

SirsiDynix Symphony Web Services 3.1

# SDK

July 2011



© 2009-2011 SirsiDynix. All rights reserved.

The processes and all the routines contained herein are the proprietary properties and trade secrets of SirsiDynix. Except as provided by license agreement, this manual shall not be duplicated, used or disclosed for any purpose or reason, in whole or in part, without the express written consent of SirsiDynix. The information in this document is subject to change without notice and should not be construed as a commitment by SirsiDynix.

SirsiDynix grants the right of copying the enclosed material solely for the internal business use of the end user if (1) this document has been obtained by purchase or by license agreement in conjunction with SirsiDynix products, and (2) this copyright statement is included with each copy. All other copying or distribution is strictly prohibited.

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in sub-paragraph (b)(3) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Rights for non-DOD U.S. Government departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Sirsi®, WebCat®, Vizion®, WorkFlows®, Unicorn®, UnicornECOLE®, UnicornOASIS®, UnicornSTILAS®, Site Source®, DRA®, DRA Net®, Inlex®, MultiLIS®, Taos®, JurisLink™, Corinthian®, Dynix®, URSA®, Horizon™, Remote Patron Authentication™, and TeleCirc™ are trademarks of SirsiDynix.

Other product and company names herein may be the trademarks of their respective owners. All titles, versions, trademarks, claims of compatibility, etc., of hardware and software products mentioned herein are the sole property and responsibility of the respective vendors. SirsiDynix makes no endorsement of any particular product for any purpose, nor claims responsibility for its operation and accuracy.

This document is compatible with SirsiDynix Symphony Web Services 3.1. Information in this document may also be compatible with later versions.

# Contents

<b>About this guide</b>	<b>xiii</b>
Summary of contents	xiii
Conventions used in this guide	xiv
Symbols	xiv
Keyboard conventions	xiv
Other typographical conventions	xv
Documentation updates	xv
Comments and suggestions	xv
 <b>Chapter 1: Getting Started</b>	 <b>1</b>
About SirsiDynix Symphony Web Services	1
Licensed operations	2
ClientIDs	2
Web Services base URL	3
Available services	3
Service endpoints	4
Namespaces	4
Web Services terminology	5
Learning about SirsiDynix Symphony Web Services	6
 <b>Chapter 2: Understanding the Symphony ILS</b>	 <b>9</b>
Station library	9
User ID or alternate ID	10
PINs or passwords	11
User access	11
Accountability	12
Volume holds	12
Requests	13
Language	15
Currency	15
Online User Registration	16
 <b>Chapter 3: Working with Web Services</b>	 <b>17</b>
Using SirsiDynix Symphony Web Services	17
Using the sample applications	18

Authenticated operations.....	18
Privileged operations.....	19
Exploring Web Services.....	20
Viewing the WSDLs and schemas.....	20
Testing REST requests with a browser.....	22
Testing SOAP requests with soapUI.....	23
Making SOAP requests.....	24
SdHeader.....	25
Making a SOAP version request.....	26
Making REST requests.....	27
Custom HTTP headers.....	27
Making a REST version request.....	29
Making JSON requests.....	29
Working with Web Services sessions.....	31
Handling faults.....	33
Debugging and Fault Checking with SOAP in Perl.....	34
Fault Checking with REST in Perl.....	35
Searching the Catalog using Perl.....	35
Searching the Catalog using REST.....	35
Interacting with patron data using Perl.....	37
Interacting with patron data using REST.....	37
<b>Chapter 4: Services and Operations.....</b>	<b>41</b>
Standard service.....	42
license.....	43
lookupBestSellersDisplayList.....	44
lookupBestSellerTitles.....	45
lookupHotSites.....	46
lookupMostPopular.....	46
LookupRecommendedReadingDisplayList.....	48
lookupRecommendedReadingTitles.....	49
lookupTitleInfo.....	50
searchCatalog.....	52
searchCatalogPaging.....	55
searchInfoDesk.....	56
searchInfoDeskPaging.....	57
version.....	58
Referenced elements.....	59
BibliographicInfo.....	59
BoundwithLinkInfo.....	60

CallInfo.....	60
HitlistTitleInfo.....	61
ItemInfo.....	61
MarcEntryInfo.....	63
MarcHoldingsInfo.....	64
MostPopularInfo.....	65
TitleAvailabilityInfo.....	65
TitleInfo.....	66
TitleOrderInfo.....	68
Security service.....	69
authenticateUser.....	70
getPassthroughToken.....	71
loginUser.....	71
logoutUser.....	72
lookupSessionInfo.....	73
lookupUserInfo.....	75
Referenced elements.....	76
AccountabilityInfo.....	77
Patron service.....	77
activateOnlineRegisteredUser.....	79
changeMyAlternativeID.....	80
cancelMyHold.....	81
changeMyPIN.....	82
createMyHold.....	82
createSelfRegisteredPatron.....	85
deleteMyList.....	86
enableMyCheckoutHistory.....	87
getMyList.....	88
lookupMyAccountInfo.....	89
lookupMyList.....	92
lookupMyLists.....	93
lookupPatronInfo.....	94
modifyMyHold.....	99
placeMyHold.....	100
renewMyCheckout.....	103
saveMyList.....	104
suspendMyHold.....	105
unsuspendMyHold.....	106
Request service.....	106
cancelRequest.....	107

createRequest .....	108
lookupRequest .....	109
lookupRequests .....	110
lookupRequestTemplate .....	111
Circulation service .....	113
lookupMailDeliveryEstimatesInfo .....	113
recordPayments .....	114
Admin service .....	116
getAccessLevelPolicy .....	119
getAccessRestrictionPolicy .....	120
getBestSellerPolicy .....	121
getBillReasonPolicy .....	121
getCitizenshipPolicy .....	122
getClearancePolicy .....	123
getCreditReasonPolicy .....	124
getGroupResponsibilityPolicy .....	125
getItemTypePolicy .....	126
getLanguagePolicy .....	126
getLibraryGroupPolicy .....	127
getLibraryPolicy .....	128
getLocationPolicy .....	129
getMailServicePolicy .....	130
getNeedToKnowPolicy .....	131
getPaymentTypePolicy .....	131
getPolicy .....	132
getRecommendedReadingPolicy .....	133
getRequestListPolicy .....	134
getRequestStatusPolicy .....	135
getRequestTypePolicy .....	136
getSearchLibraryGroupPolicy .....	137
getUserGroupResponsibilityPolicy .....	138
getUserLookupPolicy .....	139
getUserSortPolicy .....	140
isCheckoutHistoryEnabled .....	141
isMailDeliveryEnabled .....	142
isPatronSelfRegistrationEnabled .....	142
isRestrictedAccess .....	143
isUserGroupsEnabled .....	144
lookupAccessLevelPolicyList .....	144
lookupAccessRestrictionPolicyList .....	145

lookupBestSellersPolicyList .....	146
lookupBillReasonPolicyList .....	146
lookupCitizenshipPolicyList .....	147
lookupClearancePolicyList .....	148
lookupCreditReasonPolicyList .....	148
lookupGroupResponsibilityPolicyList .....	149
lookupILSConfiguration .....	150
lookupItemTypePolicyList .....	151
lookupLanguagePolicyList .....	151
lookupLibraryGroupList .....	152
lookupLibraryPolicyList .....	153
lookupLocationPolicyList .....	153
lookupMailServicePolicyList .....	154
lookupNativeCurrency .....	155
lookupNeedToKnowPolicyList .....	155
lookupPaymentTypePolicyList .....	156
lookupPolicyList .....	157
lookupRecommendedReadingPolicyList .....	158
lookupRequestListPolicyList .....	158
lookupRequestStatusPolicyList .....	159
lookupRequestTypePolicyList .....	160
lookupSearchLibraryGroupList .....	160
lookupUserLookupPolicyList .....	161
lookupUserSortPolicyList .....	162
<b>Chapter 5: Data types reference .....</b>	<b>163</b>
XSL Schema data types .....	163
xs:boolean .....	163
xs:date .....	164
xs:dateTime .....	164
xs:decimal .....	164
xs:int .....	165
xs:long .....	165
xs:nonNegativeInteger .....	165
xs:string .....	165
Web Services data types .....	165
Admin service .....	166
AccessLevelPolicy .....	167
BestSellerPolicy .....	168
BillReasonPolicy .....	168

