INN-Reach D2IR API Reference

Version 2.3

© 2020

innovative

World Headquarters
1900 Powell Street, Suite 400
Emeryville, CA 94608
+1.510.655.6200

Legal Notices

© Innovative Interfaces Incorporated and/or its affiliates. All rights reserved. All trademarks shown are the property of their respective owners.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

The software and related documentation are provided under an agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of the software, unless required by law for interoperability, is prohibited.

Contents

INN-Reach Direct to INN-Reach (D2IR) API **6**

INN-Reach system overview 6

Participation 7

API authentication **8**

Authentication errors 9

Third-party authentication 9

HTTP header elements **10**

Request properties 11

Response structure **12**

Response content 12

Local Server, Agency, and Location **15**

System Configuration **17**

Delete a Single Location 17

Get Central Item Types 18

Get Central Patron Types 19

Get Local Servers 21

Retrieve Locations List 23

Upload Locations List 24

Upload/Update a Single Location 25

Record contribution **27**

Contribute Batch Items 27

Contribute Bib 29

Decontribute Bib 31

Decontribute Item 32

Get Base64 Encoding Table 32

Look Up Bib by bibId 33

Look Up Bib by bibId and itemId 35

© 2020 Innovative Interfaces, Inc.

Update Bib Status 37
Update Item Status 38
Contribution error messages 39
Circulation **41**
Circulation overview 42
Initiating transactions 42
D2IR call sequence use case scenarios 44
Circulation transaction and states 46
Implementing endpoints 51
Terminating transactions 52
Outgoing circulation message failures 52
APIs implemented by Innovative 53
Borrower Renew Item 53
Cancel Item Hold 54
Final Item Check-In 54
Item In Transit 55
Item Shipped 56
Local Checkout 57
Owner Renew Item 58
Owning Site Cancels Request 58
Patron Claims Returned 59
Recall Item 60
Report Item Received 61
Report Unshipped Item Received 62
Return Uncirculated Item 63
Transfer Request 64
APIs implemented by third-party ILS 65
Borrower Renew Item 65
Cancel Item Hold 66
Cancel Request 67
Create Item Hold 67
Create Local Hold 69

© 2020 Innovative Interfaces, Inc.

Create Patron Hold 71
Final Item Check-In 72
Item In Transit 73
Item Shipped 74
Owner Renew Item 75
Patron Claims Returned 76
Recall Item 77
Report Item Received 78
Report Unshipped Item Received 79
Return Uncirculated Item 80
Transfer Request 81
Verify Patron 82
Get Bib Record 83
Get Status 84
Glossary **85**

© 2020 Innovative Interfaces, Inc.

INN-Reach Direct to INN-Reach (D2IR) API

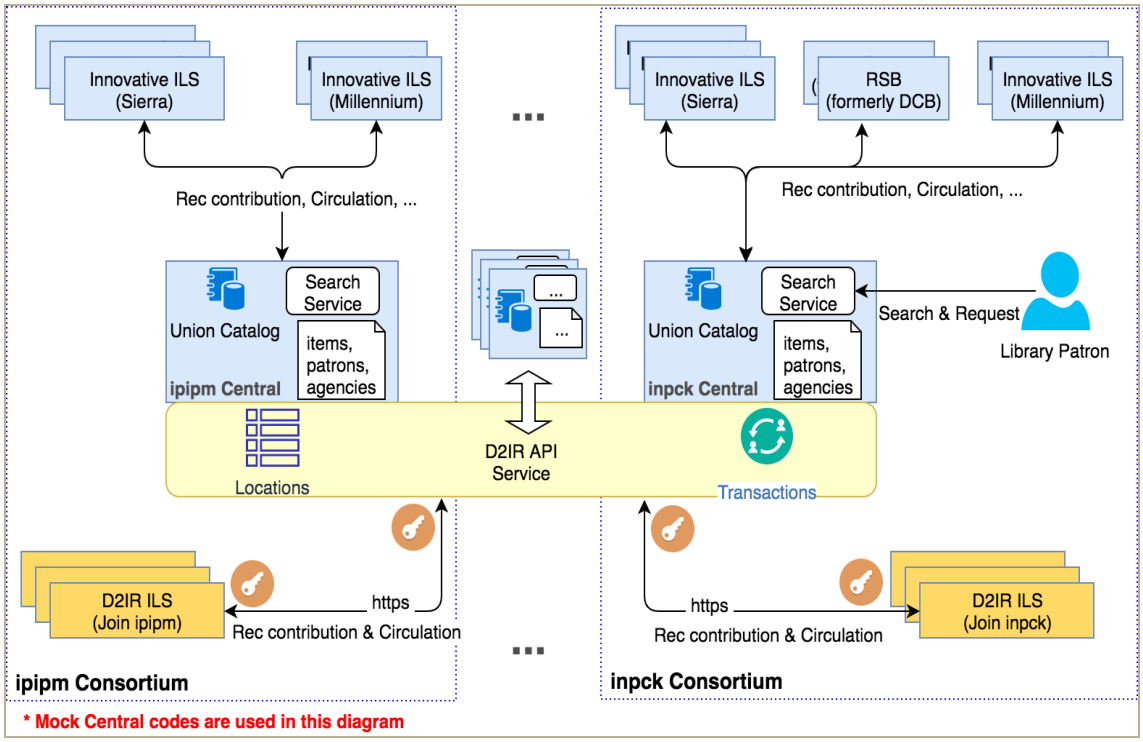
This reference manual describes methods used to facilitate communication between different INN-Reach system components. INN-Reach enables participating libraries to offer materials to other libraries in a consortial borrowing system. Patrons of participating libraries can directly request and borrow materials from other participating libraries through the union catalog. To help facilitate this, Innovative provides a set of application programming interfaces (APIs) that:

- circulate material between participating sites
- update information related to cataloged material
- manage records stored in the union catalog and participant local catalogs

In addition to API functionality, this reference also discusses security, authentication, and authorization.

INN-Reach system overview

The following diagram depicts how communication is passed between different INN-Reach system components in an INN-Reach consortium. As shown below, multiple consortia are accessible from the same D2IR service. Both the Innovative and non-Innovative ILSs are clustered near the consortia they join.



© 2020 Innovative Interfaces, Inc.

Libraries participating in an INN-Reach consortium contribute records to the central union catalog. When local records are updated, those in the union catalog are also updated. When patrons request materials from the union catalog, appropriate transactions are recorded in both borrowing and lending (owning) local libraries, and the record status is updated in both the local catalog and the union catalog. The INN-Reach central server performs patron validation at the local library. All transactions are accomplished automatically, in real time.

The INN-Reach D2IR API allows libraries using any non-Innovative integrated library software system to participate in an INN-Reach resource-sharing consortium. For more information, see Participation, below.

See Glossary for definitions of INN-Reach terms.

Participation

If you use a non-Innovative integrated library software system and are interested in participating in an INN-Reach resource-sharing consortium, contact Innovative Product Management.

You will need to provide authentication information to try the INN-Reach API. See "API authentication" on page 8.

© 2020 Innovative Interfaces, Inc.

API authentication

D2IR API requests are authenticated with an oauth2 token. An API key and secret are used to request access tokens from the INN-Reach platform.

The oauth2 endpoint uses the POST method to obtain access tokens. The client credential is provided in the HTTP authorization header with an HTTP basic authentication type. Both "grant_type" and "scope" are required parameters. The grant type is “client_credential” while “innreach_tp” is the scope for third-party users.

The following cURL sample requests an access token from our sandbox environment.

```
curl -X POST https://rssandbox-api.iii.com/auth/v1/oauth2/token -H
'Authorization: Basic YjU0ZTU2YzgtMGNlNi00MzhjLTk3NzktYzcyMWIxY2RjMz
ZhOjFjYjQ4YjNmLTE1OTAtNDc0YS1iMDhhLWZjNDRlMjczMTlmOQ=='
   -H 'Content- Type: application/x-www-form-urlencoded'
   -d 'grant_type=client_credentials&scope=innreach_tp'
```

Sample response:

```
{
   "access_token": "06b42df732f628ae2c3764d86073cd76",
   "token_type": "Bearer",
   "expires_in": 599
}
```

The provided key and secret should be concatenated using a colon separator, and the resultant string base64 encoded.

The access token is valid for API requests until it expires in 600 seconds. An expired token will result in a “401 Unauthorized” HTTP status code in the response.

> **Note:** The oauth2 endpoint to get an access token on the production environment is issued only upon request.

© 2020 Innovative Interfaces, Inc.

Authentication errors

When requesting access tokens, any missing or incorrect required property will result in a “400 Bad Request” in the response. A “401 Unauthorized” error is issued if client credentials are provided incorrectly.

```
400 Bad Request
{
   "error": "invalid_request",
   "error_description": "The scope parameter is required."
}

401 Unauthorized
{
   "error": "invalid_token",
   "error_description": "Token authentication failed"
}
```

Third-party authentication

To secure an API, third-party developers must take the same approach as described in API Authentication above. An equivalent oauth2 endpoint needs to be provided to produce a bearer token from the INN-Reach platform.

To join an INN-Reach central consortium, the following information must be provided during setup.

- The oauth2 endpoint to get an access token for requesting the APIs implemented by the third party.
- The INN-Reach credentials used to generate the access token.
- INN-Reach parameters required to specify when calling the oauth2 endpoint. For example: `grant_type=client_credentials&scope=innreach_tp`

  The following is an example call that would originate from Innovative to the third party.

```
curl -X POST http://rd-mock.polarislibrary.com/innreach/v2/oauth2/
token -H 'Accept: application/json'
   -H 'Authorization: Basic
   ZmQ1OTdlXmItMjNkLS00ZWZiLAE0ZjUtN2Y2NJFhNDc2Njk2OjVhMjcxMjdhLUk0YmUtNG
   JhYi05MDMzLWI1MmFjMzkxOTQ3Yg=='
   -H 'Content-Type: application/x-www-form-urlencoded'
   -d 'grant_type=client_credentials&scope=innreach_tp'
```

© 2020 Innovative Interfaces, Inc.

HTTP header elements

The following HTTP headers are necessary for D2IR API methods.

Property	Type	Required	Description
Accept	String	Yes	Acceptable value: application/json
Authorization	String	Yes	Authorization in form <token_type> <access_token> as returned by the authorization request
Content-Type	String	Yes	Acceptable value: application/json
X-From-Code	String	Yes	Central/local code of sending system (five-character code). The library needs only its own local code and the code for any central systems in which it participates.
X-Request-Creation-Time	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
X-To-Code	String	Yes	Central/local code of destination system (five-character code). The library needs only its own local code and the code for any central systems in which it participates.

© 2020 Innovative Interfaces, Inc.

Request properties

Request properties are sent as an HTTP request body in JSON format, which is the only format currently supported in D2IR.

Example: ContributeBatch Item


```
{
   "itemInfo":[
      {
         "agencyCode":"plag2",
         "callNumber":"940.53 Ien Ien",
         "copyNumber":null,
         "dueDateTime":null,
         "holdCount":0,
         "itemCircStatus":"Available",
         "itemId":"2475459",
         "itemNote":null,
         "centralItemType":14,
         "locationKey":"test1",
         "marc856PublicNote":null,
         "marc856URI":null,
         "suppress":"n",
         "volumeDesignation":null
      }
   ]
}
```


Example: Item Shipped


```
{
   "itemBarcode": "ibare34"
}
```


© 2020 Innovative Interfaces, Inc.

Response structure

D2IR responses use HTTP status code and content body to inform API users of the request processing status.

