 'VALID': successfully authors 'INVALID': either the user 'LOANER_LOCKED_OUT': u PatronV4 { birthday (string, optional), secondaryAddress (Address preferredLanguage (strings preferredPickupBranch (strings) address (AddressV2, optional): 	enticated is not known in the system, or an invalid combi iser has been blocked temporary because of too ssV2, <i>optional</i>), g, <i>optional</i>): Language in which the patron prefe string): ISIL of preferred pickup branch, <i>nal</i>), If not set then the patron is not on hold,	nation of authentication para o many failed login attempts rs the communication with t	
 'VALID': successfully authors 'INVALID': either the user 'LOANER_LOCKED_OUT': u PatronV4 { birthday (string, optional), secondaryAddress (Address preferredLanguage (strings preferredPickupBranch (strings) address (AddressV2, optional): 	enticated is not known in the system, or an invalid combi iser has been blocked temporary because of too ssV2, <i>optional</i>), g, <i>optional</i>): Language in which the patron prefe string): ISIL of preferred pickup branch, <i>nal</i>),	nation of authentication para o many failed login attempts rs the communication with t	
 'VALID': successfully authors 'INVALID': either the user 'LOANER_LOCKED_OUT': u PatronV4 { birthday (string, optional), secondaryAddress (Address (Address (Address (Address (AddressV2, optional), address (AddressV2, optional): address (AddressV2, opt	enticated is not known in the system, or an invalid combi iser has been blocked temporary because of too ssV2, <i>optional</i>), g, <i>optional</i>): Language in which the patron prefe string): ISIL of preferred pickup branch, <i>nal</i>), If not set then the patron is not on hold,	nation of authentication para o many failed login attempts rs the communication with t involving the patron,	ne library to take place,
 'VALID': successfully authors 'INVALID': either the user 'LOANER_LOCKED_OUT': u PatronV4 { birthday (string, optional), secondaryAddress (Address preferredLanguage (strings preferredPickupBranch (secondaress (Address V2, optional): patronId (integer): Patron receiveEmail (boolean), blockStatus (array[BlockStatus (boolean), 	enticated is not known in the system, or an invalid combi- iser has been blocked temporary because of too ssv2, <i>optional</i>), g, <i>optional</i>): Language in which the patron prefe- string): ISIL of preferred pickup branch, <i>nal</i>), If not set then the patron is not on hold, identifier to be used in subsequent service calls status], <i>optional</i>): A list of block statuses - if the p	nation of authentication para o many failed login attempts rs the communication with t involving the patron,	ne library to take place,
 'VALID': successfully authors 'INVALID': either the user 'LOANER_LOCKED_OUT': u PatronV4 { birthday (string, optional), secondaryAddress (Address preferredLanguage (strings preferredPickupBranch (strings) address (AddressV2, optional): patronId (integer): Patron receiveEmail (boolean), blockStatus (array[BlockStstrings), emailAddress (string, optional), 	enticated is not known in the system, or an invalid combi- user has been blocked temporary because of too ssV2, <i>optional</i>), g, <i>optional</i>): Language in which the patron prefe- string): ISIL of preferred pickup branch, <i>nal</i>), If not set then the patron is not on hold, identifier to be used in subsequent service calls satus], <i>optional</i>): A list of block statuses - if the p <i>onal</i>),	nation of authentication para o many failed login attempts rs the communication with t involving the patron,	ne library to take place,
 'VALID': successfully auther 'INVALID': either the user 'LOANER_LOCKED_OUT': u PatronV4 { birthday (string, optional), secondaryAddress (Addree preferredLanguage (string preferredPickupBranch (string address (AddressV2, option onHold (Period, optional): 1 patronId (integer): Patron receiveEmail (boolean), blockStatus (array[BlockStt receiveSms (boolean), emailAddress (string, option 	enticated is not known in the system, or an invalid combi- user has been blocked temporary because of too ssv2, <i>optional</i>), g, <i>optional</i>): Language in which the patron prefe- string): ISIL of preferred pickup branch, <i>nal</i>), If not set then the patron is not on hold, identifier to be used in subsequent service calls satus], <i>optional</i>): A list of block statuses - if the p <i>onal</i>), <i>tional</i>),	nation of authentication para o many failed login attempts rs the communication with t involving the patron,	ne library to take place,

defaultInterestPer	iod (integer): Length of default	nterest period in days,			
resident (boolean):	True if the user is resident in th	same municipality as the	e library		
}					
AddressV2 {					
country (string): Co	untry,				
city (string): City,					
street (string): Stree					
coName (string): c/	o name,				
postalCode (string)	Postal code				
}					
Period {					
from (string, option	<i>al</i>): Open-ended if not set,				
to (string, <i>optional</i>):	Open-ended if not set				
}					
BlockStatus {					
blockedReason (str	ing): Reason code for block,				
blockedSince (strin	g),				
message (string): M	essage about block				
}					
Response Content T	ype application/json 🗸				
Response Message	25				
HTTP Status Code	Reason		Response Model		
400	bad request				
401	client unauthorized				

POST	/external/{agencyid}/patrons/authenticate/v9	Authenticates a patron and returns a patron id, patron type and authentication status.
------	--	--

Implementation Notes

The returned patronId must be used by all subsequent service calls made on behalf of that patron. The full patron can be fetched with the operation corresponding to the patron type using the patron id and patron type.

Parameters				
Parameter	Description		Parameter Type	Data Type
agencyid	ISIL of the	agency (e.g. DK-761500)	path	string
authenticationRe	quest credentials authentica	s for patron to be	body	Model Model Schema
	uutiteittee	iii uu		AuthenticationRequest { pincode (string): The pincode that belongs to the libraryCardNumber in plain text, libraryCardNumber (string): Identifies a libraryCard This can be either a physical card or a CPR number that is used as a libraryCard }
Response Class				
Nodel Model Sch Model Addel Sch				
Model Model Sch PatronAuthenticati authenticateStat - 'VALID': successfu - 'INVALID': either - 'LOANER_LOCKED , patronId (integer)	onResponse { us (string) = ['VALID' or 'INVA illy authenticated the user is not known in the p_OUT': user has been blocke	system, or an invalid combi ed temporary because of too	OUT']: AuthenticateStatus: nation of authentication par o many failed login attempts	
Model Model Sch PatronAuthenticati authenticateStat - 'VALID': successfu - 'INVALID': either - 'LOANER_LOCKED , patronId (integer) patronType (string	onResponse { us (string) = ['VALID' or 'INVA illy authenticated the user is not known in the p_OUT': user has been blocke	system, or an invalid combi ed temporary because of too	nation of authentication par	
Model Model Sch PatronAuthenticati authenticateStat - 'VALID': successfu - 'INVALID': either - 'LOANER_LOCKED , patronId (integer) patronType (string	onResponse { us (string) = ['VALID' or 'INVA Illy authenticated the user is not known in the p_OUT' : user has been blocked $p_g) = ['PERSON' or 'COMPANY'' Type application/json \checkmark$	system, or an invalid combi ed temporary because of too	nation of authentication par	
Model Model Sch PatronAuthenticati authenticateStat - 'VALID': successfu - 'INVALID': either - 'LOANER_LOCKED , patronId (integer) patronType (string Response Content	onResponse { us (string) = ['VALID' or 'INVA Illy authenticated the user is not known in the p_OUT' : user has been blocked $p_g) = ['PERSON' or 'COMPANY'' Type application/json \checkmark$	system, or an invalid combi ed temporary because of too	nation of authentication par	
Model Model Sch PatronAuthenticati authenticateStat - 'VALID': successfu - 'INVALID': either - 'LOANER_LOCKED , patronId (integer) patronType (string Response Content Response Messag	onResponse { us (string) = ['VALID' or 'INVA illy authenticated the user is not known in the p_OUT': user has been blocke g) = ['PERSON' or 'COMPANY' Type application/json ~	system, or an invalid combi ed temporary because of too	nation of authentication par o many failed login attempts	

GET

/external/{agencyid}/patrons/company/{patronid}/v2

Implementation Notes

If a patron is blocked the reason is available as a code:

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton- 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

CompanyPatronV2 {

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

contactPerson (string),

phoneNumbers (array[PhoneNumberV1], optional),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV1], optional),

companyIdentifier (string),
name (string, optional),
receivePostalMail (boolean),
allowBookings (boolean, optional): True if the user is allowed to create bookings.,
defaultInterestPeriod (integer): Length of default interest period in days,
interests (array[InterestV1], optional)

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean),
phoneNumber (string)

}

EmailAddressV1 {

emailAddress (string), receiveNotification (boolean)

}

InterestV1 {

displayName (string): Display name of the interest,

}

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

/external/{agencyid}/patrons/company/{patronid}/v3 GET

Returns the patron details for company patron

Implementation Notes

If a patron is blocked the reason is available as a code:

- 'O': library card stolen 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

CompanyPatronV3 {

secondaryAddress (AddressV3, optional), preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place, preferredPickupBranch (string): ISIL of preferred pickup branch, address (AddressV3, optional), onHold (Period, optional): If not set then the patron is not on hold, **patronId** (integer): Patron identifier to be used in subsequent service calls involving the patron, **blockStatus** (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null, contactPerson (string), phoneNumbers (array[PhoneNumberV1], optional), tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources. notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV2], optional), companyIdentifier (string), name (string, optional), receivePostalMail (boolean), allowBookings (boolean, optional): True if the user is allowed to create bookings., defaultInterestPeriod (integer): Length of default interest period in days, interests (array[InterestV1], optional)

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}			
PhoneNumberV1 {			
receiveNotification phoneNumber (strin			
<pre>phoneNumber (sum }</pre>	ig)		
۶ EmailAddressV2 {			
emailAddress (string	σ)		
receiveNotification	-		
verified (boolean)			
}			
InterestV1 {			
displayName (strinន្	y): Display name of the interest,		
name (string): Name	e of the interest		
}			
Response Content Ty	pe application/json V		
Response Message	S		
HTTP Status Code	Reason	Response Model	
HTTP Status Code	Reason bad request	Response Model	
		Response Model	
400	bad request	Response Model	
400 401	bad request	Response Model	Returns the patron details for company patro

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

CompanyPatronV4 {

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

contactPerson (string),

phoneNumbers (array[PhoneNumberV1], optional),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

notificationProtocols (array[string], *optional*): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., **emailAddresses** (array[EmailAddressV2], *optional*),

companyIdentifier (string),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, optional): True if the user is allowed to create bookings.,

defaultInterestPeriod (integer): Length of default interest period in days,

patronNote (string, optional): Note to patron,

interests (array[InterestV1], optional)

}

AddressV3 {

country (string): Country,

city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean),
phoneNumber (string)

}

EmailAddressV2 {

emailAddress (string), receiveNotification (boolean), verified (boolean)

}

InterestV1 {

displayName (string): Display name of the interest, **name** (string): Name of the interest

}

Response Content Type application/json >

Response Messages

 1 101		Gwaggel Ol
HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

GET /external/-	{agencyid}/patrons/consent/{patronid}/v	1	Get the list of given consents for a patr
Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	owner of the consents	path	integer
Response Class			
Model Model Sche	ma		
ConsentTypes: - 'KEEP_HISTORICAL - 'SYNC_WITH_NATI(,			
Response Content T	ype application/json 🗸		
Response Messag	es		
HTTP Status Code	Reason	Response Model	
200	ok		
401	client unauthorized		

5/5/25,	1:11	ΡM
---------	------	----

404	no patron found for patronId	
HTTP Status Code	Reason	Response Model
PM		Swagger UI

СГТ	/external/(agencyid)/natrons/group/(natronid)/v2
GET	/external/{agencyid}/patrons/group/{patronid}/v2

Returns the patron details for a group patron (DEPRECATED).