ClearancePolicy.....	169
CreditReasonPolicy.....	170
ItemTypePolicy.....	170
GroupResponsibilityPolicy.....	171
LanguagePolicy.....	172
LibraryGroupPolicy.....	173
LibraryPolicy.....	173
LocationPolicy.....	176
MailServicePolicy.....	177
PaymentTypePolicy.....	178
PolicyField.....	178
PolicyInfo.....	179
PolicyList.....	180
PolicyRequest.....	181
RecommendedReadingPolicy.....	182
RequestListPolicy.....	183
RequestStatusPolicy.....	183
RequestTypePolicy.....	184
RequiredType.....	184
SearchLibraryGroupPolicy.....	185
SpecialUseType.....	185
UserLookupPolicy.....	186
UserSortPolicy.....	186
Circulation service.....	187
CheckoutHistoryRule.....	187
CheckoutInfoFilter.....	188
FeeInfoFilter.....	188
FeePayment.....	189
FeePaymentProblem.....	189
FeePaymentRecorded.....	190
HoldInactiveType.....	190
HoldInfoFilter.....	191
HoldRange.....	191
HoldType.....	192
MailDeliveryEstimatesInfo.....	192
StatusType.....	193
Common service.....	193
AddressInfo.....	194
Entity.....	194
EntityOperation.....	195



Money.....	195
NillableDate.....	196
Patron service.....	196
FeeInfo.....	197
FeeItemInfo.....	198
FeePaymentInfo.....	199
GroupInfo.....	200
GroupMemberCheckoutInfo.....	202
GroupMemberFeeInfo.....	203
GroupMemberHoldInfo.....	203
GroupMemberInfo.....	204
MyList.....	205
MyListEntry.....	206
MyListEntryInfo.....	206
MyListInfo.....	207
MyListTitleInfo.....	207
PatronAddressInfo.....	208
PatronCheckoutHistoryInfo.....	208
PatronCheckoutInfo.....	212
PatronCirculationInfo.....	215
PatronHoldInfo.....	217
PatronInfo.....	220
PatronStatusInfo.....	221
UserSuspensionInfo.....	222
Request service.....	223
RequestBriefInfo.....	224
RequestEntry.....	224
RequestEntryInfo.....	225
RequestEntryTemplateInfo.....	225
RequestEntryType.....	226
RequestInfo.....	227
Security service.....	227
PrivilegeLevel.....	227
Standard service.....	228
BestSellerDisplayInfo.....	228
BestSellerTitleInfo.....	229
BriefTitleInfo.....	229
CallSummary.....	230
HotSitesInfo.....	231
InfoDesk.....	231

RecommendedReadingDisplayInfo.....	232
RecommendedReadingTitleInfo.....	232
SearchCatalogFilters.....	233
SearchCatalogPaging.....	233
SearchCatalogResult.....	234
<b>Chapter 6: Faults.....</b>	<b>235</b>
HatIOException: Cannot connect to host.....	235
HatIOException: Connection to server has been closed.....	236
HatIOException: Read of response timed out.....	236
HatSessionException: Unable to acquire session with server.....	237
HeaderException: Symphony Soap Header is required.....	237
HeaderException: Bad Client ID.....	237
HTTP 404 File not found.....	238
HTTP 500 Internal server error.....	238
HTTP 501 Not Implemented.....	238
InvalidRequestException: Entity Operation is required.....	239
InvalidRequestException: Invalid Operation for this request.....	239
LicenseException.notLicensed.....	240
RestInvocationException.missingMethod.....	240
SecurityServiceException: Access Denied.....	240
SecurityServiceException: Session has timed out.....	241
SecurityServiceException: Unable to log in.....	241
<b>Appendix 7: Service Schemas.....</b>	<b>243</b>
Standard service.....	243
Security service.....	255
Patron service.....	257
Admin service.....	274
Request service.....	287
Circulation service.....	290
Catalog service.....	295
Common service.....	296
Selection service.....	298
<b>Appendix 8: Service WSDL Definitions.....</b>	<b>301</b>
Standard service.....	301
Security service.....	307
Patron service.....	310
Admin service.....	319
Request service.....	343

Circulation service.....	346
Catalog service.....	347



# About this guide

This *SDK* explains how to use SirsiDynix Symphony Web Services. It provides an overview of the software, including a brief summary of Web Services architecture, and describes the operations available in each service.

This guide is intended for web developers or other technical users with a thorough understanding of web services technology and protocols who want to use SirsiDynix Symphony Web Services to communicate with Symphony systems. For example, you should understand the following concepts and technologies:

- WSDL—basic concepts of services, ports, bindings, port types and the relation to XML schema
- SOAP—basic concepts, relation to WSDL (bindings, services, and so on), encoding types and message styles (SirsiDynix Symphony Web Services uses literal and document)
- REST—basic concepts, particularly HTTP GET and POST transactions and how HTTP headers are used to exchange data and status information
- XML and related technologies—XPath, XML Schema, namespaces, and so on

You can find most specifications at the website of the W3C ([www.w3.org](http://www.w3.org)). Tutorials are available at W3Schools ([www.w3schools.com](http://www.w3schools.com)).

For more information about this guide, see these topics:

- [Summary of contents](#) on page xiii
- [Conventions used in this guide](#) on page xiv
- [Documentation updates](#) on page xv
- [Comments and suggestions](#) on page xv

## Summary of contents

This guide contains these major sections:

- [Chapter Chapter 1, "Getting Started" on page 1](#) provides an overview of the SirsiDynix Symphony Web Services architecture and basic web services information.
- [Chapter Chapter 2, "Understanding the Symphony ILS" on page 9](#) gives definitions and information about how the Symphony configuration affects web services.
- [Chapter Chapter 3, "Working with Web Services" on page 17](#) walks you through the example scripts and teaches you how to make web service calls.



- [Chapter Chapter 4, "Services and Operations" on page 41](#) provides a reference for each of the Web Services operations and how to use them.
- [Chapter Chapter 5, "Data types reference" on page 163](#) defines the data types used throughout Web Services.
- [Chapter Chapter 6, "Faults" on page 235](#) lists many of the faults that may be returned by Web Services and provides information about possible causes and remedial actions.
- [Appendix Appendix 7, "Service Schemas" on page 243](#) lists the schemas for each service in Web Services.
- [Appendix Appendix 8, "Service WSDL Definitions" on page 301](#) lists the WSDL definitions for each service in Web Services.

## Conventions used in this guide

This guide uses terms, typefaces, and symbols to denote different kinds of information. Understanding these conventions can help you recognize the information you need more quickly.

### Symbols

This guide uses these symbols:

Symbol	Description
	<b>Tip</b> This symbol indicates additional notes or helpful tips.
	<b>Important</b> This symbol indicates important information or warnings.

### Keyboard conventions

This guide refers to keys following the standard convention of 101-key keyboards. For example, Ctrl is the control key, Alt is the alternate key, Enter is enter or return.

When you need to press two or more keys at the same time to do an action, the keys are connected with a plus sign (+). For example, **Alt+H** means to press and hold the **Alt** key while you also press **H**.

Keys you press appear in bold type in steps. Text or numbers that you need to enter using the keyboard also appear in bold (for example, "Type **50** in the **Minimum Sessions** field").

## Other typographical conventions

In step-by-step instructions, the names of menus, buttons, fields, and other options appear in bold type (for example, “the **OK** button” or “the **Service Name** field”).

Text in italics shows general information that you must replace with information that is specific to your system. For example, you would replace *host* with a specific host name, such as **example.org**.

File and directory names, code examples, and computer output appear in `Courier`.

## Documentation updates

Updates to this guide are posted to the customer support website between releases, as necessary. These updates provide corrections to unclear, incorrect, or incomplete information. They also provide documentation for enhancements that were not complete at the time the guide was first published.

You may access the customer support website at <http://clientcare.sirsidynix.com>.



The customer support website requires a user name and password. If you do not already have a user name and password, contact your system administrator to receive one.

Documentation updates are available as a PDF (Portable Document Format) file or as HTML help files.

To view a PDF file, you must install Adobe’s Acrobat Reader on your workstation. You can download Acrobat Reader free of charge at Adobe’s web site, <http://www.adobe.com>.

## Comments and suggestions

SirsiDynix welcomes and appreciates your comments on its documentation. We want to know what you think about our manuals and how we can make them better. If you have comments about this guide, please send them to [docs@sirsidynix.com](mailto:docs@sirsidynix.com).

Be sure to include the title and version number of the guide and tell how you used it. Then tell us your feelings about its strengths and weaknesses and any recommendations for improvements.





# Chapter 1: Getting Started

Welcome to SirsiDynix Symphony Web Services. Web Services lets client applications access select features of a SirsiDynix Symphony system through a standards-based interface.