HTTP Status Code	Description	D2IR Meaning
200	OK	All request properties are valid and successfully serviced in the central server. However, the final request status can still be a "failed" due to other reason.
400	Bad Request	The request either does not contain all the required properties or has invalid value(s) in one or more properties.
401	Unauthorized	The request does not have the appropriate authentication token in the HTTP Authorization header.
405	Method Not Allowed	The request is not submitted with the correct HTTP method. For example, using GET instead of POST as required by the specific API.
500	Internal Server Error	The central encountered problems that are not one of the above while servicing the request.
501	Not Implemented	The server does not support the functionality to fulfill the request.
503	Service Unavailable	The server is unavailable.

Response content

The response content is a JSON-formatted string.

Contribution requests are processed synchronously. The response status reflects the status of the request.

Circulation requests are validated and submitted to the central queue for processing. The response status is *not* the final status of the request. An "`ok`" status means the request processing queue accepted the request.

Property	Description
status	Either "ok" or "failed" for a string type.
reason	A short description about the final request status.
errors	The list of problems encountered in servicing the request.

© 2020 Innovative Interfaces, Inc.

Failed response examples

```
{
   "status":"failed",
   "reason":"Authorization token does not belong to 'tploc'",
   "errors":[]
}
```

```
{
   "status":"failed",
   "reason":"Unknown centralCode and trackingId combination",
   "errors":[
      {
         "type":"FieldError",
         "reason":"Invalid record key",
         "name":"centralCode",
         "rejectedValue":"pcent"},
      {
         "type":"FieldError",
         "reason":"Invalid record key",
         "name":"trackingId",
         "rejectedValue":"83748742"
      }
   ]
}
```

```
{
   "status":"failed",
   "reason":"Invalid request",
   "errors"[
{
   "type":"FieldError",
   "name":"patronAgencyCode",
   "rejectedValue":"plag1234",
   "reason":"Invalid Field Value"
},
{
   "type":"FieldError",
   "name":"patronName",
   "rejectedValue":null,
   "reason":"Field missing"
},
{
```

© 2020 Innovative Interfaces, Inc.

```
   "type":"HeaderError",
   "name":"X-From-Code",
   "rejectedValue":"ploc8",
   "reason":"Invalid Header Value"
},
{
   "type":"HeaderError",
   "name":"X-To-Code",
   "rejectedValue":"d2ir",
   "reason":"Invalid Header Value"
}
```


© 2020 Innovative Interfaces, Inc.

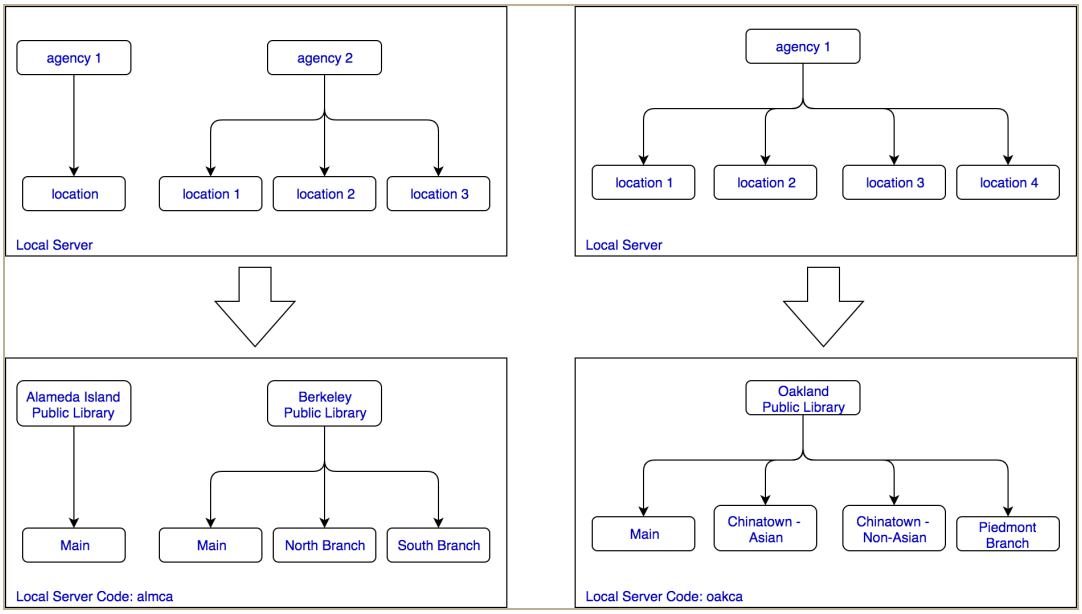
Local Server, Agency, and Location

This section discusses the relationship between the local server, agency, and location and how these work within an ILS and associated consortium.

These three elements within a consortium are defined as follows:

- **Local server:** The computer assigned to an ILS that manages one or more libraries in the consortium.
- **Agency:** A mapping that contains a library or multiple libraries managed by a non-Innovative ILS connected to an INN-Reach Central system.
- **Location:** An ILS-defined name that represents one or more library branches or shelving locations.

As shown in the diagram below, an agency is mapped to a library when multiple libraries are managed by a non-Innovative ILS in an INN-Reach Central system. When a library is mapped to an agency, its location is mapped to library branches. Each branch may contain shelving locations.



A local server code is assigned to an ILS that manages one or more libraries. Each of these libraries may contain multiple branches. An agency maps to a library to differentiate it from other libraries within the INN-Reach Central system. For each library mapped to an agency, its location is assigned to library branches and their associated shelving locations.

The diagram above shows the local server code assigned to an ILS. The ILS can manage one or more libraries with each library possibly containing multiple branches.

The agency, associated with a library, is assigned an agency code that must be unique within the consortium. Unlike the local server code or agency code, locations are defined by the ILS and represent library branches or shelving locations.

© 2020 Innovative Interfaces, Inc.

A location contains a location key and description field. If the location key is within the same consortium, it does not need to be unique. A locationKey within the same local server must be unique and cannot share the same name as another locationKey on the server. In all cases, the agency code needs to be unique, since the local server can participate in multiple consortia.

Before a record can be contributed to the Central, the ILS needs to define each location and upload them to the Central, since the location key is required for the Contribute Batch Items API.

© 2020 Innovative Interfaces, Inc.

System Configuration

Local sites can configure locations. A list of central item types, central patron types, and local servers can also be queried.

The APIs in this section include:

- "Delete a Single Location" on page 17
- "Get Central Item Types" on page 18
- "Get Central Patron Types" on page 19
- "Get Local Servers" on page 21
- "Retrieve Locations List" on page 23
- "Upload Locations List" on page 24
- "Upload/Update a Single Location" on page 25

Delete a Single Location

Deletes a single location matching <locationKey>.

DELETE /innreach/v2/location/<locationKey>

Notes: A URL can be submitted with a location key that matches a previously deleted location. This reinstates the location with the new description and original key.

URL parameters

Property	Description
locationKey	Unique code that represents the location being deleted. Lowercase alphanumeric, maximum 5 characters.

Request parameters

None

Sample cURL call

```
curl -X DELETE http://{host}:{port}/innreach/v2/location/{locationKey} -
H 'Accept: application/json'
   -H 'Authorization: Bearer f071cb6c9220434c0e349cd9d69892e9'
   -H 'X-From-Code: {localcode}' -H 'X-To-Code: {centralcode}'
```

© 2020 Innovative Interfaces, Inc.

Get Central Item Types

An Innovative "item type" combines material type and loan rule/period to determine who can borrow an item and for how long. Central item types are required for item contribution. See "Contribute Batch Items" on page 1

This method requests a list of central server item type values with descriptions, which are required by a local library to set up a mapping table between local item types and those accepted by central.

GET /innreach/v2/contribution/itemtypes

URL parameters

None

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes
ItemTypeList	[ ]	Conditional

ItemTypeList

Property	Type	Required
centralItemType	Integer	Yes
description	String	Yes

Sample cURL call

```
curl -X GET http://{host}:{port}/innreach/v2/contribution/itemtypes -H
'Accept: application/json'
   -H 'Authorization: Bearer 741619d4e1b2156b98e084a08b5a00d9'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

Sample Response:

```
{
   "status":"ok",
   "reason":"success",
   "errors":[],
```

© 2020 Innovative Interfaces, Inc.

```
    "itemTypeList":[
       {
          "centralItemType":0,
          "description":"INN-Reach zero itype"
       },
      ...
       {
          "centralItemType":255,
          "description":"Poof and piffle 12/7 5:25"
       }
    ]
}
```

Get Central Patron Types

Requests a list of central server patron type values with descriptions, which are required by a local library to set up a mapping table between their local patron types and those acceptable by central.

GET /innreach/v2/circ/patrontypes

Note: An Innovative *patron type* is required to determine who can borrow an item and for how long.

URL parameters

None

Request parameters

None

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes
patronTypeList	[ ]	Conditional

© 2020 Innovative Interfaces, Inc.

patronTypeList

Property	Type	Required
centralPatronType	Integer	Yes
description	String	Yes

Sample cURL call

```
curl -X GET http://{host}:{port}/innreach/v2/circ/patrontypes -H
'Accept:
application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

Sample Response:

```
{
   "status": "ok",
   "reason": "success",
   "errors": [],
   "patronTypeList": [
      {
      "centralPatronType": 15,
      "description": "INN-Reach ptype 15"
      },
      …
      {
      "centralPatronType": 210,
      "description": "Central two ten"
      }
   ]
}
```

© 2020 Innovative Interfaces, Inc.

Get Local Servers

Requests a list of all local server agency codes with descriptions.

`GET` /innreach/v2/contribution/localservers

URL parameters
None

Request parameters
None

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes
localServerList	[ ]	Optional

localServerList

Property	Type	Required	Description
localCode	String	Yes	5 character code
description	String	Yes	128 character maximum
agencyList	[ ]	Optional	

agencyList

Property	Type	Required	Description
agencyCode	String	Yes	5 character code
description	String	Yes	128 character maximum

Sample cURL call

```
curl -X GET http://{host}:{port}/innreach/v2/localservers
   -H'Accept:application/json'
   -H' Authorization:Bearer2af564e268eae1375c44e8c25fc77348'
   -H' X-From-Code:{localcode}'
```

© 2020 Innovative Interfaces, Inc.

```
   -H' X-To-Code:{centralcode}'
```

Sample Response:


```
{
   "status": "ok",
   "reason": "success",
   "errors": [],
   "localServerList": [
      {
         "localCode": "9irv1",
         "description": "irv system",
         "agencyList": [
            {
               "agencyCode": "9aaaa",
               "description": "Appalachian"
            },
            {
               "agencyCode": "9bbbb",
               "description": "Irv Big Bend"
            },
            ...
         ],
      },
      {
         "localCode": "9loc1",
         "description": "IRBOX 1, irdvloc1",
         "agencyList": [
            {
               "agencyCode": "9bada",
               "description": "Bad Axe Public Library"
            },
            {
               "agencyCode": "9bayl",
               "description": "Bayliss Public Library"
            },
            ...
         ],
      },
      ...
   ]
}
```


© 2020 Innovative Interfaces, Inc.

Retrieve Locations List

Requests a list of submitted location codes with descriptions.