Implementation Notes

If a patron is blocked the reason is available as a code:

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton- 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

GroupPatronV2 {

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

contactPerson (string),

phoneNumbers (array[PhoneNumberV1], optional),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included.,

emailAddresses (array[EmailAddressV1], optional),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, optional): True if the user is allowed to create bookings.,

defaultInterestPeriod (integer): Length of default interest period in days,

interests (array[InterestV1], optional)

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean),
phoneNumber (string)

}

EmailAddressV1 { emailAddress (string),

	receiveNotification (boolean)				
}					
InterestV1 {					
displayName (string): Display name of the interest,				
name (string): Name	of the interest				
}					
Response Content Ty	pe application/json V				
Response Messages	Response Messages				
HTTP Status Code	Reason	Response Model			
400	bad request				
401	client unauthorized				

GET	/external/{agencyid}/patrons/group/{patronid}/v3	Returns the patron details for a group patron
-----	--	---

Implementation Notes

If a patron is blocked the reason is available as a code: - 'O': library card stolen - 'U': exclusion

- 'F': extended exclusion
 'S': blocked by self service automaton
 'W': self created at website
- 'E': maximum amount of allowed debt exceeded

- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

GroupPatronV3 {

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

contactPerson (string),

phoneNumbers (array[PhoneNumberV1], optional),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

notificationProtocols (array[string], *optional*): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., **emailAddresses** (array[EmailAddressV2], *optional*),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, optional): True if the user is allowed to create bookings.,

defaultInterestPeriod (integer): Length of default interest period in days,

interests (array[InterestV1], optional)

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

```
5/5/25, 1:11 PM
```

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean),
phoneNumber (string)

}

EmailAddressV2 {

emailAddress (string), receiveNotification (boolean), verified (boolean)

}

InterestV1 {

displayName (string): Display name of the interest, **name** (string): Name of the interest

}

Response Content Type application/json >

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

GET

/external/{agencyid}/patrons/group/{patronid}/v4

Returns the patron details for a group patron

Implementation Notes

If a patron is blocked the reason is available as a code: - 'O': library card stolen

- 'U': exclusion
- 'F': extended exclusion

- 'S': blocked by self service automaton- 'W': self created at website

- 'E': maximum amount of allowed debt exceeded

- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

GroupPatronV4 {

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

contactPerson (string),

phoneNumbers (array[PhoneNumberV1], optional),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV2], optional),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, optional): True if the user is allowed to create bookings.,

defaultInterestPeriod (integer): Length of default interest period in days,

patronNote (string, optional): Note to patron,

interests (array[InterestV1], optional)

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean),
phoneNumber (string)

}

EmailAddressV2 {

emailAddress (string), receiveNotification (boolean), verified (boolean)

}

InterestV1 {

displayName (string): Display name of the interest, **name** (string): Name of the interest

}

Response Content Type application/json 🗸

Response Messages			
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		

Implementation Notes

If the corresponding agency configuration through /external/v1/{agencyid}/configuration/loaner.guardianVisibility.enabled is not enabled, then response message 403 will be sent back.

If there are no patrons found, that have the guardian visibility enabled, for the given guardian person identifier (e.g. CPR-number), then response message 404 will be sent back.

The returned patrons details includes a patronId that has to be used by all subsequent service calls made on behalf of that patron. Note: This method can only be used for patrons who are people, and not e.g. Companies or Libraries.

If a patron is blocked the reason is available as a code:

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton
 'W': self created at website

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
guardianPersonId		body	string

Response Class

Model Model Schema

AuthenticatedPatronV10 {

patron (PatronV9, optional): Only available if patron exists in FBS and was succesfully authenticated.,

authenticateStatus (string) = ['VALID' or 'INVALID' or 'LOANER_LOCKED_OUT']: AuthenticateStatus:

- 'VALID': successfully authenticated

- 'INVALID': either the user is not known in the system, or an invalid combination of authentication parameters has been used.

- 'LOANER_LOCKED_OUT': user has been blocked temporary because of too many failed login attempts

}

PatronV9 {

birthday (string, optional),

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

guardianVisibility (boolean),

receiveEmail (boolean),

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

keepLoanHistoricalData (boolean, optional): Patron consent to keep historical loans,

receiveSms (boolean),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

hasNationalRegistrySynchronizationConsent (boolean),

emailAddress (string, optional),

notificationProtocols (array[string], *optional*): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., **phoneNumber** (string, *optional*),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, optional): True if the user is allowed to create bookings.,

defaultInterestPeriod (integer): Length of default interest period in days,

interests (array[string], optional),

resident (boolean): True if the user is resident in the same municipality as the library

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number,

coName (string): c/o name,	
<pre>postalCode (string): Postal code,</pre>	
district (string): Dsitrict,	
subDistrict (string): Subdistrict	
}	

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block
}

Response Content Type application/json V

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	
403	the guardians do not have access to patron details on this agency	
404	no patrons found that have the guardian as a financia resposible or that have the guardian visibility enabled	

GET /external/{agencyid}/patrons/interests/v1

Returns a list of interests that are configured for the agency.

Implementation Notes

Interests can be added to patrons.

Parameters

1 PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
agencyid	agencyId ISIL of the agency (e.g. DK-761500)	path	string
Response Class			
Model Model Sch	ema		
PatronInterestsV1 interests (Map)	(
}			
Response Content	Type application/json 🗸		
GET /external	/{agencyid}/patrons/languages/v1	Get the list	t of languages supported as preferred language of a patron.
Implementation	Notes		
Returns an array o	f string representation of the language codes.		
Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
Response Class			
Model Model Sch	ema		
array[string]			
Response Content	Type application/json 🗸		
Response Messag			
HTTP Status Code	Reason	Response Model	
400	bad request	,	
401	client unauthorized		

/external/{agencvid}/patrons/library/{patronid}/v2 GET

Implementation Notes

If a patron is blocked the reason is available as a code:

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton- 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

LibraryPatronV2 {

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

phoneNumbers (array[PhoneNumberV1], optional),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included.,

emailAddresses (array[EmailAddressV1], optional),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, *optional*): True if the user is allowed to create bookings., **defaultInterestPeriod** (integer): Length of default interest period in days, **interests** (array[InterestV1], *optional*), **isil** (string)

ISII (S }

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean),
phoneNumber (string)

}

EmailAddressV1 {

emailAddress (string), receiveNotification (boolean)

}

InterestV1 {

displayName (string): Display name of the interest, **name** (string): Name of the interest

}		
Response Content T	ype application/json 🗸	
Response Message	25	
HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

	GET	/external/{agencyid}/patrons/library/{patronid}/v3
--	-----	--

Returns the patron details for a library patron

Implementation Notes

If a patron is blocked the reason is available as a code: - 'O': library card stolen

- 'U': exclusion

- 'F': extended exclusion
 'S': blocked by self service automaton
 'W': self created at website
 'E': maximum amount of allowed debt exceeded
- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

LibraryPatronV3 {

secondaryAddress (AddressV3, optional),

Swagger UI **preferredLanguage** (string, optional): Language in which the patron prefers the communication with the library to take place, preferredPickupBranch (string): ISIL of preferred pickup branch, address (AddressV3, optional), onHold (Period, optional): If not set then the patron is not on hold, **patronId** (integer): Patron identifier to be used in subsequent service calls involving the patron, **blockStatus** (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null, phoneNumbers (array[PhoneNumberV1], optional), tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources., notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV2], optional), name (string, optional), receivePostalMail (boolean), allowBookings (boolean, optional): True if the user is allowed to create bookings., defaultInterestPeriod (integer): Length of default interest period in days, interests (array[InterestV1], optional), isil (string): ISIL of the library }

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotificatio	receiveNotification (boolean),					
phoneNumber (string)						
}	}					
EmailAddressV2 {	EmailAddressV2 {					
emailAddress (strin	ng),					
receiveNotificatio	n (boolean),					
verified (boolean)						
}						
InterestV1 {						
displayName (strin	g): Display name of the interest,					
name (string): Nam	e of the interest					
}	}					
Response Content T	ype application/json 🗸					
Despense Messag						
Response Message						
HTTP Status Code	Reason	Response Model				
400	bad request					
401	client unauthorized					
GET /external/{	[agencyid}/patrons/library/{patronid}/v4		Returns the patron details for a library patron			

Implementation Notes

If a patron is blocked the reason is available as a code: - 'O': library card stolen - 'U': exclusion

- 'F': extended exclusion
- 'S': blocked by self service automaton 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

	5/5/25,	1:11	ΡM
--	---------	------	----

patronid	patron id of the patron	path	integer
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
Parameter	Description	Parameter Type	Data Type
Parameters			

Response Class

Model Model Schema

LibraryPatronV4 {

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

phoneNumbers (array[PhoneNumberV1], optional),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV2], optional),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, optional): True if the user is allowed to create bookings.,

defaultInterestPeriod (integer): Length of default interest period in days,

patronNote (string, optional): Note to patron,

interests (array[InterestV1], optional),

isil (string): ISIL of the library

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code,

district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean), phoneNumber (string)

}

EmailAddressV2 {

```
emailAddress (string),
receiveNotification (boolean),
verified (boolean)
```

}

InterestV1 {

displayName (string): Display name of the interest, **name** (string): Name of the interest

}

Response Content Type application/json >

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

PM			Swagger UI	
GET	/external/{	agencyid}/patrons/notificationproto	ocols/v1 Get the li	st of notification protocols that is possible to set for a patr
Daran	neters			
		Description	Deverenter Turc	Data Tura
Param	neter	Description	Parameter Type	Data Type
ageno	cyid	ISIL of the agency (e.g. DK-761500)	path	string
Respo	onse Class			
Notifie	l Model Scher	{	l that is suitable to be shown to the	patron
Notifio disp	l Model Scher cationProtocol layName (string		l that is suitable to be shown to the	e patron.,
Notific disp nam }	l Model Scher cationProtocol layName (string ne (string): Name	{ g): Display name of the notification protoco	l that is suitable to be shown to the	e patron.,
Notific disp nam } Respo	l Model Scher cationProtocol layName (string ne (string): Name	{ g): Display name of the notification protoco e of the notification protocol ype application/json ❤	l that is suitable to be shown to the	e patron.,
Notific disp nam } Respo Respo	I Model Scher cationProtocol olayName (string ne (string): Name	{ g): Display name of the notification protoco e of the notification protocol ype application/json ❤	l that is suitable to be shown to the Response Model	patron.,
Notific disp nam } Respo Respo	I Model Scher cationProtocol layName (string ne (string): Name onse Content Ty onse Message	{ g): Display name of the notification protocol e of the notification protocol ype application/json ✓		e patron.,

POST /external/{agencyid}/patrons/passwordReset/triggerEmail/v1

Step 1 of 2 for the reset password flow, for when the patron has forgotten their password.