This section contains the following topics:

- [About SirsiDynix Symphony Web Services on page 1](#)
- [Licensed operations on page 2](#)
- [ClientIDs on page 2](#)
- [Web Services base URL on page 3](#)
- [Available services on page 3](#)
- [Service endpoints on page 4](#)
- [Namespaces on page 4](#)
- [Web Services terminology on page 5](#)
- [Learning about SirsiDynix Symphony Web Services on page 6](#)

## About SirsiDynix Symphony Web Services

SirsiDynix Symphony Web Services is a Web application that provides simplified remote access to features of the SirsiDynix Symphony integrated library system (ILS).

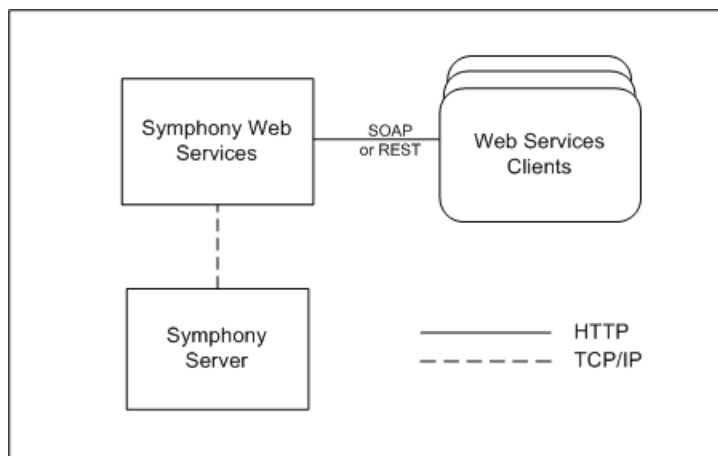
SirsiDynix Symphony Web Services 3.1 is compatible with the following SirsiDynix Symphony ILS versions:

- SirsiDynix Symphony 3.3
- SirsiDynix Symphony 3.3.1
- SirsiDynix Symphony 3.4

Web Services handles requests from clients in both SOAP and REST protocols, providing a language-independent interface to the Symphony ILS. For example, you can implement SOAP or REST clients using PHP, Java, Perl, Ruby, Python, VB, C++, or C# or other .Net frameworks, to name a few common options.

[Figure 1-1](#) illustrates the relationships of various components in the Web Services architecture.

[Figure 1-1: Web Services architecture](#)



#### Related topics

- [Getting Started on page 1](#)

## Licensed operations

Some operations in Web Services are licensed. Currently, all operations in the Patron and Request service require a license. This includes many self-service features such as the ability for library users to place, suspend, or cancel holds; review their account information; renew checkouts; and manage My Lists.

If you want to develop a client that uses these operations, the library's Web Services instance must have a license. Contact the library system administrator to learn what licensed services are enabled.

#### Related topics

- [Getting Started on page 1](#)

## ClientIDs

Every client request to Web Services must include a valid clientID. Client IDs are simple strings that can be passed as a header parameter or as a normal parameter in REST. Library system administrators determine what IDs they will allow to access each instance of Web Services. To obtain a valid ID for a specific instance of Web Services, contact the library system administrator for that Web Services instance.

It is strongly recommend that each client application have a specific ID. In addition to restricting access to a Symphony server, the client ID also provides usage statistics for specific applications.

In Symphony 3.4, the client ID is also important to the User Standing Maps feature. User Standing Maps allow library system administrators to create rules that may restrict or allow access to, for example, holds, account access, or other circulation features, depending on patron status. Consult with the library system administrator to learn how rules for User Standing Maps might affect your client application.

#### Related topics

- [Getting Started on page 1](#)

## Web Services base URL

To access Web Services, you need to know the URL for a SirsiDynix Symphony Web Services instance. The URL is made up of the host name and port and an application name.

For example, if a Web Services host name is *libraryapps.example.org*, and uses the default port, 8080, and the default application name, *symws*, the base URL for that Web Services instance would be:

**`http://libraryapps.example.org:8080/symws`**

Each service end point uses this base URL.

If the server requires SSL, be sure to use the `https:` scheme in the base URL.

#### Related topics

- [Getting Started on page 1](#)

## Available services

SirsiDynix Symphony Web Services is a collection of services organized around similar operations. These services are currently available:

Service	Description
Security	Operations necessary to authenticate library users and initiate user sessions for privileged operations
Standard	Common library features such as searching or retrieving title information
Patron	Patron-facing features such as viewing account information or managing holds
Admin	Operations to access common ILS configuration data
Request	Operations to work with patron requests

For more information about each service and the operations it provides, see [Services and Operations on page 41](#).

### Related topics

- [Getting Started on page 1](#)
- [Services and Operations on page 41](#)

## Service endpoints

Each service has a specific endpoint address. The address varies if you are using REST or SOAP.

The endpoint address is comprised of the Web Services instance base URL (see [Web Services base URL on page 3](#)), the protocol (rest or soap), and the service name (security, standard, patron, or admin).

### SOAP requests

For SOAP requests, you send the request directly to the endpoint address:

`<baseURL>/<protocol>/<service>`

For example:

`http://library.example.org:8080/symws/soap/security`

`http://library.example.org:8080/symws/soap/admin`

For more information about sending requests, see [Making SOAP requests on page 24](#).

### REST requests

For REST requests, you must also append the operation name to the URL, as follows:

`<baseURL>/<protocol>/<service>/<operationName>`

For example:

`http://library.example.org:8080/symws/rest/standard/searchCatalog`

`http://library.example.org:8080/symws/rest/security/loginUser`

For more information about sending requests, see [Making REST requests on page 27](#).

### Related topics

- [Getting Started on page 1](#)

## Namespaces

Namespaces are important to all XML inputs. Web Services will ignore any elements or attributes that are not qualified with expected namespaces.

Web Services uses these namespaces:

Used for	Service	Namespace name
admin.xsd	Admin	http://schemas.sirsidynix.com/symws/admin
circulation.xsd	Patron	http://schemas.sirsidynix.com/symws/circulation
common.xsd	Patron	http://schemas.sirsidynix.com/symws/common
header.xsd	All (SOAP)	http://www.sirsidynix.com/xmlns/common/header
patron.xsd	Patron	http://schemas.sirsidynix.com/symws/patron
security.xsd	Security	http://schemas.sirsidynix.com/symws/security
standard.xsd	Standard	http://schemas.sirsidynix.com/symws/standard
REST exceptions	All (REST)	http://schemas.sirsidynix.com/rest
SOAP exceptions	All (SOAP)	http://schemas.sirsidynix.com/symws/exception

### Related topics

- [Getting Started on page 1](#)

## Web Services terminology

As you work with SirsiDynix Symphony Web Services, it will be helpful to understand these terms.

### session

In Web Services, a session is a collection of cached data associated to a session user by means of a session token. Data includes the user's privilege level, library, user access, accountability, or other values passed with requests to the Symphony server that may affect session-sensitive operations.

### session user

A session user is the user associated with a session token. To obtain the sessionToken element, you use the Security service loginUser operation. The logged in user is the session user.

The session user is the subject or actor in the request, and can be different than a user that is the object of a request. For example, the session user for a lookupUserInfo request may be a staff user looking up information about another user in the ILS.

### session token

A session token is a universally unique token received at login and used to identify an authenticated user and any related session values on the Web Services server. The session token is sent with session-sensitive operations to identify the user and associated privileges. This token is sent in the sessionToken property of the SdHeader for SOAP requests. For REST requests, it is sent as a URL

parameter with the key of “sessionToken=” or it is sent as a custom HTTP header keyed with “x-sirs-sessionToken”.

#### **session-sensitive operation**

Any operation where some aspect of session data affects what data is retrieved or how data is processed. For example, some privileged requests require an authenticated user in order to determine that sufficient privileges exist to complete the operation. The operation will fail otherwise.

#### **self-service operation**

A privileged request where the session user is also the object of the request, lookupMyAccountInfo in the Patron service, for example. All of the My operations in the Patron service are self-service operations.

#### **privileged request**

Any request that requires authentication (that is, it requires a session token). If the sessionToken property is absent, or if the session user does not have adequate privileges, the request will fail.

#### **authenticated request**

Any request that is sent with a sessionToken to specify something other than the default session environment.

Sending a sessionToken in an unprivileged request can affect the results depending on the operation. For example, if you include a sessionToken for an accountable user with a searchCatalog request, the response will include titles and items that would otherwise be restricted due to accountability.