GET /innreach/v2/contribution/locations

URL parameters

None

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes
locationList	[ ]	Yes

locationList

Property	Type	Required	Description
locationKey	String	Yes	Unique code that represents the location. Lowercase alphanumeric, maximum 5 characters.
description	String	Yes	255 char max

Sample cURL request

```
curl -X GET http://{host}:{port}/innreach/v2/contribution/locations -H
'Accept: application/json'
   -H 'Authorization: Bearer 39d0854c74df7ddf9a54ad56fa9af15c'
   -H 'X-From-Code: jclcl'
   -H 'X-To-Code: pcent'
```

Sample response:

```
{
   "status":"ok",
   "reason":"success",
   "locationList":[
      {
         "locationKey":"main",
         "description":"Main Library"
```

© 2020 Innovative Interfaces, Inc.

```
      },
     ...
      {
         "locationKey":"wcdpl",
         "description":"Wood County District Public Library"
      }
   ],
   "errors":[]
}
```

Upload Locations List

Submits a list of location keys with descriptions. Replaces existing locations with a new set.

POST /innreach/v2/contribution/locations

URL parameters

None

Request parameters

Property	Type	Required
locationList	[]	Yes

locationList

Property	Type	Required	Description
locationKey	String	Yes	Unique code that represents the location. Lowercase alphanumeric, maximum 5 characters.
description	String	Yes	255 char max

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/contribution/locations - H
'Accept: application/json'
   -H 'Authorization: Bearer b50cb855e0a24c69f6fe-b14dd6282f95'
   -H 'Content-Type: application/json' -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{ "locationList": [
         {
```

© 2020 Innovative Interfaces, Inc.

```
            "locationKey":"main",
            "description":"JCLCL Main Library"
         },
         {
            "locationKey":"dnvl",
            "description":"JCLCL Danville Branch"
         }
      ]
   }
```

Upload/Update a Single Location

Adds a single new location or updates the description of an existing location with a matching <locationKey>.

POST	/innreach/v2/location/<locationKey>
PUT	/innreach/v2/location/<locationKey>

URL parameters

Property	Description
locationKey	Unique code that represents the location being uploaded or updated. Lowercase alphanumeric, maximum 5 characters.

Request parameters

Property	Type	Required	Description
description	String	Yes	255 char max

Sample cURL calls

POST creates a new location or updates and existing location.

```
curl -X POST http://{host}:{port}/innreach/v2/location/{locationKey} -H
'Accept: application/json'
   -H 'Authorization: Bearer 7c7b93073f829c24a6a2d628005cfe1d'
   -H 'Content-Type: application/json'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{
      "description": "JCLCL Danville Branch MOD..."
      }'
```

© 2020 Innovative Interfaces, Inc.

PUT updates an existing location.

```
curl -X PUT http://{host}:{port}/innreach/v2/location/{locationKey} -H
'Accept: application/json'
   -H 'Authorization: Bearer 7c7b93073f829c24a6a2d628005cfe1d'
   -H 'Content-Type: application/json'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{
      "description": "JCLCL Danville Branch MOD ..."
      }'
```

Failed status response

The following error will return if an incorrect or nonexistent locationKey is used:

```
{
   "status": "failed",
   "reason": "Record not found",
   "errors": []
}
```

© 2020 Innovative Interfaces, Inc.

Record contribution

Local sites contribute, decontribute, and update records with the methods listed here.

The APIs in this section include:

- "Contribute Batch Items" on page 27 (on this page)
- "Contribute Bib" on page 29
- "Decontribute Bib" on page 31
- "Decontribute Item" on page 32
- "Get Base64 Encoding Table" on page 32
- "Look Up Bib by bibId" on page 33
- "Look Up Bib by bibId and itemId" on page 35
- "Update Bib Status" on page 37
- "Update Item Status" on page 38
- "Contribution error messages" on page 39

Contribute Batch Items

Add and modify item records.

> **Note:** If one or more items submitted for update with this method already exist in the central database, but do not belong to the bib specified in the URL, then the entire batch update will fail, with a status message of **statmsg=172:<bibId>:One or more items in this batch are not linked to the specified local bib**.

POST /innreach/v2/contribution/items/<bibId>

URL parameters

Property	Description
bibId	ID of the bibliographic record. Lowercase alpha-numeric string, maximum 32 characters.

Request parameters

Property	Type	Required	Description
itemInfo	[ ]	Yes	list of itemInfo (maximum count 100)

© 2020 Innovative Interfaces, Inc.

itemInfo

Property	Type	Required	Description
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
agencyCode	String	Yes	5 character lowercase alphanumeric
centralItemType	Integer	Yes	Valid values provided by central (central-specific). See "Get Central Item Types" on page 1.
locationKey	String	Yes	Lowercase alphanumeric, maximum 5 characters. Validated against keys sent via Post Locations endpoints
itemCircStatus	String	Yes	Available/Not Available/On Loan/Non-Lendable
holdCount	Integer	No	Item hold count, maximum 99
dueDateTime	Integer	No	Epoch UNIX time stamp, for example: 1544466568
callNumber	String	No	Maximum 128 characters
volumeDesignation	String	No	Maximum 32 characters
copyNumber	Integer	Conditional	Only if nonzero
marc856URI	String	Conditional	Maximum 512 bytes (valid URL format) DO NOT include if there is a marc856URI in the contributed bib record that applies to this Item. That is, include only if there is an 856 URI that applies specifically to this item.
marc856PublicNote	String	Contitional	Maximum 64 characters. Include only if sending marc856URI for this item.
itemNote	String	No	Maximum 256 characters
suppress	String	No	ASCII 'y' or 'n' only. Default: n

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/contribution/items/{bibId}
-H 'Accept: application/json'
   -H 'Authorization: Bearer 6e1b45d436cac9114f1625f38e463e5a'
   -H 'Content-Type: application/json'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{
         "itemInfo":[
      {
         "itemId":"2475459",
         "agencyCode":"plag2",
         "centralItemType":14,
         "locationKey":"test1",
```

© 2020 Innovative Interfaces, Inc.

```
          "itemCircStatus":"Available",
          "callNumber":"940.53 Ien",
          "copyNumber":null,
          "dueDateTime":null,
          "holdCount":0,
          "itemNote":null,
          "marc856PublicNote":null,
          "marc856URI":null,
          "suppress":"n",
          "volumeDesignation":null
        },
        {
          "itemId":"3242345",
          "agencyCode":"plag2",
          "centralItemType":14,
          "locationKey":"test1",
          "itemCircStatus":"Available",
          "callNumber":"950.22 KJK",
          "copyNumber":null,
          "dueDateTime":null,
          "holdCount":0,
          "itemNote":null,
          "marc856PublicNote":null,
          "marc856URI":null,
          "suppress":"n",
          "volumeDesignation":null
        }
    ]
}'
```

Contribute Bib

Adds and modifies bib records. The bibId can be used as the local system's unique bib identifier. Local sites that have nonconforming record identifiers will need to have those identifiers converted to match the D2IR bibId restrictions described in the table below. The submitted bib ID will become the unique INN-Reach catalog bib identifier.

> **Note:** Multiple agencies can share the same bib. When this occurs, the same bibId must be used for the central reference to this bib by all the sharing agencies. The bib should be contributed only once, regardless of the number of sharing agencies. The bib's agency affiliations are reflected on the central by its contributed holdings, as separate items will be contributed by the various agencies.

POST /innreach/v2/contribution/bib/<bibId>

© 2020 Innovative Interfaces, Inc.

URL parameters

Property	Description
bibId	ID of the bibliographic record. Lowercase alpha-numeric string, maximum 32 characters.

Request parameters

Property	Type	Required	Description
marc21BibFormat	String	Yes	Currently supports "ISO2709"
marc21BibData	String	Yes	Base 64 encoded MARC data (maximum 99999 bytes before base 64 encoding). See "Get Base64 Encoding Table" on page 32. • Only standard MARC fields accepted • Must include 008 and 245 fields • Omit 9XX fields • Strip data with local meaning only or that you do not want to expose
titleHoldCount	Integer	Yes	Number of title level holds
itemCount	Integer	Yes	Number of items linked to this bib
suppress	String	No	ASCII y, n, or g; "g" means display the resource in the discovery layer as having been provided by the consortium, not by the individual site. Default: n

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/contribution/bib/{bibId} -
H 'Accept: application/json'
   -H 'Authorization: Bearer 7b9be3a7b4e5eeae96f4779e34c80662'
   -H 'Content-Type: application/json'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{
      "bibId":"38675",
      "agencyCode":"plag5",
      "marc21BibFormat":"ISO2709",
      "titleHoldCount": 0,
      "itemCount": 0,
      "marc21BibData":"MDA5ODduYW0gYTIyMDAyNTMgYSA0NTAwMDAxMDAwNzAwMDAwMD
      A1MDAxNzAwMDA3MDA4MDA0MTAwMDI0MDIwMDAyNTAwMDY1MDkyMDAwODAwMDkwMDk5M
      DAwODAwMDk4MTAwMDAxODAwMTA2MjQ1MDAzMzAwMTI0MjYwMDA0NzAwMTU3MzAwMDAy
      MTAwMjA0NTIwMDI5NDAwMjI1NjUwMDAyMTAwNTE5NjUwMDAzMTAwNTQwNjUwMDAyNzA
      wNTcxNjUwMDAyMjAwNTk4NjUwMDAyMzAwNjIwNjUwMDA0NjAwNjQzNjUxMDAzMzAwNj
      g5OTE0MDAxMTAwNzIyHjczMTE1Mh4yMDE0MDMwODAyMDMxOS4wHjA3MDwN3MyMDA3IC
```

© 2020 Innovative Interfaces, Inc.

```
        AgIG55dSAgICBlICAgICAgMDAwIDEgZW5nIHUeICAfYTA2MTg3NzcxM1gOh9jJDE2Lj
        AwHiAgH2JEZXUeICAfYkRldR4xIB9hRGV1a2VyLCBDYXJsLh4xMB9hR3ltIGNhbmR5I
        C8fY2J5IENhcmwgRGV1a2VyLh4gIB9hQm9zdG9uIDofYkhvdWdodG9uIE1pZmZsaW4g
        Q29tcGFueSwfYzIwMDcuHiAgH2EzMTMgcC4gOx9jMjIgY20uHiAgH2FHcm9vbWVkIGJ
        5IGhpcyBmYXRoZXIgdG8gYmUgYSBzdGFyIHBsYXllciwgZm9vdGJhbGwgaXMgdGhlIG
        9ubHkgdGhpbmcgdGhhdCBoYXMgZXZlciByZWFsbHkgbWF0dGVyZWQgdG8gTWljayBKb
        2huc29uLCB3aG8gd29ya3MgaGFyZCBmb3IgYSBzcG90IG9uIHRoZSB2YXJzaXR5IHRl
        YW0gaGlzIGZyZXNobWFuIHllYXIsIHRoZW4gdHJpZXMgdG8gaG9sZCBvbnRvIGhpcyB
        lZGdlIGJ5IHVzaW5nIHN0ZXJvaWRzLCBkZXNwaXRlIHRoZSBjb25zZXF1ZW5jZXMgdG
        8gaGlzIGhlYWx0aCBhbmQgc29jaWFsIGxpZmUuHiAwH2FGb290YmFsbCBzdG9yaWVzH
        iAwH2FGYXRoZXJzIGFuZCBzb25zH3ZGaWN0aW9uLh4gMB9hSGlnaCBzY2hvb2xzH3ZG
        aWN0aW9uLh4gMB9hU2Nob29scx92RmljdGlvbi4eIDAfYVN0ZXJvaWRzH3ZGaWN0aW9
        uLh4gMB9hRmFtaWx5IGxpZmUfeldhc2hpbmd0b24gKFN0YXRlKR92RmljdGlvbi4
        eIDAfYVdhc2hpbmd0b24gKFN0YXRlKR92RmljdGlvbi4eICAfYTczMTE1Mh4d"
        }'
```

Decontribute Bib

Removes a bib record. If a bib is removed, it removes all contributed items.

DELETE /innreach/v2/contribution/bib/<bibId>

URL parameters

Property	Description
bibId	ID of the record being removed. Lowercase alpha-numeric string, maximum 32 characters

Request parameters

None

Sample cURL call

```
curl -X DELETE http://{host}:{port}/innreach/v2/contribution/bib/{bibId}
-H 'Accept: application/json'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

© 2020 Innovative Interfaces, Inc.