Implementation Notes

Sends a reset password notification to the patron with the provided email identifier. The notification is sent to the patron using all the notification protocols that are enabled for agency and the patron.

Notification templates exposes agencyid as \${(data.agency.agencyISIL)} and uuid as \${(data.uuid)}. These are needed when calling step 2: /passwordReset/updatePassword/v1.

E.g. the notification template can contain a link to your website https://example.com/resetPassword/\${(data.agency.agencyISIL)}/\${(data.uuid)}.

The resulting link could be something like https://example.com/resetPassword/DK-761500/550e8400-e29b-41d4-a716-446655440000.

This way you can get the agencyid and uuid and use it in step 2.

If more than one patron uses the supplied email, then the patron identifier must also be supplied to identify the patron.

The response is successful, if a notification was successfully sent to the patron to reset their password. The response 204 will be sent back

The response 400 is sent back if: the email address has an invalid format multiple patrons found for email address The response 401 is sent back if the client is unauthorized. The response 404 is sent back if no matching email is found, if no patron using the email is found, or, if the notification could not be sent to patron using any of the configured protocols.

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
resetPasswordRequestV1	request to send password reset email	body	Model Schema
			ResetPasswordRequestV1 { patronIdentifier (string, optional): An identifier for the patron. This can e.g. be the loaner number or the library card number. This must be supplied in case the email is not unique. If the email is unique this field will be ignored, email (string): The email of the patron. Must be the preferred email of the patron }
Response Class			
Model Model Schema			
Response			
Response Content Type applica	ation/json 🗸		
Response Messages			

HTTP S	itatus Code	Reason	Response Model
204		No content	
400		bad request - invalid email format or no email found that matches request or multiple patrons found for email address or no preferred email found for patron	
401		client unauthorized	
404		no patron found for emailAddress or no email found that matches request	
500		internal server error - email could not be sent	
POST	/external/{a	gencyid}/patrons/passwordReset/updatePassv	vord/v1

/external/{agencyid}/patrons/passwordReset/updatePassword/v1

Step 2 of 2 for the reset password flow, for when the patron has forgotten their password.

Implementation Notes

Using the UUID and the AgencyISIL provided through step 1, reset the password of the patron.

Can be used whether the patron has a password set or not.

The response is successful, if the password was successfully updated.

If the supplied password is invalid, blank or too long, then response message 400 will be sent back.

If the supplied UUID is invalid, blank, not a valid format, not found, or, expired then response message 400 will be sent back.

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
updatePassword	Request to update patron password	body	Model Model Schema
			UpdatePasswordRequestV1 { password (string): The new password for the patron,

1		Swagger UI	
Parameter	Description	Parameter Type	Data Type
			uuid (string): UUID provided to patron in notification
			when resetting password
			}
esponse Class			
odel Model Sc	hema		
esponse			
esponse Conten	t Type application/json ✔		
esponse Messa	ages		
ITTP Status Code	Reason	Response Model	
00	bad request - invalid format for password or UUID		
01	client unauthorized		
.03	forbidden access to wrong agency		
rt /externa	al/{agencyid}/patrons/person/{patronid}/v2		Returns the patron details for a person patron (DEPRECA
/externa	agencyiu)/patrons/person/{patronu}/vz		Returns the patron details for a person patron (DEPRECA
nplementation	ו Notes		
a natron is bloc	ked the reason is available as a code:		
O': library card	stolen		
U': exclusion	clusion		
	elf cervice automaton		
F': extended exc			
F': extended exe S': blocked by se W': self created	at website		
'F': extended exc 'S': blocked by so 'W': self created	at website nount of allowed debt exceeded		

any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

5/5/25,

11 PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer
Response Class			
Model Model Sc	hema		
PersonPatronV2 {			
birthday (string,	optional),		
	ess (AddressV3, <i>optional</i>),		
preferredLangu	age (string, optional): Language in which the patron p	prefers the communication v	with the library to take place,
preferredPicku	pBranch (string): ISIL of preferred pickup branch,		
address (Addres	sV3, optional),		
	<i>optional</i>): If not set then the patron is not on hold,		
• • •	r): Patron identifier to be used in subsequent service	calls involving the patron,	
guardianVisibili			
	ay[BlockStatus], <i>optional</i>): A list of block statuses - if t	the patron is not blocked the	en this value is empty or null,
	(array[PhoneNumberV1], <i>optional</i>),		
	g], <i>optional</i>): Tags associated with the patron. Can be		-
		that the patron want to rece	eive notification on. SMS and EMAIL are not included.,
	(array[EmailAddressV1], <i>optional</i>),		
name (string, op			
receivePostalM	boolean, <i>optional</i>): True if the user is allowed to crea	to bookings	
	Period (integer): Length of default interest period in d		
	InterestV1], optional),	uays,	
	an): True if the user is resident in the same municipali	ity as the library	
}	in, much the user is resident in the sume mulleput	ty us the horury	
, AddressV3 {			
country (string):	Country		
city (string): City	-		

street (string): Street and number,

coName (string): c/o name,

postalCode (string): Postal code,

district (string): Dsitrict,

subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean),
phoneNumber (string)

}

EmailAddressV1 {

emailAddress (string), receiveNotification (boolean)

}

InterestV1 {

displayName (string): Display name of the interest, **name** (string): Name of the interest

}

Response Content Type application/json >

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

GET /external/{agencyid}/patrons/person/{patronid}/v3

Returns the patron details for a person patron

Implementation Notes

If a patron is blocked the reason is available as a code:

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton
- 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

PersonPatronV3 {

birthday (string, optional),

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

guardianVisibility (boolean),

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

phoneNumbers (array[PhoneNumberV1], optional),

tags (array[string], *optional*): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources., **notificationProtocols** (array[string], *optional*): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., **emailAddresses** (array[EmailAddressV2], *optional*), name (string, optional),
receivePostalMail (boolean),
allowBookings (boolean, optional): True if the user is allowed to create bookings.,
defaultInterestPeriod (integer): Length of default interest period in days,
interests (array[InterestV1], optional),
resident (boolean): True if the user is resident in the same municipality as the library

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set,
to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block, blockedSince (string), message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean),
phoneNumber (string)

}

EmailAddressV2 {

emailAddress (string), receiveNotification (boolean), verified (boolean)

}

InterestV1 {

displayName (string): Display name of the interest,	
name (string): Name of the interest	

Response Content Type application/json 🗸

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

/external/{agencyid}/patrons/person/{patronid}/v4 Returns the patron details for a person patron GET

Implementation Notes

If a patron is blocked the reason is available as a code: - 'O': library card stolen

- 'U': exclusion

- 'F': extended exclusion
 'S': blocked by self service automaton
 'W': self created at website
 'E': maximum amount of allowed debt exceeded
- 'D': deceased

The codes are informational, and can be used for looking up end user messages by the client system. However, the list is subject to change at any time, so any unexpected values should be interpreted as 'other reason'.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	patron id of the patron	path	integer

Response Class

Model Model Schema

PersonPatronV4 { birthday (string, optional), secondaryAddress (AddressV3, optional), preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place, preferredPickupBranch (string): ISIL of preferred pickup branch, address (AddressV3, optional), onHold (Period, optional): If not set then the patron is not on hold, **patronId** (integer): Patron identifier to be used in subsequent service calls involving the patron, guardianVisibility (boolean), **blockStatus** (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null, phoneNumbers (array[PhoneNumberV1], optional), tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources. notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV2], optional), name (string, optional), receivePostalMail (boolean), allowBookings (boolean, optional): True if the user is allowed to create bookings., defaultInterestPeriod (integer): Length of default interest period in days, patronNote (string, optional): Note to patron, interests (array[InterestV1], optional), resident (boolean): True if the user is resident in the same municipality as the library } AddressV3 { country (string): Country, city (string): City, street (string): Street and number,

coName (string): c/o name,

postalCode (string): Postal code,

district (string): Dsitrict,

subDistrict (string): Subdistrict

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

BlockStatus {

blockedReason (string): Reason code for block,
blockedSince (string),
message (string): Message about block

}

PhoneNumberV1 {

receiveNotification (boolean), phoneNumber (string)

}

EmailAddressV2 {

emailAddress (string), receiveNotification (boolean), verified (boolean)

}

InterestV1 {

displayName (string): Display name of the interest, **name** (string): Name of the interest

}

Response Content Type application/json

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

POST /external/{agencyid}/patrons/preauthenticated/patronId/v4

Returns a patron id, patron type and authentication status for a patron that the client has pre-authenticated using a third party.

Implementation Notes

Patrons can be looked up based on their patronId. The returned patronId must be used by all subsequent service calls made on behalf of that patron. The full patron can be fetched with the operation corresponding to the patron type using the patron id and patron type.

		Swagger UI		
Parameters				
Parameter	Description	Parameter Type	Data Type	
agencyid	ISIL of the agency (e.g. DK-761500)	path	string	
patronId	patron id of the patron	body	integer	
Response Class				
Model Model Sc	hema			
<pre>authenticateStatus (string) = ['VALID' or 'INVALID' or 'LOANER_LOCKED_OUT']: AuthenticateStatus: - 'VALID': successfully authenticated - 'INVALID': either the user is not known in the system, or an invalid combination of authentication parameters has been used. - 'LOANER_LOCKED_OUT': user has been blocked temporary because of too many failed login attempts , patronId (integer), patronType (string) = ['PERSON' or 'COMPANY' or 'LIBRARY' or 'GROUP'] }</pre>				
 'INVALID': eithe 'LOANER_LOCKI , patronId (intege patronType (stri }	r the user is not known in the system, or an invalid ED_OUT': user has been blocked temporary becaus rr), ng) = ['PERSON' or 'COMPANY' or 'LIBRARY' or 'GR	se of too many failed login atter	•	
 'INVALID': eithe 'LOANER_LOCKI patronId (intege patronType (stri Response Conter	r the user is not known in the system, or an invalid ED_OUT': user has been blocked temporary becaus rr), ng) = ['PERSON' or 'COMPANY' or 'LIBRARY' or 'GR nt Type application/json v	se of too many failed login atter	•	
 'INVALID': eithe 'LOANER_LOCKI , patronId (intege patronType (stri }	r the user is not known in the system, or an invalid ED_OUT': user has been blocked temporary becaus rr), ng) = ['PERSON' or 'COMPANY' or 'LIBRARY' or 'GR nt Type application/json ❤ ages	se of too many failed login atter	•	
 'INVALID': eithe 'LOANER_LOCKI patronId (intege patronType (stri Response Conter Response Mess	r the user is not known in the system, or an invalid ED_OUT': user has been blocked temporary becaus rr), ng) = ['PERSON' or 'COMPANY' or 'LIBRARY' or 'GR nt Type application/json ❤ ages	se of too many failed login atter OUP']	•	

POST /external/{agencyid}/patrons/preauthenticated/v10

Returns a patron id, patron type and authentication status for a patron that the client has pre-authenticated using a third party.