#### **unprivileged request**

Any request that does not require authentication (that is, it does not require a sessionToken be sent in the header). Without a sessionToken, any ILS environment or session settings are derived from default settings specified on the Web Services server.

#### **accountability operator**

The optional Accountability module in the Symphony ILS allows libraries to restrict access to users and materials based on security clearance and need-to-know privileges. Only an accountability operator (a user with a User Access Policy with Accountability Operator enabled) can view or modify accountability data for accountable users or materials.

#### **accountable user**

A user with an accountability record (clearance, need to know, and so on).

#### **Related topics**

- [Getting Started on page 1](#)

## **Learning about SirsiDynix Symphony Web Services**

There are several ways you can learn about SirsiDynix Symphony Web Services.

## Web Services Developer's Guide

This *SDK* is the best source of information about developing with SirsiDynix Symphony Web Services. Use the table of contents and index for quick access to specific topics.

## Web Services WSDLs and schemas

Information about messages and operations in Web Services, including expected data in requests and responses, can be found in the service description documents (WSDLs and schemas) for each service.

These documents are useful in understanding both SOAP and REST implementations. Even though SirsiDynix Symphony Web Services does not provide a WADL for REST operations, the schemas can be used to understand parameters of requests and responses. In most cases, the REST input parameters match the element names in a request. (Query groups in searchCatalog are the exception; see [searchCatalog on page 52](#).)

Comments in the schemas provide valuable usage information.

For more information about how to use the schemas to understand SirsiDynix Symphony Web Services, see [Exploring Web Services on page 20](#).

## Symphony documentation

Because SirsiDynix Symphony Web Services is an interface to the Symphony ILS, it may be helpful to understand specific features of the ILS. If you have access to a Symphony system, the online help for the WorkFlows client for Java is a good source of information for specific features of the ILS. If you do not have access to a Symphony system, consult with the library system administrator for information about features you need to work with.

## General Web service, SOAP, XML reference

This guide assumes that you already understand Web services technologies. If you are new to working with Web services, you might find the following resources helpful for additional background:

- <http://www.w3.org/TR/ws-arch/> — This technical report of the W3C Web Services Architecture working group provides a good introduction to core concepts and is a good starting point for exploring the W3C normative recommendations for Web services.
- <http://www.w3schools.com/webservices/> — A very brief tutorial with a working example of a Web service.
- <http://www.ibm.com/developerworks/webservices/newto/service.html> — provides introductory information about Web services and Service Oriented Architectures as well as links to other resources.

## Related topics

- [Getting Started on page 1](#)





# Chapter 2: Understanding the Symphony ILS

As you create applications, it will be helpful to keep in mind how Symphony configuration affects Web Services operations.

If you have access to a Symphony WorkFlows client for Java (the user interface to the ILS), you can learn more about the features described here by referring to the WorkFlows online help.

This section contains these topics:

- [Station library](#) on page 9
- [User ID or alternate ID](#) on page 10
- [PINs or passwords](#) on page 11
- [User access](#) on page 11
- [Accountability](#) on page 12
- [Volume holds](#) on page 12
- [Requests](#) on page 13
- [Language](#) on page 15
- [Currency](#) on page 15
- [Online User Registration](#) on page 16

## Station library

As you work with the Symphony ILS, it is important that you understand how station library influences requests made through Web Services to the ILS.

### Purposes of station library

The station library (also called “login library”) can change behavior in Symphony client applications. This applies to clients such as WorkFlows or Web Services.

- **Logging and statistics.** In transactions that modify ILS data, station library is recorded in the ILS history logs and can influence statistics. For example, when placing a checkout renewal through Web Services, the station library is logged for the request.
- **Permissions or filtering.** In some transactions, the station library can limit the data available to a request. For example, the library may limit circulation or user data by using specific libraries or library groups for display and maintenance. In that case, a response will only display data based on that configuration.

- **Order of results.** In some cases, the station library can change the order of results. For example, holdings for the station library may appear first in a call number list.
- **Workflows.** Depending on the ILS configuration, the station library can influence behavior in the ILS. For example, in holds the pickup library may default to the station library. This may in turn affect transit rules or other logic that affects how holds are filled or what materials are available to hold. In this case, a hold that might succeed for one station library could fail for a different station library.

### Default station library

A Web Services instance has a default station library depending on the Station policy and user specified in Web Services configuration parameters. The default station library is passed to the ILS for operations where you do not specify a session token.

### User-specific station library

You obtain a session token by logging in a specific user. The login operation allows you to optionally specify a workstation ID (Station policy). If no workstation ID is specified, the default Station policy is used for the login operation.

If the Station policy used for login is of type Floating, the user's library becomes the station library for requests with that session token.

If the Station policy is of type Port or Named, the user's library has no effect on station library. Instead, the library assigned to the Station policy is always the station library.

Talk to the library system administrator to learn more about how station library is used in that library's ILS configuration and to learn whether you might want to use a specific workstationID for login.

### Related topics

- [Understanding the Symphony ILS on page 9](#)

## User ID or alternate ID

The ILS Allow Lookup By User Alternative ID policy determines if the ILS will allow lookup by an alternate ID as well as by user ID.

Web Services will accept either a user ID or an alternate ID to authenticate a user. Here are some things you might need to know about the library's use of user IDs or alternate IDs:

- Whether the system uses alternate IDs
- What values the library uses for user IDs or alternate IDs, for example, barcodes, user names, external identification numbers (such as license or social security number), and so on

- Any specific patterns your application might use to validate the ID (for example, all numbers, a combination of numbers and dashes, and so on)

This information may help you in defining an interface to prompt for login credentials.

#### Related topics

- [Understanding the Symphony ILS on page 9](#)

## PINs or passwords

Most libraries require PINs to authenticate users, some do not. If the library has configured LDAP servers to use for Web Services authentication, passwords are required.

While Web Services will accept PINs or passwords, it will be helpful for you to know which is used if your application prompts for login credentials (to know what labels to use in an interface, for example).

It may also be helpful to know if there are specific patterns used for PINs (when PINs are autogenerated, for example). This information can help you if you choose to do preliminary input validation or prompt a user about an expected pattern.

#### Related topics

- [Understanding the Symphony ILS on page 9](#)

## User access

User access in Web Services follows the model in SirsiDynix Symphony. What a given user can see and do in the system depends on the user's library (see [Station library on page 9](#)) and the user's user profile. If the library has enabled Accountability in the ILS, this will also affect what users can see (see [Accountability on page 12](#) for details).

The ILS User Access policy defines a user's privileges in three important ways: user level, access type, and commands list.

You can view these settings for a user using the `lookupUserInfo` operation of the Security service.

#### User level

The user level (also called "privilege level" in Web Services) indicates, for example, if the user is a library patron or a library staff member. Based on user level, certain features may be restricted. For example, a PUBLIC user cannot see data for other users in the library.

When your client application needs to work with data for users, it will need to create a Web Services session for a user that has a privilege level of STAFF or higher or create separate sessions for each user (see [Working with Web Services sessions on page 31](#)).

#### Access type

Access type specifies if the user access policy is for an individual user (Personal) or shared by a group of users. Shared access places restrictions on what the user can do, for example, modifying account data. (You can use the `lookupUserInfo` operation in the Security service

### Command list

The command list associated with a user access policy specifies exactly what a user can do in the system.

For example, if your client application creates a user session and uses that session for a `lookupUserInfo` request, the operation will fail if the user in the session token does not have the `Display2 User` command in the profile's command list.

The [lookupUserInfo](#) operation requires specific commands.

### Related topics

- [Understanding the Symphony ILS on page 9](#)

## Accountability

Accountability is controlled at the ILS server. If the library has enabled Accountability in the ILS, this will affect Web Services operations.

For example, some operations, such as search catalog, can be performed without specifying a user. If no user is specified, the search catalog response will only include hits for titles that are not subject to accountability. Any accountability titles will be suppressed.

A client application can, however, include a session token for an accountable user with a search catalog request. In this case, the response will include any accountability titles that match that user's accountability clearance.

The same is true for the lookup title info operation. The operation will fail or will suppress accountability items depending on the clearance of the session user.

The lookup user info operation allows a client application to access accountability information for a specified user. This operation will fail if the user in the session token is not an accountability operator.

If the library uses Accountability, it will be important to discuss with the library system administrator the implications of accountability on the client application.