Decontribute Item

Removes an item record.

DELETE /innreach/v2/contribution/item/<itemId>

URL parameters

Property	Description
itemId	ID of the record being removed. Lowercase alphanumeric string, maximum 32 characters

Request parameters

None

Sample cURL call

```
curl -X DELETE http://{host}:{port}/innreach/v2/contribution/item/
{itemId} -H 'Accept: application/json'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

Get Base64 Encoding Table

Returns a 64-character string that contains the ordered character set used when submitting Base64 encoded MARC data. **See** "Contribute Bib" on page 29.

GET /innreach/v2/base64encoding

URL parameters

None

Response content

Property	Description
encodingMap	64 ASCII character mapping string (excludes '=' padding character)

© 2020 Innovative Interfaces, Inc.

Sample cURL call

```
curl -X GET http://{host}:{port}/innreach/v2/base64encoding -H 'Accept:
application/json'
   -H 'Authorization: Bearer 36a13bb1ceccca6f7a6958d572a1ef8a'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

Sample Response:

```
{
   "status": "ok",
   "reason": "success",
   "errors":[],
   "encodingMap":
   "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
}
```

Look Up Bib by bibId

This method allows a client to look up a bibliographic record by its ID.

The central returns the following status codes:

- 200 - The record was found, and the corresponding central record has been loaded.
- 404 - The record was not found in the "z" section of the i-file index.
- 500 - An error exists in the milproto format, or a system error occurred.

GET /innreach/v2/local/<localCode>/bib/<bibId>

URL parameters

Property	Description
localCode	Unique code that represents the local server.
bibId	The ID of the bibliographic record.

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[]	Yes

© 2020 Innovative Interfaces, Inc.

Sample cURL calls

Three sample cURL calls are provided with different responses.

- Successful call resulting in a 200 status code
- Unsuccessful call resulting in a 404 status code
- Unsuccessful call resulting in a 500 status code

Successful call–200 status code

```
curl -X GET http://{host}:{port}/innreach/v2/local/ploc8/bib/1 -H
'Accept: application/json'
   -H 'Authorization: Bearer 741619d4e1b2156b98e084a08b5a00d9'
   -H 'X-From-Code: ploc2'
   -H 'X-To-Code: pcent'
```

Sample Success Response:

```
{
   "status":"ok",
   "reason":"success",
   "errors":[],
}
```

Unsuccessful call–404 status code

The following call resulted in a "Record not found" error.

```
curl -X GET http://{host}:{port}/innreach/v2/local/ploc8/bib/23345 -H
'Accept: application/json'
   -H 'Authorization: Bearer 741619d4e1b2156b98e084a08b5a00d9'
   -H 'X-From-Code: ploc2'
   -H 'X-To-Code: pcent'
```

Sample Failure Response:

```
{
   "status":"failed",
   "reason":"Record not found",
   "errors":[],
}
```

Unsuccessful call–500 status code

The following call resulted in an internal error, "Failed to load central record".

```
curl -X GET http://{host}:{port}/innreach/v2/local/ploc8/bib/1 -H
```

© 2020 Innovative Interfaces, Inc.

```
'Accept: application/json'
   -H 'Authorization: Bearer 741619d4e1b2156b98e084a08b5a00d9'
   -H 'X-From-Code: ploc2'
   -H 'X-To-Code: pcent'
```

Sample Failure Response:

```
{
   "status":"failed",
   "reason":"Server internal error",
   "errors": {
      "type": "error",
      "name": "Internal error",
      "reason": "Failed to load central record"
   }
}
```

Look Up Bib by bibId and itemId

This method allows a client to look up a bibliographic record by its bibID and itemId.

The central returns the following status codes:

- 200 - The item was found, and the corresponding central record has been loaded.
- 404 - The item was not found by either the bibId or itemId.
- 500 - The item was found but belongs to a different bibliographic record than the one specified by the bibId.
- 500 - An error exists in the milproto format, or a system error occurred.

GET /innreach/v2/local/<localCode>/bib/<bibId>/item/<itemId>

URL parameters

Property	Description
localCode	Unique code that represents the local server.
bibId	The ID of the bibliographic record.
itemId	The ID of the item.

Response content

Property	Type	Required
status	String	Yes

© 2020 Innovative Interfaces, Inc.

Property	Type	Required
reason	String	Yes
errors	[]	Yes

Sample cURL calls

Three sample cURL calls are provided with different responses.

- Successful call resulting in a 200 status code
- Unsuccessful call resulting in a 404 status code
- Unsuccessful call resulting in a 500 status code

Successful call–200 status code

```
curl -X GET http://{host}:
{port}/innreach/v2/local/ploc8/bib/1/item/2475459 -H 'Accept:
application/json'
   -H 'Authorization: Bearer 741619d4e1b2156b98e084a08b5a00d9'
   -H 'X-From-Code: ploc2'
   -H 'X-To-Code: pcent'
```

Sample Success Response:

```
{
   "status":"ok",
   "reason":"success",
   "errors":[],
}
```

Unsuccessful call–404 status code

The following call resulted in a "Record not found" error.

```
curl -X GET http://{host}:
{port}/innreach/v2/local/ploc8/bib/23345/item/2475459 -H 'Accept:
application/json'
   -H 'Authorization: Bearer 741619d4e1b2156b98e084a08b5a00d9'
   -H 'X-From-Code: ploc2'
   -H 'X-To-Code: pcent'
```

Sample Failure Response:

```
{
   "status":"failed",
   "reason":"Record not found",
   "errors":[],
}
```

© 2020 Innovative Interfaces, Inc.

Unsuccessful call–500 status code

The following call resulted in an internal error, "Failed to load central record".

```
curl -X GET http://{host}:
{port}/innreach/v2/local/ploc8/bib/1/item/2475459 -H 'Accept:
application/json'
   -H 'Authorization: Bearer 741619d4e1b2156b98e084a08b5a00d9'
   -H 'X-From-Code: ploc2'
   -H 'X-To-Code: pcent'
```

Sample Failure Response:

```
{
   "status":"failed",
   "reason":"Server internal error",
   "errors": {
      "type": "error",
      "name": "Internal error",
      "reason": "Failed to load central record"
   }
}
```

Update Bib Status

Updates the number of linked bib items and the quantity of title level hold count.

POST /innreach/v2/contribution/bibstatus/<bibId>

URL parameters

Property	Description
bibId	ID of the bibliographic record.

Request parameters

Property	Type	Required	Description
titleHoldCount	Integer	Yes	Number of title level holds.
itemCount	Integer	Yes	Number of items linked to this bib.

© 2020 Innovative Interfaces, Inc.

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/contribution/bibstatus/
{bibId} -H 'Accept: application/json'
   -H 'Authorization: Bearer 6e1b45d436cac9114f1625f38e463e5a'
   -H 'Content-Type: application/json'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{"titleHoldCount":99, "itemCount":999}'
```

Update Item Status

Updates the item status, hold count, and time. This transaction can be used to update any combination of the message element fields, even when other fields haven't changed.

POST /innreach/v2/contribution/itemstatus/<itemId>

URL parameters

Property	Description
ItemId	ID of the item being updated.

Request parameters

Property	Type	Required	Description
itemCircStatus	String	Yes	Available/Not Available/On Loan/Non-Lendable
holdCount	Integer	No	Item hold count, maximum 99
dueDateTime	Integer	No	Epoch UNIX time stamp, for example: 1544466568

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/contribution/itemstatus/
{itemId} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{
         "itemCircStatus":"On Loan",
         "holdCount":50,
         "dueDateTime":955325450
      }'
```

© 2020 Innovative Interfaces, Inc.

Contribution error messages

The central sends messages to and receives messages from the local site. Messages contain one or more of the following error codes and descriptions.

Example: `statmsg=115:<bibId>:Required fields missing from MARC record`

Error Code	Description
112	Unsupported MARC format
113	Required element missing from transaction
114	Required data element empty
115	Required fields missing from MARC record
116	Connections to this central server temporarily disabled
117	[Terminate session]
118	Local server must be added to central branch file. Connection will be closed.
119	Critical system error. Notify administrator.
120	Invalid site code
121	Invalid MARC record recid
122	Only <N> connections allowed from this server
123	Record ID contains invalid characters
124	No location provided for item
126	Item circ status element missing or invalid
127	Item type must be a number between zero and 255
128	Item contribution transaction contains invalid agency code
129	RDA format of record not supported by this central
130	Specified record type not supported for record contribution
131	Missing or invalid titleHoldCount element
132	Missing or invalid itemCount element
133	Record error on central server; unable to process
135	No item record found with specified recid
137	Item type not defined on the central server.
138	Bib not allowed.
170	No bib record found with specified recid

© 2020 Innovative Interfaces, Inc.

Error Code	Description
171	Maximum allowed batch count exceeded.
172	One or more items in this batch are not linked to the specified local bib.
173	Central bibliographic record has reached the maximum allowed size.

© 2020 Innovative Interfaces, Inc.

Circulation

This section describes the methods used to circulate material between the central server and local sites.

Note: Code samples do not provide header information. Indentation has been applied for readability.

Content discussed in this section includes:

- "Circulation" (this page)
  - "Circulation overview" on page 42
  - "Initiating transactions" on page 1
  - "D2IR call sequence use case scenarios" on page 44
  - "Circulation transaction and states" on page 46
  - "Implementing endpoints" on page 51
  - "Terminating transactions" on page 52
  - "Outgoing circulation message failures" on page 52
  - "APIs implemented by Innovative" on page 53
    - "Borrower Renew Item" on page 53
    - "Cancel Item Hold" on page 54
    - "Create Local Hold" on page 69
    - "Create Patron Hold" on page 71
    - "Final Item Check-In" on page 54
    - "Item In Transit" on page 55
    - "Item Shipped" on page 56
    - "Local Checkout" on page 57
    - "Owner Renew Item" on page 58
    - "Owning Site Cancels Request" on page 58
    - "Patron Claims Returned" on page 59
    - "Report Item Received" on page 61
    - "Report Unshipped Item Received" on page 62
    - "Recall Item" on page 60
    - "Return Uncirculated Item" on page 63
    - "Transfer Request" on page 64
  - "APIs implemented by third-party ILS" on page 65
    - "Borrower Renew Item" on page 65
    - "Cancel Item Hold" on page 66
    - "Cancel Request" on page 67
    - "Create Item Hold" on page 67

© 2020 Innovative Interfaces, Inc.

- "Final Item Check-In" on page 72
- " Get Bib Record" on page 83
- "Item In Transit" on page 73
- "Item Shipped" on page 74
- "Owner Renew Item" on page 75
- "Patron Claims Returned" on page 76
- "Recall Item" on page 77
- "Report Item Received" on page 78
- "Report Unshipped Item Received" on page 79
- "Return Uncirculated Item" on page 80
- "Transfer Request" on page 81
- "Verify Patron" on page 82

Circulation overview

Circulation APIs (API endpoints) have different requirements than APIs under "Record contribution" on page 27. Clients using circulation API endpoints must implement these APIs on their local server. Doing so provides real-time status updates within the INN-Reach central consortium.