Implementation Notes

Patrons can be looked up based on a patronIdentifier number (e.g. CPR, Library card, UNI-Login). The returned patronId must be used by all subsequent service calls made on behalf of that patron. The full patron can be fetched with the operation corresponding to the patron type using

5/5/25,

PM		Swagger UI	
the patron id and pa	atron type.		
Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronIdentifier	Given patron identifier	body	string
Response Class			
Model Model Schei	ma		
}	= ['PERSON' or 'COMPANY' or 'LIBRARY' or 'GROUP	']	
Response Message	es		
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
POST /external/{	[agencyid}/patrons/resetPincode/v1	Send a reset	pincode email to the patron with the provided emailaddres
	agencyian parions/resetrincode/vr	Senu a leset	pricede entan to the pation with the provided entalladures

Implementation Notes

The response is successful, if an email was successfully sent to the patron to reset their pincode.

If the supplied emailaddress has an invalid format, then response message 400 will be sent back.

If multiple patrons within the supplied agency have the same emailaddress, then response message 400 will be sent back.

If the emailadress does not correspond to any patron, then response message 404 will be sent back.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
emailAddress	emailAddress belonging to the patron	body	string
Response Message	25		
HTTP Status Code	Reason	Response Model	
400	bad request - invalid format or multiple patrons fo for emailaddress	und	
401	client unauthorized		
403	forbidden access to wrong agency		
404	no patron found for emailAddress		

POST /external/{agencyid}/patrons/updatePincode/v1 Using the UUID provided through a reset pincode email, update the pincode of the patron.

Implementation Notes

The response is successful, if the pincode was successfully updated.

If the supplied pincode is invalid, blank or too long, then response message 400 will be sent back.

If the supplied UUID is invalid, blank or not a valid format, then response message 400 will be sent back.

If the UUID has expired or cannot be found, then response message 404 will be sent back.

5, 1:11	PM		Swagger UI	
	Parameter	Description	Parameter Type	Data Type
	agencyid	ISIL of the agency (e.g. DK-761500)	path	string
	updatePincode	Request to update patron pincode	body	Model Model Schema UpdatePincodeRequestV1 { pincode (string): The new pincode for the patron, uuid (string): UUID provided to patron in email when resetting pincode }
	Response Messages			
	HTTP Status Code	Reason	Response Model	
	400	bad request - invalid format for pincode or UUID		
	401	client unauthorized		
	403	forbidden access to wrong agency		
	404	UUID not found		

POST /external/{agencyid}/patrons/v10

Create a new patron who is a person.

Implementation Notes

When a patron doesn't have a patron account in the library system, but logs in using a trusted authentication source (e.g NemId), the patron account can be created using this service. Name and address will be automatically fetched from configured person registry, and cannot be supplied by the client. If the configured person registry is not authorized to provide information about the patron, then response message 404 will be sent back

Multiple email addresses and phone numbers can be stored for a patron.

If multiple email addresses are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If multiple phone numbers are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

5/5/25.

5/5/25, 1:11 PM

Swagger UI

This version includes a possibility of including a BlockStatus in the request. The reason of the blockstatus must be one of the following one-letter choices

- 'O': library card stolen
- 'F': extended exclusion
- 'U': exclusion

- 'S': blocked by self service automaton
 'W': self created at website
 'E': maximum amount of allowed debt exceeded
- 'D': deceased

If the request includes a blockstatus for the patron, and the blockedSince date is not given or in the wrong format, or the blockedReason is not one of the one-letter choices shown above, then response message 400 will be sent back.

Setting nationalRegistryConsent to true will allow the data of the created patron to be synchronized with the Norwegian national registry.

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
createPatronRequest	the patron to be created	body	Model Model Schema
			CreatePatronRequestV8 { pincode (string, optional), patron (PatronSettingsV6), nationalRegistryConsent (boolean, optional), personIdentifier (string), blockStatusRequest (BlockStatusRequest, optional) } PatronSettingsV6 { preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place If left empty default library language will be used, notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV1], optional): Existing email addresses are overwritten with these values If left empty existing email addresses are deleted,

Parameter	Description	Parameter Type Data Type
		<pre>preferredPickupBranch (string): ISIL-number of preferred pickup branch, onHold (Period, optiona): If not set then the patron not on hold, guardianVisibility (boolean), receivePostalMail (boolean), interests (array[String], optiona): A list of interests the patron., phoneNumbers (array[PhoneNumberV1], optiona) Existing phonenumbers are overwritten with these values If left empty existing phonenumbers are deleted } EmailAddress (string), receiveNotification (boolean) } Period { from (string, optiona): Open-ended if not set, to (string, optiona): Open-ended if not set } PhoneNumberV1 { receiveNotification (boolean), phoneNumber (string) } BlockStatusRequest { blockedReason (string): Reason code for block, blockedSince (string): The date from which the patron should be blocked. The dateformat is YYYY- MM-DD, so 9th of March 2023 is written 2023-03-05 }</pre>
Response Class		
Model Model Sch	nema	
AuthenticatedPatr patron (PatronV9	onV10 { , optional): Only available if patron exists in	BS and was succesfully authenticated.,

authenticateStatus (string) = ['VALID' or 'INVALID' or 'LOANER_LOCKED_OUT']: AuthenticateStatus:

- 'VALID': successfully authenticated

- 'INVALID': either the user is not known in the system, or an invalid combination of authentication parameters has been used.

- 'LOANER_LOCKED_OUT': user has been blocked temporary because of too many failed login attempts

}

PatronV9 {

birthday (string, *optional*),

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

guardianVisibility (boolean),

receiveEmail (boolean),

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

keepLoanHistoricalData (boolean, optional): Patron consent to keep historical loans,

receiveSms (boolean),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

hasNationalRegistrySynchronizationConsent (boolean),

emailAddress (string, optional),

notificationProtocols (array[string], *optional*): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., **phoneNumber** (string, *optional*),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, optional): True if the user is allowed to create bookings.,

defaultInterestPeriod (integer): Length of default interest period in days,

interests (array[string], optional),

resident (boolean): True if the user is resident in the same municipality as the library

}

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code,

district (string): Dsi		
subDistrict (string)	Subdistrict	
}		
Period {		
	nal): Open-ended if not set,	
to (string, optional)	: Open-ended if not set	
}		
BlockStatus {		
blockedReason (st	ring): Reason code for block,	
blockedSince (strin	lg),	
message (string): N	lessage about block	
}		
Response Content 7	ype application/json 🗸	
Response content i		
Response Messag	es	
		Response Model
Response Messag	es	Response Model
Response Messag HTTP Status Code	es Reason	Response Model

POST /external/{agencyid}/patrons/v9

Create a new patron who is a person.

Implementation Notes

When a patron doesn't have a patron account in the library system, but logs in using a trusted authentication source (e.g NemId), the patron account can be created using this service. Name and address will be automatically fetched from configured person registry, and cannot be supplied by the client. If the configured person registry is not authorized to provide information about the patron, then response message 404 will be sent back

Multiple email addresses and phone numbers can be stored for a patron.

If multiple email addresses are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If multiple phone numbers are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

This version includes a possibility of including a BlockStatus in the request. The reason of the blockstatus must be one of the following one-letter choices

- 'O': library card stolen 'F': extended exclusion
- 'U': exclusion
- 'S': blocked by self service automaton 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

If the request includes a blockstatus for the patron, and the blockedSince date is not given or in the wrong format, or the blockedReason is not one of the one-letter choices shown above, then response message 400 will be sent back.

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
createPatronRequest	the patron to be created	body	Model Model Schema
			CreatePatronRequestV7 { pincode (string), patron (PatronSettingsV6), personIdentifier (string), blockStatusRequest (BlockStatusRequest, optional) } PatronSettingsV6 { preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place If left empty default library language will be used, notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., emailAddresses (array[EmailAddressV1], optional): Existing email addresses are overwritten with these values If left empty existing email addresses are deleted,

Swagger UI

Parameter	Description	Parameter Type Data Type
		<pre>preferredPickupBranch (string): ISIL-number of preferred pickup branch, onHold (Period, optionah): If not set then the patron not on hold, guardianVisibility (boolean), receivePostalMail (boolean), interests (array[string], optionah): A list of interests of the patron., phoneNumbers (array[PhoneNumberV1], optionah): Existing phonenumbers are overwritten with these values If left empty existing phonenumbers are deleted } EmailAddressV1 { emailAddress(string), receiveNotification (boolean) } Period { from (string, optionah): Open-ended if not set, to (string, optionah): Open-ended if not set } PhoneNumberV1 { receiveNotification (boolean), phoneNumberV1 { receiveNotification (boolean), phoneNumberV1 { receiveNotification (boolean), phoneNumber (string) } BlockStatusRequest { blockedReason (string): Reason code for block, blockedSince (string): The date from which the patron should be blocked. The dateformat is YYYY- MM-DD, so 9th of March 2023 is written 2023-03-09. }</pre>
Response Class		
Model Model Sch	ema	
AuthenticatedPatro patron (PatronV9,	onV10 { , <i>optional</i>): Only available if patron exists in	BS and was succesfully authenticated.,

authenticateStatus (string) = ['VALID' or 'INVALID' or 'LOANER_LOCKED_OUT']: AuthenticateStatus:

- 'VALID': successfully authenticated

- 'INVALID': either the user is not known in the system, or an invalid combination of authentication parameters has been used.

- 'LOANER_LOCKED_OUT': user has been blocked temporary because of too many failed login attempts

}

PatronV9 {

birthday (string, *optional*),

secondaryAddress (AddressV3, optional),

preferredLanguage (string, optional): Language in which the patron prefers the communication with the library to take place,

preferredPickupBranch (string): ISIL of preferred pickup branch,

address (AddressV3, optional),

onHold (Period, optional): If not set then the patron is not on hold,

patronId (integer): Patron identifier to be used in subsequent service calls involving the patron,

guardianVisibility (boolean),

receiveEmail (boolean),

blockStatus (array[BlockStatus], optional): A list of block statuses - if the patron is not blocked then this value is empty or null,

keepLoanHistoricalData (boolean, optional): Patron consent to keep historical loans,

receiveSms (boolean),

tags (array[string], optional): Tags associated with the patron. Can be used to e.g. give permissions to patron in external ressources.,

hasNationalRegistrySynchronizationConsent (boolean),

emailAddress (string, optional),

notificationProtocols (array[string], *optional*): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included., **phoneNumber** (string, *optional*),

name (string, optional),

receivePostalMail (boolean),

allowBookings (boolean, optional): True if the user is allowed to create bookings.,

defaultInterestPeriod (integer): Length of default interest period in days,

interests (array[string], optional),

resident (boolean): True if the user is resident in the same municipality as the library

AddressV3 {

country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code,

[}]

	trict,	
subDistrict (string)	Subdistrict	
}		
Period {		
from (string, optior	al): Open-ended if not set,	
to (string, optional)	Open-ended if not set	
}		
BlockStatus {		
blockedReason (st	ring): Reason code for block,	
blockedSince (strin	g),	
message (string): N	lessage about block	
}		
Response Content 1	ype application/json 🗸	
Response Messag	es	Demense Medel
		Response Model
Response Messag	es	Response Model
Response Messag HTTP Status Code	es Reason	Response Model

PUT /external/{agencyid}/patrons/withGuardian/v2

Updates a person patron's guardian (eg A financial responsible).