### Related topics

- [Understanding the Symphony ILS on page 9](#)

## Volume holds

When a patron makes a hold request, the hold is placed by default on the first call number attached to a bibliographic record. For multi-volume sets, the hold is usually

placed on volume 1. However, if the patron wants to get the third volume of a set, they must be able to specify which volume number to place a hold on.

In Symphony, volume information is stored in what is called the Analytic Z portion of the call number.

To determine whether a title has an Analytic Z portion that requires a volume-level hold, turn on the `includeCallNumberSummary` element in your `LookupTitleInfoRequest` (for the `lookupTitleInfo` operation). This returns a list of `CallSummary` objects in the `TitleInfo` object. The presence of data in the Analytic Z field of the `CallSummary` then indicates that a hold request requires the call number to be passed in.

In order to place a volume-level hold, use the `createMyHold` or `placeMyHold` operation and as part of the request pass in the call number (that has Analytic Z as part of the `CallSummary`) for the specified title. This triggers the volume-level hold request.

### Related topics

- [createMyHold](#)

[placeMyHold](#)

## Requests

The Requests feature of SirsiDynix Symphony allows Library users to communicate their needs or requests to the library by sending a particular type of request through e-Library, in person, or through Web Services. Each request can be associated with a particular user, item, or both, depending on its request type. If the request is associated with a user, the WorkFlows wizards can be used to reply to the user. For more information about Requests, see the Requests chapter of the WorkFlows online help.

In order to use Web Services to work with Requests, you need to understand the entire workflow of the available operations.

1. First, use the `lookupRequestListPolicyList` operation to find the policyIDs of the different Request List policies that are available. For example, STAFF and PUBLIC.

This is important because each Request List has a defined set of requests that can be made. So, if I want to build an interface that asks a patron to select the kind of request they want to make, then I need to use the appropriate Request List (PUBLIC in this case).

2. Use the `getRequestListPolicy` operation to get the list of all the requestTypes that are available for a specified Request List policyID (see step 1). This requestType becomes an ID for subsequent requests, so we'll need to store it.

**Note:** The `lookupRequestTypePolicyList` operation retrieves a list of all the possible requestTypes. You can use the `policyDescription` that is returned here for displaying a choice for the user. Therefore, you would cross-reference the `requestType` of `GetRequestListPolicyResponse` with the `policyID` of `LookupRequestTypePolicyListResponse`.

3. You use the previous two steps to display a list of request options to a user. After the user selects one of the options, we need to call `getRequestTypePolicy` using

the selected requestType as the policyID of the operation's request. From the information returned, we need to know the value of requestTextRequired so that we know if the Request the user will be making requires a VED (view, edit, delete). The value may be MANDATORY (a VED is required), OPTIONAL (a VED can be supplied if wanted), or NEVER (any VED data supplied will be ignored).

**Note:** The other data returned in this operation are important, but can be retrieved through other operations.

4. Now we need to determine the fields to display to the user. To do this, use the **lookupRequestTemplate** operation.

This returns information about whether a userID or catalogKey are required for the Request (if they are, therefore, we must make sure to get that information and include one or both as part of the Request). This operation also returns the default status of the request. You can specify this or another appropriate value (you can get a list of available status values using **lookupRequestStatusPolicyList**).

The operation also returns a requestEntryInfo (RequestEntryTemplateInfo type) for each field to display to the user in the request form. The entryDescription can be used as the field label. If allowedEditCreate is false, then you should only display the data (rather than let the user change it). Use the required element to help determine which fields the user must supply values for and which you might want to provide validation for. The dataType, minimumSize, and maximumSize elements are also important for validation purposes.

The entryID and absoluteEntryNumber values will be used in the createRequestRequest, so save those with each field as well.

5. After the user supplies values for the fields on the request form, the next operation to call is **createRequest**. This operation requires the sessionToken from an authenticated user who has staff level or higher privileges. Therefore, you must first call **loginUser** for someone other than the patron. Perhaps you want to use a shared staff account or a Web Services-specific staff account to do this.

As part of CreateRequestRequest, make sure to include values from any required fields. For example, the userID or call number or VED data.

If the request is successful, the system returns the requestKey of the request.

6. Using the requestKey, you can call **lookupRequest** to display the information that the system received.

You can also use **lookupRequests** to get a list of all the requests submitted by the specified user.

This is a basic workflow for using the requests feature. However, there are some other operations and options available in the service that you may want to take advantage of depending on your library and the configuration of the ILS.

## Language

SirsiDynix Symphony Web Services allows client applications to specify a preferred language for requests as part of the header (see [SdHeader on page 25](#) for SOAP or [Custom HTTP headers on page 27](#) for REST). For example, if a client specifies de\_DE as the locale of the request, Web Services will return strings or messages in German if that language is supported by the ILS.

If Web Services cannot find translations for a given locale, it will return strings from the ILS default language policy.

For details about the Language Wizard and configuring Language policies, refer to the help for the WorkFlows Client for Java.

When only the language code is specified and not the country code, that Language policy will be used for all variants of that language. For example, a Language policy with the “de” language code and no country code will be used for de\_DE, de\_AT, de\_CH, and so on.

You only need to specify a country code if you want to restrict that Language policy to a specific locale.

If it appears that changes to Language policies are not reflected in Web Services, you may need to reset the ILS caches in Web Services.

### Related topics

- [Understanding the Symphony ILS on page 9](#)

## Currency

All requests and responses that specify monetary values use a Money type to represent these values. A required attribute of the Money type is “currency” and should be the ISO-4217 currency code that represents the money value. For example, the currency code USD, which represents United States Dollars.

If currency codes received from a Web Service instance are not a valid ISO-4217 code, contact the Web Service administrator about configuring the Web Service Currency mappings.

To identify what the native currency code is for a site, use the [lookupNativeCurrency](#) operation from the admin service. Most requests you make that specify monetary values need to include this native currency code.

For information about currency mappings, see the the help for the WorkFlows Client for Java or the *SirsiDynix Symphony Web Services 3.0 Setup Guide*.

**Related topics**

- [Understanding the Symphony ILS on page 9](#)

## Online User Registration

SirsiDynix Symphony Web Services provides an operation in the patron service that allows for patron self-registration. In order for this feature to work, Online User Registration must be enabled and properly configured on the ILS server.

To enable and configure Online User Registration, you must contact SirsiDynix and refer to the Online User Registration guide available in the iBistro/iLink/e-Library documentation section of the SirsiDynix Client Care website.

The Web Services create self-registered patron operation is only available for SirsiDynix Symphony 3.4 and later.

**Related topics**

- [Understanding the Symphony ILS on page 9](#)



# Chapter 3: Working with Web Services

After you understand the basics of how web services work (see [Getting Started on page 1](#)), you are ready to getting started using web services to send requests to and get responses from your ILS server.

If you do not understand how the Symphony ILS affects web service requests and responses, you should first read [Understanding the Symphony ILS on page 9](#).

This section contains these topics:

- [Using SirsiDynix Symphony Web Services on page 17](#)
- [Authenticated operations on page 18](#)
- [Privileged operations on page 19](#)
- [Exploring Web Services on page 20](#)
- [Making SOAP requests on page 24](#)
- [Making REST requests on page 27](#)
- [Making JSON requests on page 29](#)
- [Working with Web Services sessions on page 31](#)
- [Handling faults on page 33](#)
- [Searching the Catalog using Perl on page 35](#)
- [Interacting with patron data using Perl on page 37](#)

## Using SirsiDynix Symphony Web Services

As you work with SirsiDynix Symphony Web Services, there are some general things to keep in mind.

- Every request you submit to Web Services must include a SirsiDynix header or certain HTTP header values. If you do not include them, the request will be rejected. For more information, see [SdHeader on page 25](#) for SOAP or [Custom HTTP headers on page 27](#) for REST.
- Many operations in Web Services require an authenticated user by specifying a session token. For specific information about this, see [Authenticated operations on page 18](#).
- Many other operations in Web Services require a specific user with specific permissions. For more details, see [Privileged operations on page 19](#).
- If an operation cannot be completed, Web Services will usually return a fault message. For information, see [Handling faults on page 33](#) and [Faults on page 235](#).