The INN-Reach central serves as an intermediary between local clients and affiliated delegates. D2IR ILSs pass information updates and status states between locals and delegates through circulation calls to API endpoints. API calls that originate from INN-Reach central are defined in "APIs implemented by third-party ILS" on page 65.

As illustrated in the diagram below, a patron verification request is made to the borrowing library when an item is requested. A successful verification means that the patron is an actual patron at the library, and the information they provided was sufficient to identify them as such. The item request transaction is then initiated, if a request can be placed to the INN-Reach central.

Initiating transactions

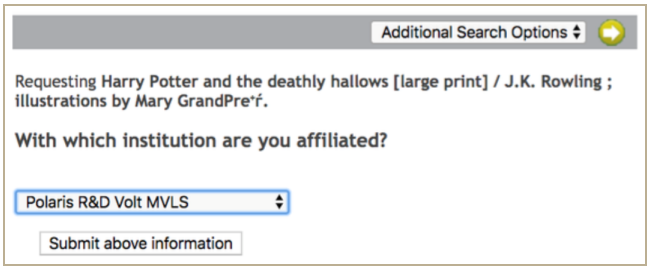
This section describes the transactions that occur when a patron initiates a borrow request for an item from the INN-Reach Central search service.

> **Note:** The Verify Patron API must be implemented on the local server.

To request an item, the patron does the following:

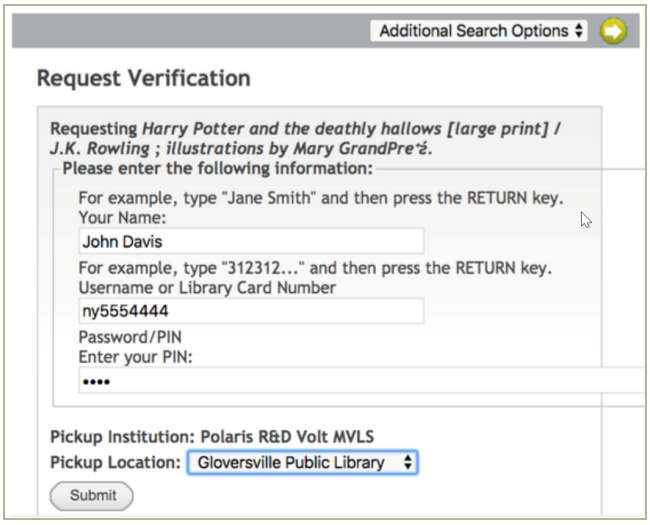
1. Searches the Central catalog to locate an item with a status of Available. The item is requested.
2. When the item is requested, a page displays asking which institution is affiliated with the patron. The patron selects their affiliated institution from a drop-down menu, shown in the following image.

© 2020 Innovative Interfaces, Inc.



3. When the institution is chosen and the **Submit above information** button is selected, the patron is redirected to a Request Verification page that accepts the patron's username and password/PIN

   Note: The Password/PIN may not be needed. It is configurable and subject to the requirements of each local server.



4. The credentials are entered for patron verification.
5. From the same page, the patron chooses a pickup location from the drop-down menu, and then selects **Submit**.

When the patron clicks the Submit button, the following activity initiates the circulation request.

- The Verify Patron API sends a call from the Central to the borrowing site with data collected in the Request Verification window. Request parameters are gathered, such as patronAgency, visiblePatronId, passcode, and patronName. With patron parameters gathered by the search UI, the central can call the local and determine whether the patron is able to request the item.
- If data is not successfully received, an error message is sent to the patron.
- The central initiates a request transaction by making the CreateItemHold API call from the central to the owning site and the CreatePatronHold API call from the central to the borrowing site.

© 2020 Innovative Interfaces, Inc.

Successful verification of the patron does not imply that the patron can place a central hold. A successful verification means that the patron is an actual library patron, and the provided information was sufficient to identify them as such. When successfully verified, even if the patron is not locally blocked (for example, the requestAllowed parameter in the Verify Patron response is true), the patron may still be unable to place a central request. It does not mean that an Item Hold and Patron Hold request will ensue.

For more information, see the D2IR call sequence use case scenarios. This section contains call-sequence diagrams that visually shows how API calls flow between the borrowing site, the Central, and the owning site.

D2IR call sequence use case scenarios

The following use case scenarios describe a Create Item Hold and Create Patron Hold transaction sequence. These examples show the importance of each local site implementing the necessary endpoints to enable circulation transactions to occur. They also express the importance of maintaining a trackingId for subsequent API calls related to these transactions.

Sequence diagram: successful item request

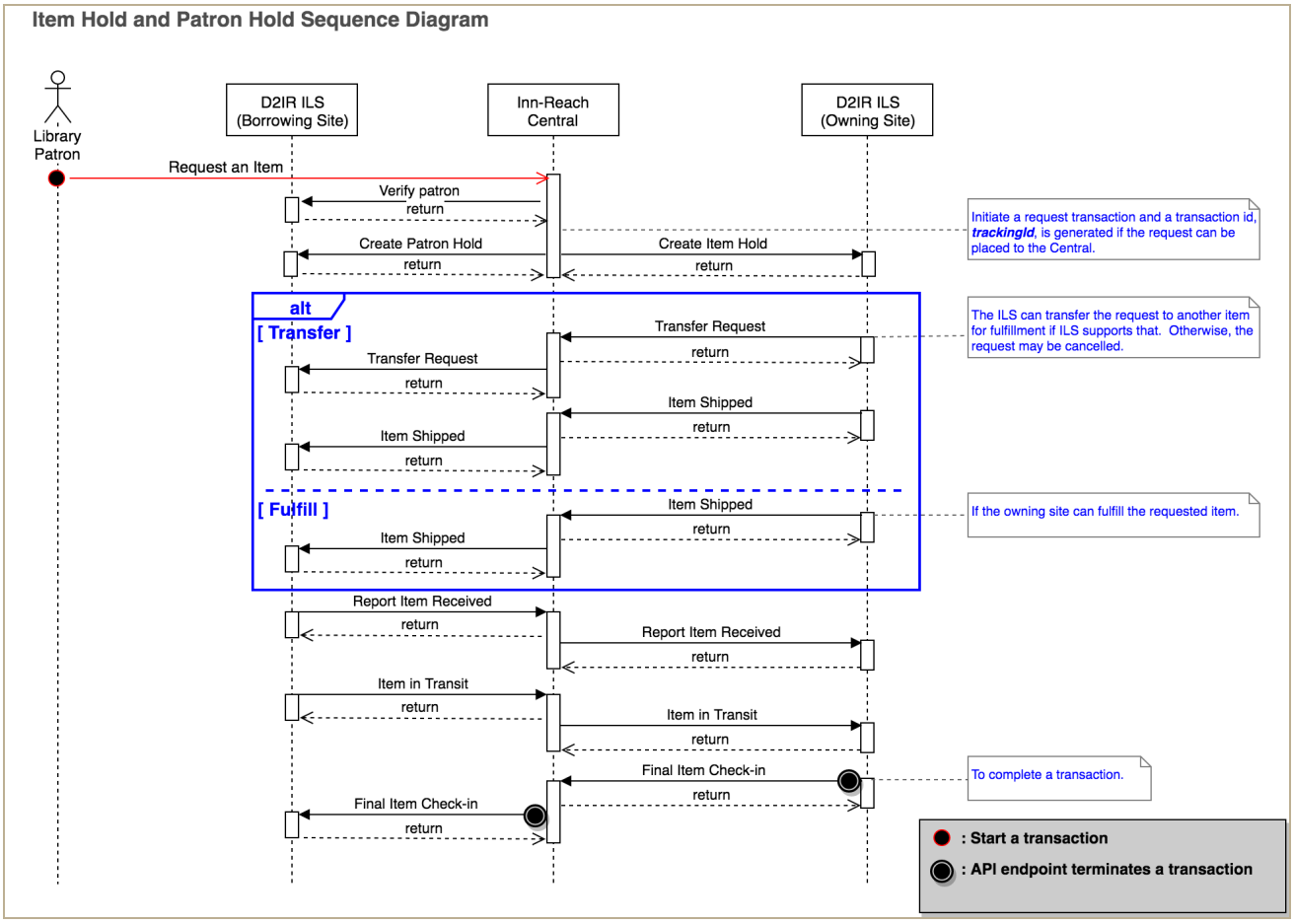
The following example describes the messages generated by a successful patron request, which creates both an Item Hold and Create Patron Hold transaction. As shown in the following diagram, the central server communicates with the ILS owning site and borrowing site.

In this example, a transaction is initiated by a library patron who requests an item. When the patron requests the item, INN-Reach central initiates a call to the Create Item Hold and Create Patron Hold endpoints. A transaction ID (trackingId) is generated, which is used to track the transaction, and both the tracking ID and central code are sent to the local site. The central code identifies the central server that initiated the transaction. The local site stores only the tracking ID. If a site participates in multiple consortia, then the central code will need to be stored.

If the ILS locates a requestable item, holds are generated at the borrowing and lending sites, and the owning site then sends an API call to Item Shipped to inform the central server that the requested physical item is being transported to the patron's library. Central then informs the borrowing site that the item is being shipped. When the borrowing site receives the item, a call is made to Report Item Received, and the central informs the owning site. When the item is returned, a call is made to Item in Transit, and central informs the owning site. When the item is received at the owning site, Final Item Check-in is called, and the borrowing site is informed that the item has been received at its originating location.

At any time before the item ships, the hold request can be canceled by the patron or library staff. If the transaction is never canceled, it is completed by a Final Item Check-In call from the owning library to the borrowing library where the patron’s request was fulfilled.

© 2020 Innovative Interfaces, Inc.



Sequence diagram: canceled item request

The following example describes a canceled Create Item Hold and Create Patron Hold transaction. This is initiated by the owning library or at the borrowing library by either a patron or the library staff.

There are two types of cancellation requests:

- A request initiated by the patron/staff at the borrowing library
- A request initialed by the lending library.

When a request is canceled by the patron/staff, the central server communicates with the owning library to indicate that the patron no longer wants the item. When the request is canceled by the lending library, the central communicates with the borrowing site informing it that either the central item has been found or that no such item could be located and the patron's request will be canceled. When the request is canceled by the patron/staff, the Cancel Item Hold transaction is sent and the hold is cancelled. This is a terminating transaction.