Implementation Notes

If the person doesn't have a guardian, a new one is created with the information provided. Returns the id of the patron if the request succeeds. Name and address will be automatically fetched from the configured person registry.

If the configured person registry is not authorized to provide information about the patron and guardian, then response message 404 will be sent back.

If the supplied person identifier (e.g. CPR-number) of the patron equals that of the guardian, then response message 400 will be sent back.

If the email of the guardian is invalid, then response message 400 will be sent back.

In case of a successful update of the guardian, a confirmation email is sent to the guardian. In case of failure an email is sent to guardian stating the update failed.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
updateGuardianRequest	The payload with information for the	body	Model Model Schema
	guardian update		UpdateGuardianRequestV2 {
			patronId (integer, optional),
			personId (string, <i>optional</i>): If patronId is provided,
			this field will be ignored,
			guardian (GuardianV2)
			}
			GuardianV2 {
			mobilePhoneNumber (string, optional),
			address (AddressV3, optional),
			name (string): The full name of the guardian,
			personIdentifier (string),
			email (string): Must be valid
			}
			AddressV3 {
			country (string): Country,
			city (string): City,
			street (string): Street and number,
			coName (string): c/o name,
			postalCode (string): Postal code,
			district (string): Dsitrict,
			subDistrict (string): Subdistrict
			}
Response Class			

Response Class

Model Model Schema

integer

		Swayyer Or			
Response Content T	Response Content Type application/json 🗸				
Response Message	Response Messages				
HTTP Status Code	Reason	Response Model			
200	Ok				
400	bad request				
401	client unauthorized				
404	Data not found				

POST /external/{agencyid}/patrons/withGuardian/v3	
---	--

Creates a person patron with a guardian (eg A financial responsible).

Implementation Notes

Returns the id of the patron if the request succeeds. Name and address will be automatically fetched from the configured person registry. This version includes a possibility of including a BlockStatusRequest in the PatronWithGuardianRequest. The reason of the blockstatusRequest must be one of the following one-letter choices

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton
- 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

If the configured person registry is not authorized to provide information about the patron, then response message 404 will be sent back.

If the supplied person identifier (e.g. CPR-number) of the patron equals that of the guardian, then response message 400 will be sent back.

If the email of the guardian is invalid, then response message 400 will be sent back.

If an email or phone number for the patron is supplied and it is invalid, then response message 400 will be sent back.

Multiple email addresses and phone numbers can be stored for a patron.

If multiple email addresses are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If multiple phone numbers are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the

rest will be stored as not preferred.

If the request includes a blockstatus for the patron, and the blockedSince date is not given or in the wrong format, or the blockedReason is not one of the one-letter choices shown above, then response message 400 will be sent back

In case of a successful creation of the patron, a confirmation email is sent to the guardian. In case of failure an email is sent to guardian stating the creation failed.

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronWithGuardianRequest	The payload with information for the	body	Model Schema
	patron to create		PatronWithGuardianRequestV3 {
			pincode (string),
			emailAddresses (array[EmailAddressV1],
			optional): Must be valid if supplied,
			address (AddressV3, optional),
			<pre>preferredPickupBranch (string): Must be a valid</pre>
			branch id (eg. DK-761501),
			name (string): The full name of the patron,
			personId (string),
			guardian (GuardianV2),
			<pre>blockStatusRequest (BlockStatusRequest,</pre>
			optional),
			<pre>phoneNumbers (array[PhoneNumberV1],</pre>
			<i>optional</i>): Must be valid if supplied
			}
			EmailAddressV1 {
			emailAddress (string),
			receiveNotification (boolean)
			}
			AddressV3 {
			country (string): Country,
			city (string): City,
			street (string): Street and number,
			coName (string): c/o name,

5/5/25, 1:11 PM

Swagger UI

M		Swagger UI	
Parameter	Description	Parameter Type	Data Type
			postalCode (string): Postal code,
			district (string): Dsitrict, subDistrict (string): Subdistrict
			}
			, GuardianV2 {
			mobilePhoneNumber (string, optional),
			address (AddressV3, optional),
			name (string): The full name of the guardian,
			personIdentifier (string), email (string): Must be valid
			}
			, BlockStatusRequest {
			blockedReason (string): Reason code for block,
			blockedSince (string): The date from which the
			patron should be blocked. The dateformat is YYYY
			MM-DD, so 9th of March 2023 is written 2023-03- 09.
			1
			, PhoneNumberV1 {
			receiveNotification (boolean),
			phoneNumber (string)
			}
Response Class			
Model Model Sche	ma		
integer			
integer			
Response Content T	ype application/json V		
Response Messag	es		
HTTP Status Code	Reason	Response Model	
200	Ok		
400	bad request		
401	client unauthorized		

PM		Swagger UI
HTTP Status Code	Reason	Response Model
404	Data not found	

/external/{agencyid}/patrons/withGuardian/v4 POST

Creates a person patron with a guardian (eg A financial responsible).

Implementation Notes

Returns the id of the patron if the request succeeds. Name and address will be automatically fetched from the configured person registry. This version includes a possibility of including a BlockStatusReguest in the PatronWithGuardianReguest. The reason of the blockstatusReguest must be one of the following one-letter choices

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

If the configured person registry is not authorized to provide information about the patron, then response message 404 will be sent back.

If the supplied person identifier (e.g. CPR-number) of the patron equals that of the guardian, then response message 400 will be sent back.

If the email of the guardian is invalid, then response message 400 will be sent back.

If an email or phone number for the patron is supplied and it is invalid, then response message 400 will be sent back.

Multiple email addresses and phone numbers can be stored for a patron.

If multiple email addresses are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If multiple phone numbers are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If the request includes a blockstatus for the patron, and the blockedSince date is not given or in the wrong format, or the blockedReason is not one of the one-letter choices shown above, then response message 400 will be sent back

In case of a successful creation of the patron, a confirmation email is sent to the guardian. In case of failure an email is sent to guardian stating the creation failed.

Setting nationalRegistryConsent to true will allow the data of the created patron to be synchronized with the Norwegian national registry.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronWithGuardianRequest	The payload with information for the	body	Model Model Schema
patronwithGuardiankequest	patron to create	body	<pre>PatronWithGuardianRequestV4 { pincode (string, optional), nationalRegistryConsent (boolean, optional), emailAddresses (array[EmailAddressV1], optional): Must be valid if supplied, address (AddressV3, optional), preferredPickupBranch (string): Must be a valid branch id (eg. DK-761501), name (string): The full name of the patron, personId (string), guardian (GuardianV2), blockStatusRequest (BlockStatusRequest, optional), phoneNumbers (array[PhoneNumberV1], optional): Must be valid if supplied } EmailAddressV1 { emailAddress(string), receiveNotification (boolean)</pre>
			} AddressV3 {
			<pre>country (string): Country, city (string): City, street (string): Street and number, coName (string): c/o name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Subdistrict }</pre>
			GuardianV2 { mobilePhoneNumber (string, optional), address (AddressV3, optional),

Parameter	Description	Parameter Type	Data Type
	Description	r di difficieri Type	name (string): The full name of the guardian,
			personIdentifier (string),
			email (string): Must be valid
			}
			BlockStatusRequest {
			blockedReason (string): Reason code for block, blockedSince (string): The date from which the
			patron should be blocked. The dateformat is YYY MM-DD, so 9th of March 2023 is written 2023-03
			09. }
			ہ PhoneNumberV1 {
			receiveNotification (boolean),
			phoneNumber (string)
			}
Response Class			
Model Model Sche	ma		
integer			
Response Content T	ype application/json 🗸		
Response Messag	es		
HTTP Status Code	Reason	Response Model	
200	Ok		
400	bad request		
401	client unauthorized		
404	Data not found		

OST	/externa	l/{agencyid	}/patrons	/withGuarc	lian/v5
-----	----------	-------------	-----------	------------	---------

Creates a person patron with a guardian (eg A financial responsible).

Implementation Notes

Returns the id of the patron if the request succeeds. Name and address will be automatically fetched from the configured person registry. This version includes a possibility of including a BlockStatusRequest in the PatronWithGuardianRequest. The reason of the blockstatusRequest must be one of the following one-letter choices

- 'O': library card stolen
- 'U': exclusion
- 'F': extended exclusion
- 'S': blocked by self service automaton
- 'W': self created at website
- 'E': maximum amount of allowed debt exceeded
- 'D': deceased

If the configured person registry is not authorized to provide information about the patron, then response message 404 will be sent back.

If the supplied person identifier (e.g. CPR-number) of the patron equals that of the guardian, then response message 400 will be sent back.

If the email of the guardian is invalid, then response message 400 will be sent back.

If an email or phone number for the patron is supplied and it is invalid, then response message 400 will be sent back.

Multiple email addresses and phone numbers can be stored for a patron.

If multiple email addresses are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If multiple phone numbers are supplied having receiveNotification as true, then only one of them will be randomly stored as preferred and the rest will be stored as not preferred.

If the request includes a blockstatus for the patron, and the blockedSince date is not given or in the wrong format, or the blockedReason is not one of the one-letter choices shown above, then response message 400 will be sent back

In case of a successful creation of the patron, a confirmation email is sent to the guardian. In case of failure an email is sent to guardian stating the creation failed.

Setting nationalRegistryConsent to true will allow the data of the created patron to be synchronized with the Norwegian national registry.