- As with all web services, you need to be careful to use the appropriate namespaces for the requests you submit. For more information, see [Namespaces on page 4](#).
- Make sure you know whether the site uses secured HTTP or unsecured so that you can connect correctly. This is also important to keep in mind if you are developing an application that may connect to many different servers where some may be secured and some not. Also, you may want to ensure that a secure connection is used for requests that send or return sensitive data such as login information.
- Special or reserved characters need to be URL encoded when using REST. For example, if a user may use special or reserved characters in a password, such as "h%llo", you would send something like the following login request:  
`http://libraryapps.example.org/symws/rest/security/loginUser?clientId=SymWSTestClient&login=JoeB&password=h%25llo`

## Using the sample applications

In order to help you get started with Web Services, SirsiDynix provides several sample applications. They are available for Java, C Sharp, Perl, Javascript and JSON, and VB as part of the SDK package or through this SDK's web site. Most of the applications contain comments that can help you learn how to use Web Services. In the following sections, we will go into more depth on what the examples are doing and why.

For the applications, there are often variables defined near the top of the code that you must modify to match your configuration.

For example, you may need to specify your BASE\_URL of your web services application, the clientId, and login information (userID, password, and userLibrary) for a user.

### Related topics

- [Working with Web Services on page 17](#)

## Authenticated operations

Some operations in Web Services require a user. These are called authenticated operations. For these operations, you must have first logged in the user and received a session token.

Then as part of the request for the authenticated operation, you include the session token for the authenticated user (see [Working with Web Services sessions on page 31](#)). If an operation also requires specific user privileges, it is known as a privileged operation. For the differences, see [Privileged operations on page 19](#).

Additionally, if the **Always Require Authentication** option is enabled in the Web Services Admin, then all operation requests (except [isRestrictedAccess](#), [license](#), [loginUser](#), and [version](#)) will require a valid sessionToken.

For example, with this option enabled, the normally un-authenticated [searchCatalog](#) request would require a session token.

Use the [isRestrictedAccess](#) operation to determine whether the **Always Require Authentication** setting is turned on or off.

If an authenticated operation request does not include a valid sessionToken element, the system returns an Access Denied exception (see [SecurityServiceException: Access Denied](#) on page 240).

The PatronSamples example shows how to make the login request and return the session token for use in authenticated operations. See [Interacting with patron data using Perl](#) on page 37 for a specific example.

#### Related topics

- [Working with Web Services](#) on page 17
- [Services and Operations](#) on page 41
- [Privileged operations](#) on page 19
- [Working with Web Services sessions](#) on page 31

## Privileged operations

While [Authenticated operations](#) only require a user, some operations in Web Services require a user and specific user permissions. These are called privileged operations.

As with authenticated operations, for a privileged operation you must have first logged in the user and received a session token. But unlike authenticated operations, in a privileged operation, the user must also have additional SirsiDynix Symphony user rights. These rights might be that the user has a privilege level of Staff or higher, or has command code rights to execute the request, or both.

Then as part of the request for the privileged operation, you include the session token for the user. If the session token is valid and the user has sufficient rights, the request is processed (see [Working with Web Services sessions](#) on page 31).

The [lookupUserInfo](#) operation is an example of a privileged operation that requires a session token and sufficient user rights.

See [Command list](#) on page 12 for information on the rights required for each privileged operation.

The PatronSamples example shows how to make the login request and return the session token for use in privileged operations. See [Interacting with patron data using Perl](#) on page 37 for a specific example.

**Related topics**

- [Working with Web Services](#) on page 17
- [Services and Operations](#) on page 41
- [Authenticated operations](#) on page 18
- [Working with Web Services sessions](#) on page 31

## Exploring Web Services

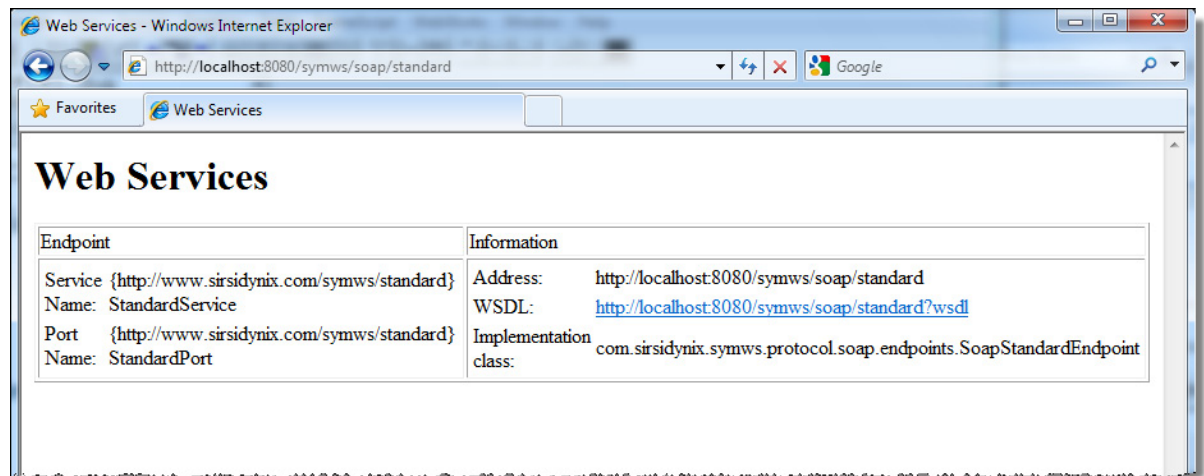
One of the best ways to learn about SirsiDynix Symphony Web Services is to browse the service definition documents (WSDLs and schemas) and manually try various operations in a Web browser or a tool such as SOAP UI.

This section contains these topics:

- [Viewing the WSDLs and schemas](#) on page 20
- [Testing REST requests with a browser](#) on page 22
- [Testing SOAP requests with soapUI](#) on page 23
- [Working with Web Services](#) on page 17

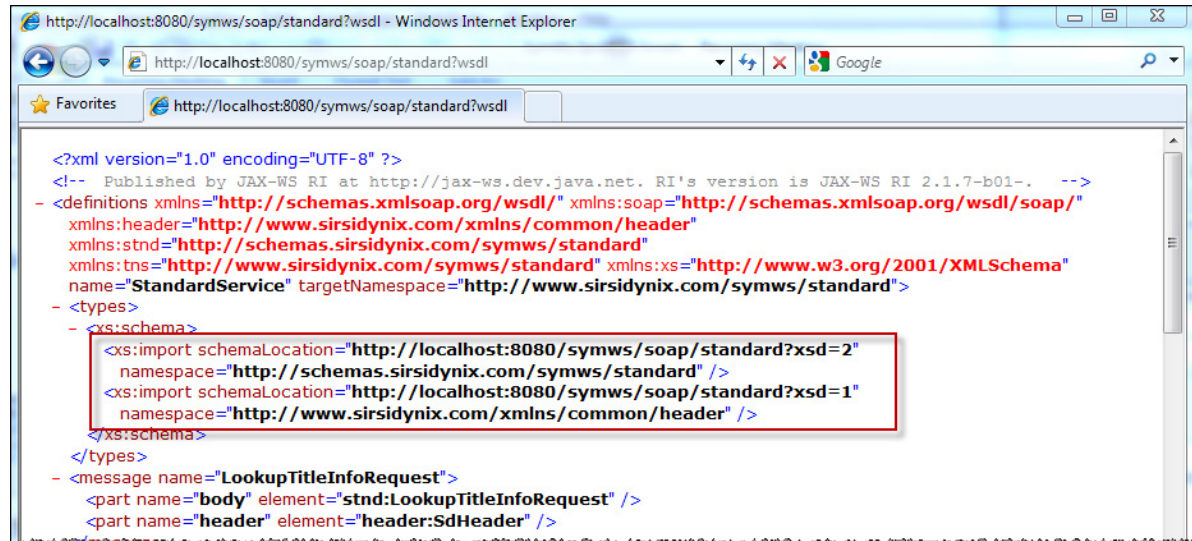
### Viewing the WSDLs and schemas

If you use a browser to access the SOAP endpoint for a service (see [Service endpoints](#) on page 4), Web Services displays a summary of the service, including the URL for the WSDL that defines that endpoint.



The WSDL URL is what you would enter into a SOAP-aware tool such as SOAP UI or an IDE such as Microsoft Visual Studio to automatically create request templates or generate classes for that service. In this example, that URL is `http://localhost:8080/symws/soap/standard?wsdl`.