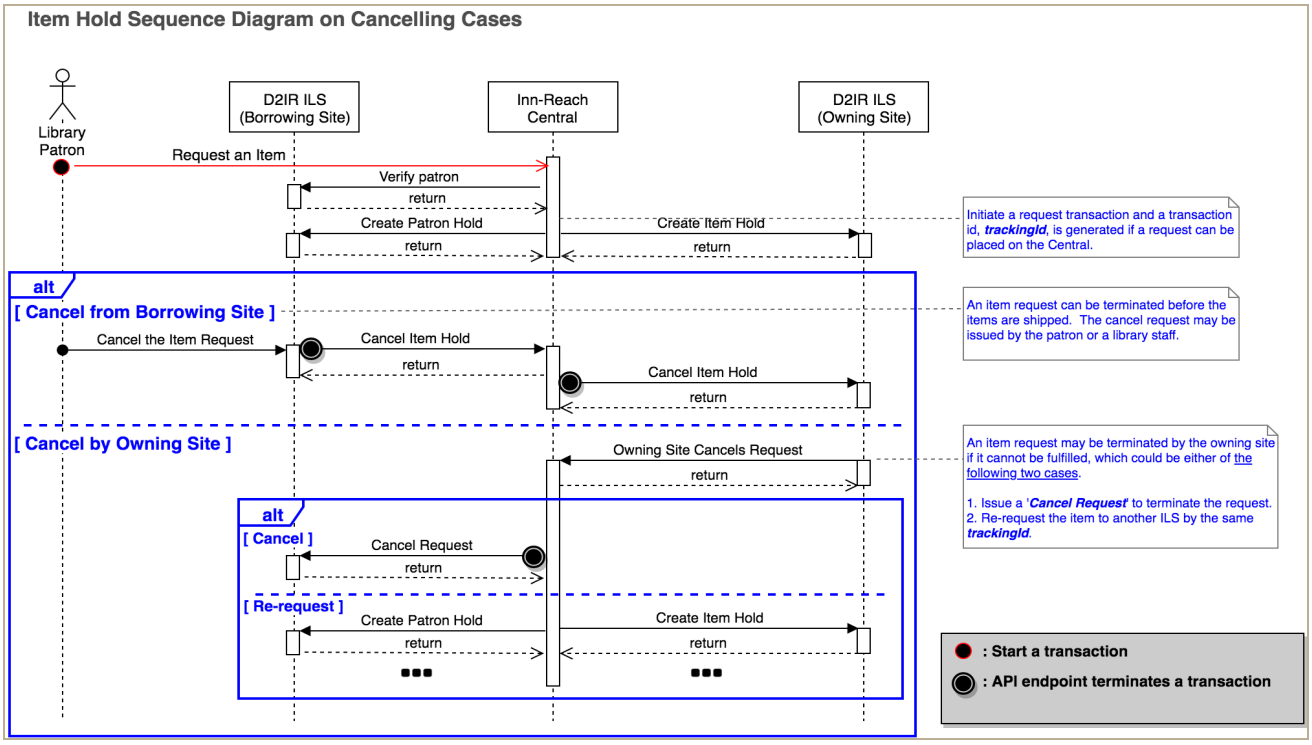
For a borrowing site-initiated cancellation, the patron/staff at the borrowing library cancels the request before the item is shipped from the owning library. A Cancel Item Hold transaction is sent from the borrowing site to central and passed to the owning site. This is a terminating transaction. No further exchanges between the borrowing, central, and owning sites will occur. The owning site cancels the request and the central tries to locate

© 2020 Innovative Interfaces, Inc.

an item from a different library to fill the patron's request. If the item is not located, a Cancel Request is sent to the borrowing site. This is a terminating transaction, and the patron's request is canceled.

If a substitute item is found to fill the patron's request, an new Create Patron Hold transaction is sent from central to the borrowing library. The same trackingId is used as that of the original request, and a new Create Item Hold transaction is sent to the new owning site.

Owning Site Cancel transactions may occur multiple times for the same request, and the above re-request sequence can be repeated until all possible supplying sites are exhausted or an item is shipped to fill the hold. Each successive Create Patron Hold and Create Item Hold transaction uses the same trackingId as that in the transactions created when the patron made the original request.



Circulation transaction and states

This section describes circulation transactions, state changes, and their relationship with different call scenarios. Error conditions are also discussed with examples.

A circulation transaction starts from a borrowing patron requesting an item and ends when the item is returned to the owning site. A circulation transaction includes several calls to circulation endpoints, and its state is changed with each successful API endpoint call. For example, a circulation transaction is changed to an ITEM_SHIPPED state after the Item Shipped endpoint is successfully called by an owning site.

The followings are defined circulation request types in Inn-Reach.

© 2020 Innovative Interfaces, Inc.

ITEM_HOLD	LOCAL_HOLD
PATRON_HOLD	BORROWER_RENEW
BORROWING_SITE_CANCEL	ITEM_IN_TRANSIT
PATRON_HOLD	BORROWER_RENEW
RECEIVE_UNANNOUNCED	RETURN_UNCIRCULATED
CLAIMS_RETURNED	ITEM_RECEIVED
ITEM_SHIPPED	LOCAL_CHECKOUT
CANCEL_REQUEST	FINAL_CHECKIN
RECALL	TRANSFER

Out-of-sequence circulation request

An out-of-sequence circulation request occurs when an unallowed circulation endpoint call is made against a current circulation transaction state. An HTTP response header, X-IR-Allowed-Circulation, is available for clients to check for allowed circulation request types.

For example, a call to the *Item in Transit* endpoint by the borrowing site against a circulation transaction in the ITEM_SHIPPED state will be rejected. You'll see an `HTTP 400 Bad Request` and [ITEM RECEIVED] for the X-IR-Allowed-Circulation in the response, as shown below.

```
curl --location --request POST
'http://centapi.iii.com:8083/innreach/v2/circ/intransit/100001/pcent' \
> --header 'Content-Type: application/json' \
> --header 'Accept: application/json' \
> --header 'X-From-Code: mclc2' \
> --header 'X-To-Code: pcent' \
> --header 'Authorization: Bearer 4d0480bbb50812acd6b33a48a3128fc9' --
verbose
* Trying 10.1.125.66...
* TCP_NODELAY set
* Connected to centapi.iii.com (10.1.125.66) port 8083 (#0)
> POST /innreach/v2/circ/intransit/100001/pcent HTTP/1.1
> Host: centapi.iii.com:8083
> User-Agent: curl/7.54.0
> Content-Type: application/json
> Accept: application/json
> X-From-Code: mclc2
> X-To-Code: pcent
> Authorization: Bearer 4d0480bbb50812acd6b33a48a3128fc9
>
< HTTP/1.1 400 Bad Request

< Server: Apache-Coyote/1.1
< Set-Cookie: rememberMe=deleteMe; Path=/innreach; Max-Age=0;
Expires=Thu, 16-Jan-2020 23:52:31 GMT
< X-IR-Allowed-Circulation: [ITEM RECEIVED]
< Content-Type: text/json;charset=utf-8
< Transfer-Encoding: chunked
< Date: Fri, 17 Jan 2020 23:52:31 GMT
```

© 2020 Innovative Interfaces, Inc.

```
<
* Connection #0 to host centapi.iii.com left intact
{
   "status": "failed",
   "reason": "Circulation request not allowed",
   "errors": [
      {
         "type": "error",
         "name": "InternalError",
         "reason": "The ITEM IN TRANSIT API request is not allowed,
         please check X-IR-Allowed-Circulation response header"
      }
   ]
}
```

The following example shows what happens if the owning site calls the *Owner Renew Item* endpoint against a circulation transaction that is in the RETURN_UNCIRCULATED state. The API endpoint call is rejected with an `HTTP 400 Bad Request` and [FINAL_CHECKIN] for the X-IR-Allowed-Circulation response header, as shown in the following return.

```
curl --location --request POST
'http://centapi.iii.com:8083/innreach/v2/circ/ownerrenew/100001/pcent?du
eDateTime=1580150979' \
> --header 'Content-Type: application/json' \
> --header 'Accept: application/json' \
> --header 'X-From-Code: ploc8' \
> --header 'X-To-Code: pcent' \
> --header 'Authorization: Bearer 4fec08ccd110adb232e35337c036104a' --
verbose
* Trying 10.1.125.66...
* TCP_NODELAY set
* Connected to centapi.iii.com (10.1.125.66) port 8083 (#0)
> POST /innreach/v2/circ/ownerrenew/100001/pcent?dueDateTime=1580150979
HTTP/1.1
> Host: centapi.iii.com:8083
> User-Agent: curl/7.54.0
> Content-Type: application/json
> Accept: application/json
> X-From-Code: ploc8
> X-To-Code: pcent
> Authorization: Bearer 4fec08ccd110adb232e35337c036104a
>
< HTTP/1.1 400 Bad Request

< Server: Apache-Coyote/1.1
< Set-Cookie: rememberMe=deleteMe; Path=/innreach; Max-Age=0;
Expires=Sun, 26-Jan-2020 18:49:49 GMT
< X-IR-Allowed-Circulation: [FINAL CHECKIN]
< Content-Type: text/json;charset=utf-8
< Transfer-Encoding: chunked
< Date: Mon, 27 Jan 2020 18:49:49 GMT
```

© 2020 Innovative Interfaces, Inc.

```
<
* Connection #0 to host centapi.iii.com left intact
{
   "status": "failed",
   "reason": "Circulation request not allowed",
   "errors": [
      {
         "type": "error",
         "name": "InternalError",
         "reason": "The OWNER RENEW API request is not allowed, please
         check X-IR-Allowed-Circulation response header"
      }
   ]
}
```

If an empty X-IR-Allowed-Circulation response header is returned, this indicates that the circulation transaction is completed.

```
< HTTP/1.1 400 Bad Request
< Server: Apache-Coyote/1.1
< Set-Cookie: rememberMe=deleteMe; Path=/innreach; Max-Age=0;
Expires=Thu, 16-Jan-2020 23:58:44 GMT
< X-IR-Allowed-Circulation: []
< Content-Type: text/json;charset=utf-8
< Transfer-Encoding: chunked
< Date: Fri, 17 Jan 2020 23:58:44 GMT
<
* Connection #0 to host centapi.iii.com left intact
{
   "status": "failed",
   "reason": "Circulation request not allowed",
   "errors": [
      {
         "type": "error",
         "name": "InternalError",
         "reason": "The ITEM IN TRANSIT API request is not allowed since
         the transaction is completed"
      }
   ]
}
```

Wrong way circulation request

A wrong-way circulation request refers to a circulation endpoint call by an invalid caller, which may include one of the following conditions.

- Calling a borrowing circulation endpoint by the owning site of a circulation transaction.
- Calling an owning circulation endpoint by the borrowing site of a circulation transaction.

© 2020 Innovative Interfaces, Inc.

- Calling any circulation endpoint by a local that is neither the owning or borrowing site of a circulation transaction.

A wrong-way circulation request is rejected whether or not the circulation transaction state is completed.

```
curl --location --request POST
'http://centapi.iii.com:8083/innreach/v2/circ/claimsreturned/100001/pcen
t?claimsReturnedDate=1580152749' \
> --header 'Content-Type: application/json' \
> --header 'Accept: application/json' \
> --header 'X-From-Code: ploc2' \
> --header 'X-To-Code: pcent' \
> --header 'Authorization: Bearer 660a408a9f76881fe775591abb31f4bd' --
verbose
* Trying 10.1.125.66...
* TCP_NODELAY set
* Connected to centapi.iii.com (10.1.125.66) port 8083 (#0)
> POST
/innreach/v2/circ/intransit/100001/pcent?claimsReturnedDate=1580152749
HTTP/1.1
> Host: centapi.iii.com:8083
> User-Agent: curl/7.54.0
> Content-Type: application/json
> Accept: application/json
> X-From-Code: ploc2
> X-To-Code: pcent
> Authorization: Bearer 660a408a9f76881fe775591abb31f4bd
>
< HTTP/1.1 400 Bad Request

< Server: Apache-Coyote/1.1
< Set-Cookie: rememberMe=deleteMe; Path=/innreach; Max-Age=0;
Expires=Sun, 26-Jan-2020 19:19:22 GMT
< Content-Type: text/json;charset=utf-8
< Transfer-Encoding: chunked
< Date: Mon, 27 Jan 2020 19:19:22 GMT
< Connection: close
<
* Connection #0 to host centapi.iii.com left intact
{
   "status": "failed",
   "reason": "Invalid request",
   "errors": [
      {
         "type": "HeaderError",
         "name": "X-From-Code",
         "rejectValue": "ploc2",
         "reason": "Invalid Header Value"
      }
   ]
}
```

© 2020 Innovative Interfaces, Inc.