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string

Parameter Description Parameter Type Data Type patronWithGuardianRequest The payload with information for the patron to create body Model Model Schema PatronWithGuardianRequestV5 { pincode (string, optional), nationalRegistryConsent (boolean, optional), preferredLanguage (string, optional): Language in	patronWithGuardianRequest The payload with information for the patron to create body Model Model Model Schema PatronWithGuardianRequestV5 { pincode (string, optional), nationalRegistryConsent (boolean, optional), nerfersetEnguage (tring, optional), communication with the library to take place, emailAddresses (array[EmailAddressV1], optional), Nust be availd branch (string): Must be availd branch (string): The full name of the patron, performed(string), receivent(GaurdianV2), blockStatusRequest, optional), phoneNumbers (array[PhoneNumberV1], optional), array[PhoneNumberV1], o
patron to create PatronWithGuardianRequestV5 { pincode (string, optional), nationalRegistryConsent (boolean, optional),	patron to create Patron/WithGuardianRequestV5 { pincode (string, optional), nationalRegistryConsent (boolean, optional), preferredLanguage (string, optional), Language in which the patron prefers the communication with the library to take place, emailAddresses (array[EmailAddressV1], optional; Must be valid if Supplied, address (AddressV3, optional), preferredPickupBranch (string); Must be a valid branch id (eg. DK-761501), name (string); The full name of the patron, personid (string), guardian (GuardianV2), blockStatusRequest (BlockStatusRequest, optional), mphoneNumbers (array[PhoneNumberV1], optional; Must be valid if supplied } blockStatusRequest (BlockStatusRequest, optional), mane (string); Must be valid if supplied } country (string): Counts), city (string); Cliv, street (string); Street and number, coloame (string); Street and number,
<pre>the library to take place, emailAddress(array[EmailAddressV1, optiona): Must be valid if supplied, address (AddressV3, optiona), preferredPickupBranch (string): Must be a valid branch id (eg. DK-761501), name (string): The full name of the patron, personId (string), guardian (GuardianV2), blockStatusRequest (BlockStatusRequest, optiona); Must be valid if supplied } blockStatusRequest (BlockStatusRequest, optiona); Must be valid if supplied } EmailAddress (string), receiveNotification (boolean) } AddressV3 { country (string): Country, city (string): Country, city (string): Street and number,</pre>	district (string): Dsitrict,

1 PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
Response Class		Tarameter type	<pre>personIdentifier (string), email (string): Must be valid } BlockStatusRequest { blockedReason (string): Reason code for block, blockedSince (string): The date from which the patron should be blocked. The dateformat is YYYY- MM-DD, so 9th of March 2023 is written 2023-03- 09. } PhoneNumberV1 { receiveNotification (boolean), phoneNumber (string) }</pre>
Model Model Sche	ma		
integer			
Response Content T	ype application/json 🗸		
Response Message	es		
HTTP Status Code	Reason	Response Model	
200	Ok		
400	bad request		
401	client unauthorized		

POST /external/{agencyid}/patrons/withoutIntegration/v1

Create a person patron with a guardian without using an integration (such as a person registry) to provide patron information.

Implementation Notes

When a patron is not registered in the library system, the patron can be created using this service.

Only the details provided in the payload are used for registration; no integration is used to retrieve patron data (e.g., from a person registry).

Multiple email addresses and phone numbers can be stored for a patron.

If multiple email addresses are provided with "receiveNotification" set to true, only the first one in the list with this setting ("receiveNotification" = true) will be stored as the preferred email. The rest will be stored as not preferred.

If multiple phone numbers are provided with "receiveNotification" set to true, only the first one in the list with this setting ("receiveNotification" = true) will be stored as the preferred phone number. The rest will be stored as not preferred.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronWithoutIntegrationRequest	The patron to be created	body	Model Schema
			CreatePatronWithoutIntegrationRequestV1 { addressInfo (PatronAddressInfoV1, optional): Patron address information, bankInfo (PatronBankInfoV1, optional): Patron bank information, notificationProtocols (array[string], optional): Notification protocols that the patron want to receive notification on. SMS and EMAIL are not included, contactInfo (PatronContactInfoV1, optional): Patron contact information, pinCode (string, optional): The pin code of the patron, guardianInfo (PatronGuardianInfoV1, optional): Patron guardian information, guardianVisibility (boolean, optional): Patron guardianVisibility, generalInfo (PatronGeneralInfoV1): Patron general information,

Parameter	Description	Parameter Type	Data Type
Parameter	Description	Parameter Type	<pre>Data Type libraryInfo (PatronLibraryInfoV1): Patron library information } PatronAddressInfoV1 { secondaryAddress (AddressV3, optional): Secondary address of the person. The country code must be specified in a two-letter format (e.g. DK, SE, NO, DE), primaryAddress (AddressV3, optional): Primary address of the person. The country code must be specified in a two-letter format (e.g. DK, SE, NO, DE) } AddressV3 { country (string): Country, city (string): City, street (string): Street and number, coName (string): Cro name, postalCode (string): Postal code, district (string): Dsitrict, subDistrict (string): Dsitrict, subDistrict (string): Subdistrict } PatronBankInfoV1 { accountHolder (string, optional): The name of the account holder, iban (string, optional): The International Bank Account Number (IBAN) of the account, bic (string, optional): The Bank Identifier Code (BIC) of the account } PatronContactInfoV1 { emailAddresses (array[EmailAddressV1], optional): Patron's email addresse. If multiple email addresses are provided and have 'receiveNotification': true, the first one is considered the primary email address, phoneNumbers (array[PhoneNumberV1], optional): Patron's phone numbers. If multiple phone numbers are provided and have</pre>
8080/externalapidocs/			

Parameter	Description	Parameter Type	Data Type
			'receiveNotification': true, the first one is considered
			the primary phone number
			}
			EmailAddressV1 {
			emailAddress (string),
			receiveNotification (boolean)
			}
			PhoneNumberV1 {
			receiveNotification (boolean),
			phoneNumber (string)
			}
			PatronGuardianInfoV1 {
			country (string, <i>optional</i>): Country of the guardian.
			The country code must be specified in a two-letter
			format (e.g. DK, SE, NO, DE),
			address (string, optional): Address of the guardian,
			gender (string, <i>optional</i>): Gender of the guardian. Allowed values are 'MALE', 'FEMALE', 'OTHER',
			'UNKNOWN',
			mobilePhone (string, <i>optional</i>): Mobile phone
			number of the guardian,
			city (string, <i>optional</i>): City of the guardian,
			homePhone (string, optional): Home phone
			number of the guardian,
			postalCode (string, <i>optional</i>): Postal code of the
			guardian,
			name (string): Name of the guardian,
			personIdentifier (string, <i>optional</i>): Person
			identifier of the guardian,
			workPhone (string, <i>optional</i>): Work phone number of the guardian,
			birthDate (string, <i>optional</i>): Birthdate of the
			guardian,
			email (string): Email of the guardian. Must be valid
			}
			PatronGeneralInfoV1 {
			gender (string, optional): Gender of the person.
			Allowed values are 'MALE', 'FEMALE', 'OTHER',

I PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
Response Class Model Model Sche integer			<pre>'UNKNOWN', name (string): Name of the person, personIdentifier (string, optional): Person identifier, birthDate (string, optional): Birth date of th person } PatronLibraryInfoV1 { preferredLanguage (string, optional): Lang which the patron prefers the communicatio the library to take place. If left empty defau language will be used, preferredPickupBranch (string): Preferred branch of the patron }</pre>
Response Content T	ype application/json 🗸		
Response Messag	es		
HTTP Status Code	Reason	Response Model	
200	Ok		
400	Bad Request - The request is malformed or contains invalid data.		
401	Client Unauthorized		
	Client Forbidden - The client does not have access to		

Authentication

Show/Hide List Operations Expand Operations Raw

POST /external/v1/{agencyid}/authentication/login

Swagger UI

Implementation Notes

The other services can only be used by an authenticated client system. This service authenticates the client system to be able to use the other services.

The response contains a session key which must be supplied in the HTTP header 'X-Session' of all subsequent service calls. The session key can timeout (yielding HTTP Status code 401 on a service call), and if this happens the client system must login again.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency to log into (e.g. DK-761500)	path	string
login	credentials for the client system	body	Model Model Schema
			Login { password (string): Clear text password, username (string): Username for the client system }
Response Class			
Model Model Schem	la		
ExternalAPIUserInfo { sessionKey (string): 5 }	Session key for subsequent API calls		
Response Content Ty	pe application/json 🗸		
Response Messages	5		
HTTP Status Code	Reason	Response Model	
400	bad request		
403	invalid client credentials		

Configuration

GET /external/v1/{agencyid}/configuration/{key}

Get a configuration setting based on a configuration key.

Implementation Notes

Returns a string representation of the setting.

Note: If the settings for the key is a list of values, this method is not suitable for getting the settings.

The list of available configuration settings will be distributed seperately to the clients that needs them, along with a description on how the settings is encoded in the string representation.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
key	the configuration key to look up	path	string
Response Class string			
Response Content Ty	0e application/json ➤		
Response Messages			
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
404	key not found		

GET	/external/v1/{agencyid}/configuration/{key}/list
-----	--

Get a configuration setting based on a configuration key.

Implementation Notes

Returns an array of strings representation of the setting.

This method is suitable for keys that has a list of values.

The list of available configuration settings will be distributed seperately to the clients that needs them, along with a description on how the settings is encoded in the string representation.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
key	the configuration key to look up	path	string
Response Class			
Model Model Scher	na		
array[string]			
Response Content Type application/json V			
Response Messages			
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
404	key not found		

Patron Group

Show/Hide List Operations Expand Operations Raw

/external/v1/{agencyid}/patrongroups

Returns the root group of a specific agency.

Implementation Notes

GET

Returns the root of the PatronGroup and the whole groups hierarchy in the specified agency.

Parameters

Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
Response Class			
Model Model Sche	ma		
<pre>patronGroupId (in name (string): The description (string) childGroups (array }</pre>	ateger): The number of all members in this group, teger): The patron group identifier, name of the group,): The description of the group, [PatronGroup], <i>optional</i>): The array of child group:		
•			
Response Messag HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
Booking			Show/Hide List Operations Expand Operations Raw
GET /external/	v1/{agencyid}/patrons/{patronid}/bookin	gs	Retrieve all bookings made by the patron.
Implementation N	lotes		
Returns an array of	booking details in one of the following states		
- active			

- fulfilled - beingFulfilled

Bookings having any other state will not be received.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the ID of the patron that owns the bookings	path	integer

Response Class

Model Model Schema

Booking {

recordId (string): The record ID,

preferredMaterials (integer): The preferred number of materials,

note (string, optional): Additional information about this booking,

period (Period): The booking period information containing the start and the end date,

minimumMaterials (integer): The minimum number of materials,

patronGroupId (integer): The patron group ID for this booking,

automaticForwardLoan (boolean): True if automatic forward is active for this booking,

state (string): The booking state,

requestedFromBranchId (string): The branch that provides the material for booking,

bookingId (string): The booking identifier,

deliverBranchId (string): The delivery branch identifier

}

Period {

from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set

}

Response Content Type application/json 🗸

Response Messages

5/5/25, 1:11	I PM		Swagger UI
	HTTP Status Code	Reason	Response Model
	400	bad request com.dantek.dl.rest.RestException	
	401	client unauthorized	
	404	patron not found	

POST /external/v1/{agencyid}/patrons/{patronid}/bookings

Create a new bookings for the patron.

Implementation Notes

Given a CreateBookingBatch, it creates a list of bookings and returns an array of BookingResponse.

Each response element contains a result of the creation and if ALL results have the value success the created Booking is returned. Otherwise the field is null.