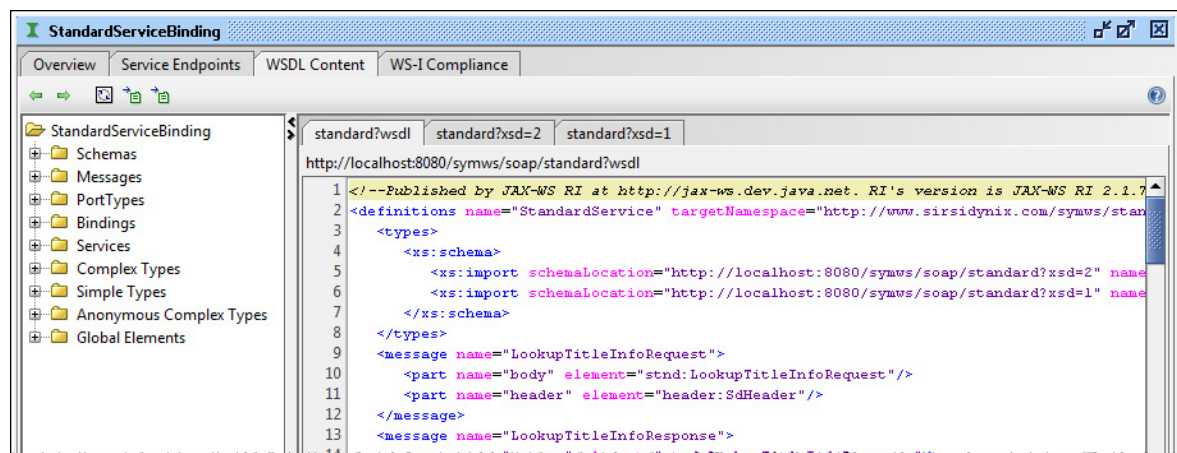
When you follow the link for the WSDL, Web Services returns the WSDL document for that service.



The WSDL imports the schemas that define the structures and elements of requests and responses in the service.

Copy and paste the URL from the schemaLocation attribute into your browser's address bar to view that schema. In this example, the URL `http://localhost:8080/symws/soap/standard?xsd=2` retrieves the standard schema and the URL `http://localhost:8080/symws/soap/standard?xsd=1` retrieves the header schema.

SOAP UI (an open-source Web service testing tool available at [www.soapui.org](http://www.soapui.org)) also allows you to easily view the WSDLs and schemas that define a Web service using the built-in Interface Viewer.



**Related topics**

- [Exploring Web Services on page 20](#)
- [Testing REST requests with a browser on page 22](#)
- [Testing SOAP requests with soapUI on page 23](#)

**Testing REST requests with a browser**

Because most REST requests are simply an operation URL with basic query parameters, you can test most REST requests with a Web browser. (Operations that require XML input, such as the `saveMyList` operation, are the exception.)

The URL for a REST request that you put in your browser looks like this:

`<baseUrl>/rest/<service>/<operation>?<parameters>`

`<baseUrl>` is the base URL for the Web Services instance (see [Web Services base URL on page 3](#)).

`<service>` is the name of the service (see [Available services on page 3](#)).

`<operation>` is an operation name from the WSDL for that service (see [Services and Operations on page 41](#) for the operations available for each service).

`<parameters>` are the elements from the service schema that provide input values for the request, including any required header parameters. They are specified as a name/value pair (such as, `clientID=SampleClient`). Each parameter is separated by an ampersand (&).

For example: `clientID=SampleClient&titleID=54321&libraryFilter=MAIN`

*Sending a version REST request through a browser*

One of the first web service calls you may make is to verify that Web Services is running and to get the version information for Web Services and Symphony. This request does not require any parameters with the operation name.

You can make this request and get the response through a web browser.

To get the version information, if your Tomcat host name is *libraryapps.example.org*, and you used the default port, 8080, and had the default application name, *symws*, the URL for a version request would be:

`http://libraryapps.example.org:8080/symws/rest/standard/version`

If Web Services is running and properly configured, the software returns an XML document with version information about Symphony and Web Services, for example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<VersionResponse xmlns="http://schemas.sirsidynix.com/symws/standard">
  <version>
    <product>SYMPHONY</product>
    <version>3.4.0.6013</version>
  </version>
  <version>
    <product>SYM-WS</product>
```

```
<version>v3_0_1818 2010-09-23 04:02:26</version>
</version>
</VersionResponse>
```

For information on how to make this request and what to expect, see [version on page 58](#) and the *Web Services Setup Guide*. That guide also provides some troubleshooting information.

### *Sending a searchCatalog REST request through a browser*

Another operation that you can use to test your requests through a web browser is searchCatalog.

To get the results of the [searchCatalog](#) request, if your Tomcat host name is *libraryapps.example.org*, and you used the default port of 8080, and had the default application name of *symws*, and used the default test clientID of *SymWSTestClient*, and wanted to get the first 5 results of an author search for “William Shakespeare”, the URL for the request would be the following:

```
http://-
/libr-
aryapps.example.org:8080/symws/rest/standard/searchCatalog?clientID=SymWSTestClient&term1=William
Shakespeare&filter=author&hitsToDisplay=5
```

If Web Services is running and properly configured, the software returns an XML document with lots of information about the 5 results. This is useful for finding out what information can be retrieved from the catalog and how the catalog item information is set up. See [Searching the Catalog using Perl on page 35](#) for a specific example of using Perl from an application rather than the browser URL.

Also, notice that the parameter “term” has had a number appended to it. This is allowed (and expected) for parameters where more than one are allowed in a request. Check the details of each parameter in the operation to learn more about the cardinality.

### **Related topics**

- [Exploring Web Services on page 20](#)
- [Viewing the WSDLs and schemas on page 20](#)
- [Testing SOAP requests with soapUI on page 23](#)

## **Testing SOAP requests with soapUI**

Eviware provides helpful documentation for using either the free or Pro version of the soapUI software. Refer to the online *Getting Started Guide* or *User Guide* (see [www.soapui.org](http://www.soapui.org)) for specific help with the soapUI software.

To get started with the software so you can test SOAP requests for SirsiDynix Symphony Web Services, you need to create a project and add all the WSDLs. Then, you can modify or add appropriate requests for testing.

### **To create a project and add WSDLs**

- 1 Create a new project by going to **File > New soapUI Project**.

- 2 Provide a Project Name, such as `SymWS`.
- 3 Click **OK** to create the project.
- 4 Add each of the four WSDLs to your new project by right-clicking the project name and choosing **Add WSDL**.
- 5 Type the URL location of the WSDL. For example:  

```
http://localhost:8080/symws/soap/standard?wsdl
```

The URL follows the format of `<baseURL>/soap/<service>?wsdl` where `<baseURL>` is the base URL for the Web Services instance (see [Web Services base URL on page 3](#)) and `<service>` is the name of the service (see [Available services on page 3](#)) -- admin, circulation, patron, security, or standard.
- 6 Make sure **Create sample requests for all operations?** is checked.
- 7 Click **OK**.
- 8 Repeat steps 4-7 as needed to add all four WSDLs.
- 9 Go to **File > Save All Projects**.

After your project is set up, you are ready to start testing various requests.

The following are some tips for making the requests work:

- Each request must have a value for clientID. Replace the question mark (?) in the XML with the appropriate clientID. See [ClientIDs on page 2](#).
- For requests that do not require a session token, remove the question mark (?) from the XML. Failing to remove the question mark will result in an error message.
- You may need to remove question marks or provide values for certain operations.
- When you are ready to send the request, click the green arrow icon in the Request window. The Response displays in the Response pane or tab (depending on how you have configured the display).

#### Related topics

- [Exploring Web Services on page 20](#)
- [Viewing the WSDLs and schemas on page 20](#)
- [Testing REST requests with a browser on page 22](#)

## Making SOAP requests

Each SOAP request must contain certain information in order for Web Services to accept it.

You must provide an appropriate endpoint for the request. For details, see [Service endpoints on page 4](#).

You also need to specify the appropriate namespace. See [Namespaces on page 4](#).



Finally, you must include a SirsiDynix header. For information on what can be included, see [SdHeader on page 25](#).

### Related topics

- [SdHeader on page 25](#)
- [Making a SOAP version request on page 26](#)
- [Working with Web Services on page 17](#)

## SdHeader

Every SOAP request you submit to Web Services must include a SirsiDynix header (SdHeader element, header.xsd). SdHeader includes these elements:

Element	Description
<b>clientID</b>	<p>The clientID element specifies the client identifier for your application. Your client must provide a valid clientID with each request. If the clientID is invalid, the request will be rejected.</p> <p>Contact the library system administrator to obtain a clientID for your application.</p> <p>clientID is a string (see <a href="#">xs:string on page 165</a>).</p>
<b>locale</b>	<p>The locale element specifies the locale that Web Services should use when returning the response. This may be a language code only (for example, “fr”) or a language and country code (for example, “fr_CA”).</p> <p>Case is significant. Language codes are lowercase, country codes uppercase. (For more information, see <a href="#">Language on page 15</a>.)</p> <p>locale is a string (see <a href="#">xs:string on page 165</a>).</p> <p>This element is optional.</p>
<b>sessionToken</b>	<p>The sessionToken element identifies an active Web Services session for a specific user with specific privileges and (optionally) a specific station policy (workstationID).</p> <p>A valid session token must be included in the message header for all privileged operations (see <a href="#">Privileged operations on page 19</a>).</p> <p>A Web Services client obtains a session token by using the loginUser operation (see <a href="#">loginUser on page 71</a>).</p> <p>You end a session using LogoutUserRequest (see <a href="#">logoutUser on page 72</a>).</p> <p>sessionToken is a string (see <a href="#">xs:string on page 165</a>).</p>

The following shows a sample SOAP request, including the required header:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:head="http://www.sirsidynix.com/xmlns/common/header"
  xmlns:sec="http://schemas.sirsidynix.com/symws/security">
  <soapenv:Header>
    <head:SdHeader>
      <head:clientID>SampleClientID</head:clientID>
      <head:locale>en</head:locale>
      <head:sessionToken>a9e4341c-2697-4ac6-bb04-596940215e01<
head:sessionToken>
    </head:SdHeader>
  </soapenv:Header>
  <soapenv:Body>
    <sec:LookupUserInfoRequest>
      <sec:userID>USER</sec:userID>
```

```

</sec:LookupUserInfoRequest>
</soapenv:Body>
</soapenv:Envelope>