Implementing endpoints

Important: In order to receive circulation messages on the local server, you must implement the endpoints described for each message sent from the central server to the local server as listed in this section.

Transactions initiated from the INN-Reach central require the local server to store the tracking ID (trackingId). The central code (centralCode) should be saved. These properties are referenced with the initiating transaction and all future API requests. They are also used to record the transaction status at the local server.

The transaction ID must be maintained throughout the life of a patron/item association. All transactions referring to the patron/item association must use the same tracking ID. This applies from when a patron places a hold on an item to when the terminating transaction is initiated. If a patron checks the same item out again, a new tracking ID is issued.

Borrowing site

Third-party borrowing sites must implement the following APIs:

- **Cancel Request** PUT /innreach/v2/circ/cancelrequest/<trackingId>/<centralCode>
- **Final Item Check-In** PUT /innreach/v2/circ/finalcheckin/<trackingId>/<centralCode>
- **Item Shipped** PUT /innreach/v2/circ/itemshipped/<trackingId>/<centralCode>
- **Create Local Hold** PUT /innreach/v2/circ/localhold/<trackingId>/<centralCode>
- **Create Patron Hold** POST /innreach/v2/circ/patronhold
- **Owner Renew Item** PUT /innreach/v2/circ/ownerrenew/<trackingId>/<centralCode>
- **Recall Item** PUT /innreach/v2/circ/recall/<trackingId>/<centralCode>
- **Transfer Request** PUT /innreach/v2/circ/transferrequest/<trackingId>/<centralCode>

Owning site

Third-party owning sites must implement the following APIs:

- **Borrower Renew Item** PUT /innreach/v2/circ/borrowerrenew/<trackingId>/<centralCode>
- **Create Item Hold** PUT /innreach/v2/circ/cancelitemhold/<trackingId>/<centralCode>
- **Create Local Hold** PUT /innreach/v2/circ/localhold/<trackingId>/<centralCode>
- **Patron Claims Returned** PUT /innreach/v2/circ/claimsreturned/<trackingId>/<centralCode>
- **Item In Transit** PUT /innreach/v2/circ/intransit/<trackingId>/<centralCode>
- **Report Item Received** PUT /innreach/v2/circ/itemreceived/<trackingId>/<centralCode>
- **Report Unshipped Item Received** PUT /innreach/v2/circ/receiveunshipped/<trackingId>/<centralCode>
- **Return Uncirculated Item** PUT /innreach/v2/circ/returnuncirculated/<trackingId>/<centralCode>

© 2020 Innovative Interfaces, Inc.

Terminating transactions

When a terminating transaction is received, the patron/item messages associated with a tracking ID will cease, and any data retained will be stored for statistical purposes only.

The terminating transactions are:

- **Final Item Check-In** PUT /innreach/v2/circ/finalcheckin/<trackingId>/<centralCode>
- **Cancel Item Hold** PUT /innreach/v2/circ/cancelitemhold/<trackingId>/<centralCode>
- **Cancel Request** PUT /innreach/v2/circ/cancelrequest/<trackingId>/<centralCode>
- **Local Checkout** PUT /innreach/v2/circ/localcheckout/<trackingId>/<centralCode>
- **Patron Claims Returned** PUT /innreach/v2/circ/claimsreturned/<trackingId>/<centralCode>

Outgoing circulation message failures

Outgoing circulation (circ) messages are sent to non-Innovative local servers even if the local server is unable to receive the message due to a connection or authentication error. If the central server fails to send an outgoing circ message to a non-innovative local server, the message is requeued to a central server queue until the transmission is successful.

© 2020 Innovative Interfaces, Inc.

APIs implemented by Innovative

The APIs in this section communicate from the local server to the central server.

Borrower Renew Item

Sends a transaction from the borrowing local site to the central server.

POST /innreach/v2/circ/borrowerrenew/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

Property	Type	Required	Description
dueDateTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/borrowerrenew/{track-
ingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}' -H 'X-To-Code: {centralcode}'
   -d '{"dueDateTime": 8362365423}'
```

© 2020 Innovative Interfaces, Inc.

Cancel Item Hold

Sends a terminating transaction from the borrowing local site to the central server.

POST /innreach/v2/circ/cancelitemhold/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

None

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/cancelitemhold/{track-
ingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

Final Item Check-In

Sends a transaction from the owning local site to the central server when a borrowed item from an off-site patron is checked back into the owning library. This transaction terminates the loan to the patron.

POST /innreach/v2/circ/finalcheckin/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

None

© 2020 Innovative Interfaces, Inc.

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/finalcheckin/{track-
ingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

Item In Transit

Sends a transaction from the borrowing local site to the central when an item is being shipped back to the owning site.

POST /innreach/v2/circ/intransit/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lowercase alphanumeric, 3-5 characters.

Request parameters

None

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/intransit/{track-
ingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

© 2020 Innovative Interfaces, Inc.

Item Shipped

Sends a transaction from the owning local site to the central server to indicate that the requested physical item is being transported to the patron's library. This is the first transaction where the item barcode is sent to the receiving site. When the item is received at the borrowing site, the patron's library can take appropriate action for their circulation system.

POST /innreach/v2/circ/itemshipped/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
callNumber	String	No	
itemBarcode	String	Conditional	If present in item or affiliated bib

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/itemshipped/{track-
ingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{"itemBarcode": "ibare 34"}'
```

© 2020 Innovative Interfaces, Inc.

Local Checkout

Sends a terminating transaction from the local owning site to the central server when a local item with a hold created by a Create Local Hold transaction is checked out to the patron. This transaction must use the trackingId from the original Create Local Hold transaction.

POST /innreach/v2/circ/localcheckout/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lowercase alphanumeric, 3-5 characters.

Request parameters

Property	Type	Required	Description
localBibId	String	Yes	
message	String	No	Optional checkout/item note. Maximum 256 bytes.
itemLocation	String	No	Location of the item on the local library shelf. Maximum 80 char.
itemBarcode	String	Conditional	If present in item or affiliated bib

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/localcheckout/{track-
ingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{
      "localBibId":"12343435",
      "message":"Message from the Wiser",
      "itemLocation":"9loc2",
      "itemBarcode":"bar32243"
   }'
```

© 2020 Innovative Interfaces, Inc.

Owner Renew Item

Sends a transaction from the owning local site to the central server.

POST /innreach/v2/circ/ownerrenew/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

Property	Type	Required	Description
dueDateTime	Integer	Yes	Epoch UNIX time stamp, for example:1544466568

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/ownerrenew/{track-
ingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{"localBibId":"12343435"}'
```

Owning Site Cancels Request

Sends a transaction from the local owning site to the central server.

POST /innreach/v2/circ/owningsitecancel/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

© 2020 Innovative Interfaces, Inc.

Request parameters

Property	Type	Required	Description
localBibId	String	Yes	Lowercase alphanumeric, maximum 32 characters
reason	String	No	
reasonCode	Integer	Yes	"7" This value will always be 7.
patronName	String	Yes	

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/owningsitecancel/
{trackingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{
      "localBibId":"12343435",
      "reason":"Item lost",
      "reasonCode":7,
      "patronName":"John Smith"
   }'
```

Patron Claims Returned

Sends a terminating transaction from the borrowing local site to the central server.

POST /innreach/v2/circ/claimsreturned/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

Property	Type	Required	Description
claimsReturnedDate	Integer	Yes	Epoch UNIX time stamp, for example:1544466568

© 2020 Innovative Interfaces, Inc.

Note: A value of `-1` means the claims returned date is unknown.

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/claimsreturned/{track-
ingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{"claimsReturnedDate": 8362365423}'
```

Recall Item

Sends a transaction from the owning site to the central.

POST /innreach/v2/circ/recall/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

Property	Description
dueDateTime	A new item due date for the recall.

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[]	Yes

© 2020 Innovative Interfaces, Inc.

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/recall/{trackingId}/
{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{
      "dueDateTime":1544466568
   }'
```

Report Item Received

Sends a transaction from the borrowing local site to the central.

POST /innreach/v2/circ/itemreceived/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

None

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[]	Yes

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/itemreceived/{track-
ingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
```

© 2020 Innovative Interfaces, Inc.

```
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

Report Unshipped Item Received

Sends a transaction from the borrowing site to the central when an INN-Reach Item is received at the borrowing library, but the borrowing library never received an Item Shipped message for this item.

POST /innreach/v2/circ/receiveunshipped/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lowercase alphanumeric, 3-5 characters.

Request parameters

None

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/receiveunshipped/
{trackingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

© 2020 Innovative Interfaces, Inc.

Return Uncirculated Item

Sends a transaction from central to the third-party local server. This API must be implemented on the local server. This transaction is used to inform the owning site that an item is being returned.

POST /innreach/v2/circ/returnuncirculated/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

None

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[]	Yes

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/returnuncirculated/
{trackingId}/{centralCode} -H 'Accept: application/json' -H 'Content-
Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

© 2020 Innovative Interfaces, Inc.

Transfer Request

Sends a transaction from the owning local site to the central server indicating that the request will be fulfilled using an alternate item than was originally requested from the same agency.

POST /innreach/v2/circ/transferrequest/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

Property	Type	Required	Description
newItemId	String	Yes	Must be from the same agency as the original item.

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/transferrequest/
{trackingId}/{centralCode} -H 'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
   -d '{"newItemId": "newid1234"}'
```

© 2020 Innovative Interfaces, Inc.

APIs implemented by third-party ILS

This section describes API methods that must be implemented by a third-party site.

Borrower Renew Item

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/borrowerrenew/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
dueDateTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Cancel Item Hold

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/cancelitemhold/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Cancel Request

Sends a cancel request terminating transaction from the central server to the borrowing site. This cancellation message may result from an `owningsitecancel` transaction from the owning site when the central system fails to find an alternate item to fill the patron's request The `cancelrequest` message results if the `owningsitecancel` message fails to find an alternate item for the patron when the original item is not available.

PUT /innreach/v2/circ/cancelrequest/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
reason	String	No	
reasonCode	Integer	Yes	"7" This value will always be 7.

Create Item Hold

Originates an item hold at central that is sent to the owning site.

POST /innreach/v2/circ/itemhold/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server. 3-5 character lowercase alphanumeric

© 2020 Innovative Interfaces, Inc.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
pickupLocation	String	Yes	A colon-delimited list of three or four 512-byte (maximum) strings, with the last string being optional. Example: **<Pickup Loc Code>:<Display Name>:<Print Name>:<Delivery Stop>**
patronId	String	Yes	lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	lowercase alphanumeric string, maximum 32 characters
needBefore	Integer	No	Epoch UNIX time stamp, for example: 1544466568
centralPatronType	Integer	Yes	0-255
patronName	String	Yes	

Sample cURL call

```
curl -X POST http://{host}:{port}/innreach/v2/circ/itemhold -H 'Accept:
application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}' -H 'X-To-Code: {centralcode}'
   -d '{
      "trackingId": 12345,
      "centralCode": "pcent",
      "transactionTime": 1544466568,
      "pickupLocation": "1234 Library Rd. Paradise City, CA",
      "patronId":"cacclib-p12345",
      "patronAgencyCode":"jcg01",
      "itemAgencyCode":"ydg01",
      "itemId":"cacclib-i12345",
      "needBefore": 1745467531,
      "centralPatronType":200,
      "patronName":"test patron"
   }'
```

© 2020 Innovative Interfaces, Inc.

Create Local Hold

Sends a transaction from the central server to the local site when a patron from that site requests an item. Not all consortia allow patrons to initiate local requests. If allowed, the item request must originate from the same local server, but the item does not have to belong to the same agency as the patron. A local server may host items from multiple agencies.