No booking is created if ANY element in the CreateBookingBatch fails to be created

The result of the creation can have the following values:

- success
- notEnoughMaterials
- notUpdatable
- patronDoesNotHavePermission
- other

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the ID of the patron that owns the bookings	path	integer
batch	information about bookings that are going to be created	body	Model Model Schema CreateBookingBatch { bookings (array[CreateBooking])
			<pre>} CreateBooking { recordId (string): The record ID,</pre>

11 PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
			<pre>preferredMaterials (integer): The preferred number of materials, note (string, optional): Additional information about this booking, period (Period): The booking period information containing the start and the end date, minimumMaterials (integer): The minimum number of materials, patronGroupId (integer): The patron group ID for this booking, automaticForwardLoan (boolean): True if automatic forward is active for this booking, requestedFromBranchId (string): The branch that provides the material for booking, deliverBranchId (string): The delivery branch identifier } Period { from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set } </pre>
Response Class			
Model Model Sch	ema		
BookingResponse { result (string): The booking (Booking }		d by the create/update operation	
note (string, optic period (Period): T minimumMateria	The record ID, als (integer): The preferred number of n <i>anal</i>): Additional information about this he booking period information containi als (integer): The minimum number of r nteger): The patron group ID for this bo	booking, ng the start and the end date, naterials,	
automaticForwa	rdLoan (boolean): True if automatic for	ward is active for this booking,	

state (string): The booking state,

bookingId (string):	The booking identifier,	provides the material for booking,	
deliverBranchId (s	tring): The delivery branch identif	ier	
}			
Period {			
from (string, optior	<i>nal</i>): Open-ended if not set,		
to (string, optional)	: Open-ended if not set		
}			
Response Content T	ype application/json 🗸		
Response Messag	es		
HTTP Status Code	Reason	Response Model	
400	bad request com.dantek.dl.re	st.RestException	
401	client unauthorized		
404	patron not found		

PUT /external/v1/{agencyid}/patrons/{patronid}/bookings

Update existing bookings

Implementation Notes

Given an UpdateBookingBatch, it updates the list of existing bookings and returns an array of BookingResponse.

Each response element contains a result of the update operation and the updated Booking if the operation succeeds. On failure, the result field is updated accordingly and the booking is set to null.

The result of the creation can have the following values:

- success

- notEnoughMaterials

- notUpdatable

- patronDoesNotHavePermission

- other

Parameters

/25, 1:11 PM		Swagger UI	
Parameter	Description	Parameter Type	Data Type
agencyid		path	string
patronid	the ID of the patron that owns the bookings	path	integer
batch	information about bookings that are going to be updated	body	Model Schema
			UpdateBookingBatch {
			bookings (array[Booking])
			}
			Booking {
			recordId (string): The record ID,
			preferredMaterials (integer): The preferred number of materials,
			note (string, <i>optional</i>): Additional information about this booking,
			period (Period): The booking period information containing the start and the end date,
			minimumMaterials (integer): The minimum number of materials,
			patronGroupId (integer): The patron group ID for this booking,
			automaticForwardLoan (boolean): True if automatic
			forward is active for this booking,
			state (string): The booking state,
			requestedFromBranchId (string): The branch that
			provides the material for booking,
			bookingId (string): The booking identifier,
			deliverBranchId (string): The delivery branch identifier
			}
			Period {
			<pre>from (string, optional): Open-ended if not set, to (string, optional): Open-ended if not set</pre>
			}
Response Class			
Model Model Sch	nema		
	i ci i u		

BookingResponse {

result (string): The	operation result,				
booking (Booking, <i>optional</i>): The booking data as returned by the create/update operation					
}					
Booking {					
recordId (string): T	he record ID,				
preferredMaterial	s (integer): The preferred nu	mber of materials,			
note (string, option	<i>al</i>): Additional information a	pout this booking,			
period (Period): The	e booking period information	n containing the start and the end date,			
	ls (integer): The minimum nu				
patronGroupId (in	teger): The patron group ID f	or this booking,			
		matic forward is active for this booking,			
state (string): The b	ooking state,				
•		hat provides the material for booking,			
	The booking identifier,				
deliverBranchId (s	string): The delivery branch io	entifier			
}					
Period {					
from (string, option	<i>nal</i>): Open-ended if not set,				
to (string, optional)	: Open-ended if not set				
}					
Response Content T	Гуре application/json ❤				
Response Messag	es				
HTTP Status Code	Reason	Response Model			
400	bad request com.dantek	dl.rest.RestException			
401	client unauthorized				
404	patron not found				

DELETE /external/v1/{agencyid}/patrons/{patronid}/bookings

Deletes bookings with the specified IDs.

Implementation Notes

Deletes the bookings corresponding to the given booking IDs

Parameters

Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the ID of the patron that owns the bookings	path	integer
Response Message	S		
HTTP Status Code	Reason	Response Model	
400	bad request com.dantek.dl.rest.RestException		
401	client unauthorized		
404	patron not found		

Reservations	Show/Hide List Operations	Expand Operations Raw
PUT /external/v1/{agencyid}/patrons/{patronid}/reservations		Update existing reservations.
Implementation Notes Returns an array of the updated reservation details. The response contains reservation state, which can be any of these values: - reserved - readyForPickup - inTransit - other The values are subject to change. If an unrecognized value is encountered, it should be tr The response contains loanType, which can be any of these values: - loan - interLibraryLoan The values are subject to change. If an unrecognized value is encountered, it should be tr		

11 PM		Swagger UI	
Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the reservations	path	integer
reservations	the reservations to be updated	body	Model Model Schema
			<pre>UpdateReservationBatch { reservations (array[UpdateReservation]) } UpdateReservation { expiryDate (string, optional): The date where the patron is no longer interested in the reserved material. If not set, the reservation will keep the original date., pickupBranch (string, optional): ISIL-number of pickup branch. If not set, the reservation will keep the original pickup branch., reservationId (integer): Identifies the reservation }</pre>
Response Class			
Model Model Schem	a		
expiryDate (string): T reservationId (intege	FAUST number,): ISIL-number of pickup branch, 'he date when the patron is no longer interested in er): Identifies the reservation for use when updatin	g or deleting the reservatio	ın,

pickupDeadline (string, optional): Set if reserved material is available for loan,

loanType (string),

dateOfReservation (string),

periodical (Periodical, *optional*): Present if material is a periodical,

ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans,

state (string),

numberInQueue (integer, optional): The number in the reservation queue.,

pickupNumber (string, optional): The reservation number. Will be present if the reservation is ready for pickup (the state is 'readyForPickup')

Periodical {

volume (string, optional),

volumeYear (string, optional),

displayText (string): A representation of the periodica volume information that is suitable for display,

volumeNumber (string, optional)

}

}

ILLBibliographicRecord {

author (string, optional): The author of the material, isbn (string, optional): ISBN-information from the bibliographic record, periodicalNumber (string, optional): Issue number of a periodical, edition (string, optional): Edition-information from the bibliographic record, language (string, optional): Language of the requested material., bibliographicCategory (string, optional): Bibliographic category from danMARC2 008 *t, title (string, optional): The title of the material, publicationDateOfComponent (string, optional): Publication date of an item component, or article., recordId (string): The FAUST number, issn (string, optional): ISSN-information from the bibliographic record, placeOfPublication (string, optional), mediumType (string): Type of the requested material - from danMARC2 009 *a+*g (general and specific), periodicalVolume (string, optional): Volume name of a periodical, publisher (string, optional): Publisher of the requested material., publicationDate (string, optional): Publication date of the requested material. }

Response Content Type application/json 🗸

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

DELETE /external/v1/{agencyid}/patrons/{patronid}/reservations

Delete existing reservations.

Parameters				
Parameter	Description	Parameter Type	Data Type	
agencyid	ISIL of the agency (e.g. DK-761500)	path	string	
patronid	the patron that owns the reservations	path	integer	
reservationid	a list of reservation ids for reservations that are to be deleted	query	array[integer]	
Response Messages				
HTTP Status Code	Reason	Response Model		
400	bad request			
401	client unauthorized			

GET /external/v1/{agencyid}/patrons/{patronid}/reservations/v2

Get all unfulfilled reservations made by the patron.

Implementation Notes

Returns an array of reservation details.

When the patron picks up the reserved materials, the reservation will no longer be returned. Expired or deleted reservations will not be returned.

The response contains reservation state, which can be any of these values:

- reserved
- readyForPickup
- inTransit
- other

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'other'.

The response contains reservationType, which can be any of these values:

- NORMAL
- PARALLEL
- SERIAL
- INTER_LIBRARY

The values are subject to change. If an unrecognized value is encountered, iit should be treated as 'normal'

The response contains a transactionId, which links together parallel reservations.

Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that owns the reservations	path	integer

Response Class

Model Model Schema

ReservationDetailsV2 {

pickupBranch (string): ISIL-number of pickup branch,

pickupDeadline (string, optional): Set if reserved material is available for loan,

dateOfReservation (string),

ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans,

numberInQueue (integer, optional): The number in the reservation queue.,

pickupNumber (string, optional): The reservation number. Will be present if the reservation is ready for pickup (the state is 'readyForPickup'),

transactionId (string): Identifies the transaction of reservations,

recordId (string): The FAUST number,

expiryDate (string): The date when the patron is no longer interested in the reserved material,

reservationId (integer): Identifies the reservation for use when updating or deleting the reservation,

periodical (Periodical, optional): Present if material is a periodical,

reservationType (string),

state (string)

}

ILLBibliographicRecord {

author (string, optional): The author of the material,

isbn (string, optional): ISBN-information from the bibliographic record,

periodicalNumber (string, optional): Issue number of a periodical,

edition (string, optional): Edition-information from the bibliographic record,

language (string, optional): Language of the requested material.,

bibliographicCategory (string, optional): Bibliographic category from danMARC2 008 *t,

title (string, optional): The title of the material,

publicationDateOfComponent (string, optional): Publication date of an item component, or article.,

recordId (string): The FAUST number,

issn (string, optional): ISSN-information from the bibliographic record,

5/5/25,	1:11	ΡM
---------	------	----

periodicalVolume (strin publisher (string, option publicationDate (string) } Periodical { volume (string, optional volumeYear (string, optional displayText (string): A r	ype of the requested material - from danMARC2 009 *a+*g (general and specific), ng, <i>optional</i>): Volume name of a periodical, <i>nal</i>): Publisher of the requested material., g, <i>optional</i>): Publication date of the requested material. <i>h</i> , <i>tional</i>), epresentation of the periodica volume information that is suitable for display,
volumeNumber (string	, optional)
}	
Response Content Type	application/json ✓
Response Messages	
HTTP Status Code R	Reason Response Model
400 b	pad request
401 c	lient unauthorized

POST /external/v1/{agencyid}/patrons/{patronid}/reservations/v2

Create new reservations for the patron.

Implementation Notes

Given a CreateReservationBatch, it creates a list of reservations and returns a ReservationResponse.

The CreateReservationBatch.type indicates the reservation type of the request. If left out the request will be considered of type normal. The type can be any of the following values:

- normal
- parallel

The values are subject to change.

ReservationResponse.success indicates if the reservations were created sucessfully. If any of the reservations have failed then all reservations will be failed and ReservationResponse.success will be false. If all reservations are successfully created ReservationResponse.success will be true.

ReservationResponse.reservationResults contains details about each reservation. A ReservationResult.result has the status of a reservation and can be any of the following values:

- success
- patron_is_blocked
- patron_not_found
- already reserved
- already_loaned
- material not loanable
- material not reservable
- material lost
- material Discarded
- loaning_profile_not_found
 material_not_found
- material_part_of_collection
 not_reservable
- no reservable materials
- interlibrary_material_not_reservable
 previously_loaned_by_homebound_patron
- exceeds max reservations

The values are subject to change. If an unrecognized value is encountered, it should be treated as an error.