PUT /innreach/v2/circ/localhold/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
pickupLocation	String	Yes	A colon-delimited list of three or four 512-byte (maximum) strings, with the last string being optional. Example: `<Pickup Loc Code>:<Display Name>:<Print Name>:<Delivery Stop>`
patronId	String	Yes	lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
patronHomeLibrary	String	No	
patronPhone	String	No	
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	lowercase alphanumeric string, maximum 32 characters
centralItemType	Integer	Yes	0-255
title	String	Conditional	256 bytes maximum Send if the item has a title.
author	String	No	256 bytes maximum
callNumber	String	No	256 bytes maximum
needBefore	Integer	No	Epoch UNIX time stamp, for example: 1544466568
centralPatronType	Integer	Yes	0-255
patronName	String	Yes	

© 2020 Innovative Interfaces, Inc.

Response

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[]	Yes

Sample cURL response

```
curl -X POST http://{host}:{port}/innreach/v1/localhold -H 'Author-
ization: Bearer 708357a59050533832437628d4087b80'
   -H 'Content-Type: application/json'
   -H 'Accept: application/json' -X POST
   -H 'X-From-Code: pcent'
   -H 'X-To-Code: ploc8'
   -d '
      {
         "centralCode":"pcent",
         "trackingId":"2134565",
         "itemAgencyCode":"da34e",
         "itemId":"214324",
         "centralItemType":"34",
         "needBefore":"",
         "patronAgencyCode":"deg12",
         "patronId":"24354",
         "patronHomeLibrary":"Main",
         "patronName":"John Doe",
         "pickupLocation":"Main, Second",
         "transactionTime": "123234134",
         "centralPatronType":"34",
         "reRequest":"false"
      }'
```

© 2020 Innovative Interfaces, Inc.

Create Patron Hold

Originates a patron hold at central that is sent to the borrowing site.

POST /innreach/v2/circ/patronhold/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
pickupLocation	String	Yes	A colon-delimited list of three or four 512-byte (maximum) strings, with the last string being optional. Example: **<Pickup Loc Code>:<Display Name>:<Print Name>:<Delivery Stop>**
patronId	String	Yes	lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	lowercase alphanumeric string, maximum 32 characters
centralItemType	Integer	Yes	0-255
title	String	Conditional	256 bytes maximum Send if the item has a title.
author	String	No	256 bytes maximum
callNumber	String	No	256 bytes maximum
needBefore	Integer	No	Epoch UNIX time stamp, for example: 1544466568

Sample cURL response

```
curl -X POST http://{host}:{port}/innreach/v2/circ/patronhold -H
'Accept: application/json'
   -H 'Content-Type: application/json'
   -H 'Authorization: Bearer 2af564e268eae1375c44e8c25fc77348'
   -H 'X-From-Code: {localcode}'
   -H 'X-To-Code: {centralcode}'
```

© 2020 Innovative Interfaces, Inc.

```
-d '{
   "trackingId": 12345,
   "centralCode": "pcent",
   "transactionTime": 1544466568,
   "pickupLocation": "1234 Library Rd. Paradise City, CA",
   "patronId":"cacclib-p12345",
   "patronAgencyCode":"jcg01",
   "itemAgencyCode":"ydg01",
   "itemId":"cacclib-i12345",
   "centralItemType":80,
   "title":"Ants in my pant"
}'
```

Final Item Check-In

Sends a transaction from central to the third-party local server.

This message is sent when the physical item borrowed by the off-site patron has been checked back in to the owning library to terminate the loan. The message indicates that the patron/item interaction has concluded.

PUT /innreach/v2/circ/finalcheckin/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters

© 2020 Innovative Interfaces, Inc.

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

Item In Transit

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/intransit/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Item Shipped

Sends a transaction from central to the third-party local server. This API must be implemented on the local server.

This message indicates that the requested physical item is being transported to the patron's library. When received at the borrowing site, the patron's library can take appropriate action for their circulation system. This is the only transaction where the barcode of the item being shipped is provided.

`PUT` /innreach/v2/circ/itemshipped/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
centralItemType	Integer	Yes	0-255
itemBarcode	String	No	If present in item or affiliated bib
title	String	Conditional	256 bytes maximum Send if the item has a title.
author	String	No	256 bytes maximum
callNumber	String	No	256 bytes maximum
itemLocation	String	No	Description of item's home location. This should correspond to one of the locations provided by the "Upload Locations List" on page 24.
pickupLocation	String	Yes	A colon-delimited list of three or four 512-byte (maximum) strings, with the last string being optional. Example: `<Pickup Loc Code>:<Display Name>:<Print Name>:<Delivery Stop>`
needBefore	Integer	No	Epoch UNIX time stamp, for example: 1544466568

© 2020 Innovative Interfaces, Inc.

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

Owner Renew Item

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/ownerrenew/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
dueDateTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Patron Claims Returned

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/claimsreturned/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
claimsReturnedDate	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters

Note: A value of `-1` means the claims returned date is unknown.

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Recall Item

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/recall/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
dueDateTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Report Item Received

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/itemreceived/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
centralItemType	Integer	Yes	0-255
author	String	Conditional	If present
title	String	Conditional	256 bytes maximum Send if the item has a title.
itemBarcode	String	Conditional	If present in item or affiliated bib
callNumber	String	No	256 bytes maximum
centralPatronType	Integer	Yes	0-255

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Report Unshipped Item Received

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/receiveunshipped/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example:1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Return Uncirculated Item

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/returnuncirculated/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
author	String	Conditional	If present
title	String	Conditional	256 bytes maximum Send if the item has a title.

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Transfer Request

Sends a transaction from central to the third-party local server.

PUT /innreach/v2/circ/transferrequest/<trackingId>/<centralCode>

URL parameters

Property	Description
trackingId	ID used for transaction tracking.
centralCode	Unique code that identifies the central server.

Request parameters

Property	Type	Required	Description
transactionTime	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568
patronId	String	Yes	lowercase alphanumeric string, maximum 32 characters
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
itemAgencyCode	String	Yes	5 character lowercase alphanumeric
itemId	String	Yes	Lowercase alphanumeric string, maximum 32 characters
newItemId	String	Yes	Must be from the same agency as the original item

Response content

Property	Type	Required
status	String	Yes
reason	String	Yes
errors	[ ]	Yes

© 2020 Innovative Interfaces, Inc.

Verify Patron

Sends a message from the central to the local server. A patron must be found for successful verification.

POST /innreach/v2/circ/verifypatron

URL parameters

None

Request parameters

Property	Type	Required	Description
visiblePatronId	String	Yes	
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
passcode	String	*Conditional	PIN or password
patronName	String	Yes	Maximum 128 characters

* PIN or password required if the local site requires its patrons to supply PINs or passwords when authenticating. If the local site requires PINs or passwords, inform the central server administrator. The central server must be set up appropriately.

Response content

Property	Type	Required	Description
status	String	Yes	
reason	String	Yes	
errors	[ ]	Yes	
requestAllowed	Boolean	Yes	If the value is false, the patron cannot place a central request (they will be locally blocked).
patronInfo	(See below)	Optional	patronInfo is returned in the response if there is a matched patron and requestAllowed is True.

patronInfo

Property	Type	Required	Description
patronId	String	Yes	Permanent unique patron ID
patronAgencyCode	String	Yes	5 character lowercase alphanumeric
centralPatronType	Integer	Yes	0-255
patronExpireDate	Integer	Yes	Epoch UNIX time stamp, for example: 1544466568

© 2020 Innovative Interfaces, Inc.

Property	Type	Required	Description
localLoans	Integer	Yes	Number of local (within library) loans
nonLocalLoans	Integer	Yes	Number of non-local (INN-Reach, ILL, etc.) loans
patronName	String	Yes	Maximum 128 characters

Get Bib Record

This method requests that an already contributed record be returned to central.

POST /innreach/v2/getbibrecord/<bibId>/<centralCode>

URL parameters

Property	Description
bibId	ID of the bibliographic record. Lowercase alpha-numeric string, maximum 32 characters.
centralCode	Unique code that identifies the central server. Lower-case alphanumeric, 3-5 characters.

Request parameters

None

Response content

Property	Type	Required	Description
status	String	Yes	
reason	String	Yes	
errors	[ ]	Yes	
bibInfo	(see below)	Conditional	Returned only if bib record exists.

bibInfo

Property	Type	Required	Description
marc21BibFormat	String	Yes	Currently supports "ISO2709"
marc21BibData	String	Yes	Base 64 encoded MARC data (maximum 99999 bytes before base 64 encoding). See "Get Base64 Encoding Table" on page 32. • Only standard MARC fields accepted • Must include 008 and 245 fields

© 2020 Innovative Interfaces, Inc.

Property	Type	Required	Description
			• Omit 9XX fields • Strip data with local meaning only or that you do not want to expose
titleHoldCount	Integer	Yes	Number of title level holds (capped at 100,000)
itemCount	Integer	Yes	Number of items linked to this bib (capped at 500,000)
suppress	String	No	1 character n - no (default) y - yes g - display as owned by system, not by site

Get Status

Check the status of the API. this is a ping test to check that the API is running.

A successful call will return a 200 OK message.

GET /innreach/v2/status

URL parameters
None

Request parameters
None

© 2020 Innovative Interfaces, Inc.

Glossary

Agency

Collections of bibs, serials holdings, items, and patrons are defined as belonging to an organization (agency) on a Local Server, which may host multiple agencies. Each agency is identified as a separate organization on the INN-Reach Catalog. Because it is assumed that all agencies on a Local Server share the same namespace, record IDs must be unique within the Local Server. When patrons request items from the INN-Reach Catalog, they must identify the agency to which they belong.

Agency Code

Five-character code that represents an agency. The code must be unique among all agencies using INN-Reach, regardless of the resource sharing consortium in which the agency participates. Innovative helps INN-Reach customers determine that the agency code is unique.

INN-Reach Catalog

A combined database of the holdings from all the Local Servers that contribute to the INN-Reach Central Server. The INN-Reach Catalog comprises master bibliographic records and detailed holdings from the sites that own copies of those records. The INN-Reach Central Server receives records and updates from Local Servers in real time.

INN-Reach Central Server

Houses and provides Discovery for the INN-Reach Catalog. Also serves as the hub through which circulation requests and messages are received and sent to owning and borrowing sites. Uses INN-Reach software to perform a variety of tasks, including building and maintaining the INN-Reach Catalog; building and maintaining INN-Reach statistics; load balancing; and determination of lender.

INN-Reach Request

Request for an item on the INN-Reach Catalog.

INN-Reach System

A group of Local Servers and the INN-Reach Central Server to which they all contribute. Local Servers that contribute to more than one INN-Reach Central Server belong to more than one INN-Reach System.

Local Server

Stores records that are contributed to an INN-Reach Central Server. A Local Server is often the integrated library system for the libraries it represents.

Local Server Code

Five-character code that represents the Local Server. The code must be unique among all Local Servers using INN-Reach, regardless of the resource sharing consortium in which the Local Server participates. Innovative helps INN-Reach customers determine that the Local Server code is unique.

Non-Innovative Libraries

Libraries that do not use Innovative's integrated library systems for their OPAC or circulation modules.

Owning/Lending Site

The site to which the requested INN-Reach item belongs.

© 2020 Innovative Interfaces, Inc.

Patron/Borrowing Site

The site to which the requesting patron belongs.

Site

Generic term that represents an agency on a Local Server.

© 2020 Innovative Interfaces, Inc.