The reservation detail in the response contains a reservation state, which can be any of these values:

- reserved
- readyForPickup
- inTransit
- other

The values are subject to change. If an unrecognized value is encountered, it should be treated as 'other'.

When making a reservation of a periodical, the values to put in the PeriodicalReservation structure can be obtained from the periodical information retrieved with the Catalog service.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
patronid	the patron that makes the reservations	path	integer
createReservationBatch	the reservations to be created	body	Model Model Schema

CreateReservationBatchV2 {

Swagger UI Parameter Type Parameter Description Data Type reservations (array[CreateReservation]): Reservations with duplicate record id's will be removed., type (string, optional): Will be considered normal if not set } CreateReservation { recordId (string): Identifies the bibliographical record to reserve - The FAUST number, expiryDate (string, optional): The date where the patron is no longer interested in the reserved material. If not set, a date will be calculated from the agency default interest period, pickupBranch (string, optional): ISIL-number of pickup branch. If not set, will default to patrons preferred pickup branch, periodical (PeriodicalReservation, optional): Present if making reservation on a periodical } PeriodicalReservation { volume (string, optional), volumeYear (string, optional), volumeNumber (string, optional) } **Response Class**

Response Class

Model Model Schema

ReservationResponseV2 {

success (boolean): True if all reservations were created successfully otherwise false, **reservationResults** (array[ReservationResultV2]): Result of each reservation

}

ReservationResultV2 {

recordId (string): Recordid of the record to reserve,
 result (string): The reservation result,
 periodical (PeriodicalReservation, optional): Periodical information of the reservation,
 reservationDetails (ReservationDetailsV2, optional): The reservation data as returned by the create/update operation

PeriodicalReservation {

volume (string, optional),
volumeYear (string, optional),
volumeNumber (string, optional)

}

}

ReservationDetailsV2 {

pickupBranch (string): ISIL-number of pickup branch,

pickupDeadline (string, optional): Set if reserved material is available for loan,

dateOfReservation (string),

ilBibliographicRecord (ILLBibliographicRecord, optional): Additional bibliographic information for inter-library loans,

numberInQueue (integer, optional): The number in the reservation queue.,

pickupNumber (string, *optional*): The reservation number. Will be present if the reservation is ready for pickup (the state is 'readyForPickup'), **transactionId** (string): Identifies the transaction of reservations,

recordId (string): The FAUST number,

expiryDate (string): The date when the patron is no longer interested in the reserved material,

reservationId (integer): Identifies the reservation for use when updating or deleting the reservation,

periodical (Periodical, optional): Present if material is a periodical,

reservationType (string),

state (string)

}

ILLBibliographicRecord {

author (string, optional): The author of the material, isbn (string, optional): ISBN-information from the bibliographic record, periodicalNumber (string, optional): Issue number of a periodical, edition (string, optional): Edition-information from the bibliographic record, language (string, optional): Language of the requested material., bibliographicCategory (string, optional): Bibliographic category from danMARC2 008 *t, title (string, optional): The title of the material, publicationDateOfComponent (string, optional): Publication date of an item component, or article., recordId (string): The FAUST number, issn (string, optional): ISSN-information from the bibliographic record, placeOfPublication (string, optional), mediumType (string): Type of the requested material - from danMARC2 009 *a+*g (general and specific), periodicalVolume (string, optional): Volume name of a periodical, publisher (string, optional): Publisher of the requested material.,

		Swagger Of
<pre>publicationDate (s }</pre>	tring, optional): Publication date of	the requested material.
Periodical { volume (string, opt volumeYear (string)		
): A representation of the periodica	volume information that is suitable for display,
}		
Response Content T	Type application/json 🗸	
Response Messag	es	
HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	
404	patron not found	
Placement		Show/Hide List Operations Expand Operations Raw

GET	/external/v1/{	agencyid}/{branchId}/closingdates Ge	et closing dates for a branch i	in an interval, inclusively between startDate and endDate.
	mentation Note s array of closing			
Param	neters			
Param	eter	Description	Parameter Type	Data Type
agenc	yid	ISIL of the agency (e.g. DK-761500)	path	string
branc	hId	ISIL of branch (e.g. DK-761501)	path	string
start	Date	Starting date of the interval in the format yyyy- mm-dd	query	string

		Swagger UI		
Parameter	Description	Parameter Type	Data Type	
endDate	End date of the interval in the format yyyy-mm- dd	query	string	
Response Class				
Model Model Sch	ema			
ClosingDateV1 { branchId (string), description (string closingDate (strin }				
	Type application/json 🗸			
Response Messag	ges			
HTTP Status Code	Reason	Response Model		
400	bad request			
	bad request client unauthorized			
400 401				Get opening hours for a bra
400 401 GET /external. Implementation	client unauthorized /v1/{agencyid}/{branchId}/openinghours Notes			Get opening hours for a bra
400 401	client unauthorized /v1/{agencyid}/{branchId}/openinghours Notes			Get opening hours for a bra
400 401 GET /external. Implementation Returns array of op	client unauthorized /v1/{agencyid}/{branchId}/openinghours Notes	Parameter Type	Data Type	Get opening hours for a bra
400 401 GET /external. Implementation Returns array of op Parameters	client unauthorized /v1/{agencyid}/{branchId}/openinghours Notes pening hours.	Parameter Type path	Data Type string	Get opening hours for a bra

5/5/25,

Model Model Schema

OpeningHoursV1 {

branchId (string): ISIL of branch (e.g. DK-761501),

serviceHoursStartTime (string): The opening time for the service hours Format: HH:MM:SS,

serviceHoursEndTime (string): The closing time for the service hours Format: HH:MM:SS,

openHoursEndTime (string): The closing time for the opening hours Format: HH:MM:SS,

day (string) = ['MONDAY' or 'TUESDAY' or 'WEDNESDAY' or 'THURSDAY' or 'FRIDAY' or 'SATURDAY' or 'SUNDAY']: The day of the specified opening hours, **openHoursStartTime** (string): The opening time for the opening hours. Format: HH:MM:SS

}

Response Content Type application/json ~

Response Messages

HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

GET	/external/v1/{agencyid}/branches	
UL:	, excernal, in [agency a], brancies	

Get branches for an agency.

Implementation Notes

Returns array of branches.

Can be used for giving the patron the option of choosing a preferred branch or where to pick up reservations.

Parameters

Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
exclude	Identifies the branchIds which are excluded from the result	query	array[string]
Response Class			

Model Model Sch		Swagger UI		
model model bei	nema			
AgencyBranch {				
	: ISIL of branch (e.g. DK-761501),			
title (string): Nam	ne of branch			
}				
D	-			
Response Content	Type application/json ✓			
Response Messa	ges			
HTTP Status Code	Reason	Response Model		
400	bad request			
401	client unauthorized			
GET /externa	l/v1/{agencyid}/branches/contact			Get contact information for branche
Implementation	Notes			
Implementation Returns a list of b				
Implementation Returns a list of b				
Returns a list of b		Parameter Type	Data Type	
Returns a list of b Parameters	ranch contacts.	Parameter Type path	Data Type string	
Returns a list of be Parameters Parameter	ranch contacts. Description			
Returns a list of b Parameters Parameter agencyid branchIds	Description ISIL of the agency (e.g. DK-761500)	path	string	
Returns a list of b Parameters Parameter agencyid branchIds Response Class	ranch contacts. Description ISIL of the agency (e.g. DK-761500) List of branch IDs to get contact information for	path	string	
Returns a list of b Parameters Parameter agencyid branchIds	ranch contacts. Description ISIL of the agency (e.g. DK-761500) List of branch IDs to get contact information for	path	string	
Returns a list of b Parameters Parameter agencyid branchIds Response Class	ranch contacts. Description ISIL of the agency (e.g. DK-761500) List of branch IDs to get contact information for	path	string	
Returns a list of br Parameters Parameter agencyid branchIds Response Class Model Model Sch BranchContact { branchId (string);	ranch contacts. Description ISIL of the agency (e.g. DK-761500) List of branch IDs to get contact information for	path	string	
Returns a list of br Parameters Parameter agencyid branchIds Response Class Model Model Sch BranchContact { branchId (string); zipCode (string);	ranch contacts. Description ISIL of the agency (e.g. DK-761500) List of branch IDs to get contact information for	path	string	
Returns a list of br Parameters Parameter agencyid branchIds Response Class Model Model Sch BranchContact { branchId (string): zipCode (string): website (string):	ranch contacts. Description ISIL of the agency (e.g. DK-761500) List of branch IDs to get contact information for	path	string	

		Swagger Of	
phone (string): Pl title (string): Nan	where the branch is located, hone number of the branch, ne of the branch, nail address of the branch		
}			
•			
Response Conten	t Type application/json 🗸		
Response Messa	ages		
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
GET /externa	l/v1/{agencyid}/departments	Go	t translations from department identifiers to displayable text.
Vexterna	, vir (agencyla), acpartments		
Implementation	Notes		
Returns array of o	lepartments.		
_			
Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
Response Class			
Model Model Scl	nema		
AgencyDepartmer			
	tring): Department identifier,		
	ne of the department		
}			
Response Conten	t Type application/json ✔		
Response Messa	ages		

5/5/25,	1:11 PM	
---------	---------	--

PM		Swagger UI	
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
GET /external/	v1/{agencyid}/locations		Get translations from location identifiers to displayable tex
Implementation N	lotes		
Returns array of loc			
Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
Response Class			
Model Model Sche	ma		
AgencyLocation { locationId (string): title (string): Name }			
Response Content 1	ype application/json 🗸		
Response Messag	es		
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		

Get translations from section identifiers to displayable text.

1 PM		Swagger UI	
Implementation N	lotes		
Returns array of sec			
Returns anay or sec			
Parameters			
Parameter	Description	Parameter Type	Data Type
agencyid	ISIL of the agency (e.g. DK-761500)	path	string
Response Class			
Model Model Scher	ma		
AgencySection {			
sectionId (string): S	ection identifier,		
title (string): Name	of the section		
}			
Response Content T	ype application/json 🗸		
Response Message	25		
HTTP Status Code	Reason	Response Model	
400	bad request		
401	client unauthorized		
GET /external/\	/1/{agencyid}/sublocations	Get t	translations from sub-location identifiers to displayable text.
Implementation N	lotes		
Returns array ofsub	-locations.		
Parameters			
Parameter	Description	Parameter Type	Data Type
	· · · · · · · · · · · · · · · · · · ·		
agencyid	ISIL of the agency (e.g. DK-761500)	path	string

Response Class		
Model Model Scher	ma	
AgencySublocation {		
title (string): Name	of the sub-location,	
	ng): Sub-location identifier	
}	-	
Response Content T	ype application/json 🗸	
Dochonco Moscog		
Response Message	25	
HTTP Status Code	Reason	Response Model
400	bad request	
401	client unauthorized	

[BASE URL: http://localhost:8080/externalapidocs/apidocs , API VERSION: 3]