```

Other elements defined in SdHeader (params, param, and version) are reserved for future expansion and are not currently implemented in Web Services. For information on how to implement this header in your application, see the sample applications or refer to the information in [Searching the Catalog using Perl on page 35](#) and [Interacting with patron data using Perl on page 37](#).

### Related topics

- [Making SOAP requests on page 24](#)

## Making a SOAP version request

The VersionSamples example also shows how to set up the SOAP request with all the required information.

Let's walk through the important parts of the Perl code.

We'll start with lines 25-26.

```

my $standardService = SOAP::Lite -> service($BASE_URL.'soap/standard?wsdl')
->proxy($BASE_URL.'soap/standard');

```

These two lines define the service to use. For this example, [version](#) is part of the [Standard service](#).

You also need to specify the header (as indicated in [SdHeader on page 25](#)). Lines 28-31 do this, as follows:

```

my $header = SOAP::Header->name('SdHeader' =>
\SOAP::Header->value(
SOAP::Header->name(clientID => $clientID)->type('')
))->attr({xmlns => 'http://www.sirsidynix.com/xmlns/common/header'});

```

These lines specify that you are creating a header called SdHeader. For this header, we only want to send the clientID. In the final line, you reference the header namespace.

In the next set of lines (33-36), you specify the request and the namespace.

```

my $method = SOAP::Data
->name('VersionRequest')
->prefix('stan')
->uri('http://schemas.sirsidynix.com/symws/standard');

```

VersionRequest is the name of the request. The prefix must correspond to the service type (adm for admin, pat for patron, sec for security, or stan for standard). Last, you qualify the namespace. In this case, the version operation is part of the standard service.

Next, you create the request and send it to a variable for the response.

Finally, you iterate through each set of responses in lines 51-53:

```
foreach($response->dataof('//VersionResponse/version/*')) {
    print $_->name . ": " . $_->value . "\n";
}
```

In the foreach loop statement, you specify the response variable and only look at the requested data in the version node of the returned XML. The print statement merely displays the name/value pair information for each version sub-node.

See the VersionSamples example for the rest of the code.

## Making REST requests

Each REST request must contain certain information in order for Web Services to accept it.

You must provide an appropriate endpoint for the request. For details, see [Service endpoints on page 4](#).

You may want to include a custom header in order to avoid passing potentially sensitive information through the URL or that may show up in logs. For information on what can be included, see [Custom HTTP headers on page 27](#).

### Related topics

- [Custom HTTP headers on page 27](#)
- [Making a REST version request on page 29](#)
- [Working with Web Services on page 17](#)

## Custom HTTP headers

REST requests require Web Services parameters in addition to the operation-specific parameters you provide for each request. In general, these parameters are passed in using the following custom HTTP headers:

Header	Description
<b>x-sirs-clientID</b>	<p>The clientID specifies the client identifier for your application. Your client must provide a valid clientID with each request. If the clientID is invalid, the request will be rejected.</p> <p>Contact the library system administrator to obtain a clientID for your application.</p> <p>clientID is a string (see <a href="#">xs:string on page 165</a>).</p>

Header	Description
<b>x-sirs-locale</b>	<p>The locale specifies the locale that Web Services should use when returning the response. This may be a language code only (for example, “fr”) or a language and country code (for example, “fr_CA”).</p> <p>This parameter is optional. If you do not specify a value, the system default is used.</p> <p>Case is significant. Language codes are lowercase, country codes uppercase. locale is a string (see <a href="#">xs:string on page 165</a>).</p>
<b>x-sirs-sessionToken</b>	<p>The sessionToken identifies an active Web Services session for a specific user with specific privileges and (optionally) a specific station policy (workstationID).</p> <p>A valid session token must be included in the message header for all privileged operations (see <a href="#">Privileged operations on page 19</a>).</p> <p>A Web Services client obtains a session token by using the loginUser operation (see <a href="#">loginUser on page 71</a>).</p> <p>You end a session using logoutUser (see <a href="#">logoutUser on page 72</a>).</p> <p>sessionToken is a string (see <a href="#">xs:string on page 165</a>).</p>

The following shows a sample REST custom HTTP header:

```
Content-Type: application/x-www-form-urlencoded; charset=utf-8
x-sirs-clientID: SymWSTestClient
x-sirs-locale: en
Content-Length: 24
User-Agent: Java/1.6.0_20
Host: 127.0.0.1:8080
Accept: text/html, image/gif, image/jpeg, *, q=.2, */*; q=.2
Connection: keep-alive
```

The “x-sirs-” prefix is used here to avoid potential naming conflicts in the header.

However, passing these parameters in the header has the following advantages:

- It prevents potentially sensitive information from being stored in logs as part of the URL.
- It allows library system administrators to better secure their applications by restricting access based on specific headers or header values.

You may also use the default HTTP header and send these header values through the browser URL. When doing this, you leave off the “x-sirs-” prefix. See [Testing REST requests with a browser on page 22](#).

Depending on the library configuration, you may also pass in a significant portion of the header key as a parameter of a GET or POST request. For example,

```
GET /symws/rest/standard/lookupTitleInfo?titleID=12225&includeItemInfo=true&clientID=SampleID
HTTP/1.1
```

### Related topics

- [Making REST requests on page 27](#)

## Making a REST version request

The VersionSamples example also shows how to set up the REST request with all the required information. Although you can make this same request through a browser (see [Testing REST requests with a browser on page 22](#)), it is helpful to see how it works through this Perl application.

So, let's go through the important parts of the Perl code.

First, on lines 62-63, you declare the variables used for the HTTP POST command and used for handling the returned data in XML format.

```
my $ua = new LWP::UserAgent;
my $xml = new XML::Simple (KeyAttr=>[]);
```

LWP::UserAgent manages HTTP connections and performs requests for you. The xml variable is set to handle the returned XML, and to assign name elements in order to make it easier to access each node.

In line 66, you make the request.

```
my $response = $ua->post($BASE_URL.'rest/standard/version');
```

Here, you send the specified endpoint using the POST command. With REST, the operation, [version](#), is automatically included with the service. If there are any parameters, you can pass them here as well. For example:

```
my $response = $ua->post($BASE_URL.'rest/standard/searchCatalog',
    {clientId=>$clientId,
      term1=>$searchTerm});
```

This example shows how you can add the parameters to the base endpoint. In making this same request through the browser URL, you append a question mark (?) after the operation in order to indicate the start of the parameters; the parameters are then separated by an ampersand (&). In the Perl application, you can referencing the values of the variables as part of the post function.

For the VersionSamples application, the final step (in line 70) is to print the content.

```
print $response->content."\n";
```

This line just dumps the content to the display as raw XML.

## Making JSON requests

Each JSON (JavaScript Object Notation) request must contain certain information in order for Web Services to accept it. Making a JSON call is almost the same as making a REST call.

You must provide an appropriate endpoint for the request. For details, see [Service endpoints on page 4](#).

You may want to include a custom header in order to avoid passing potentially sensitive information through the URL or that may show up in logs. For information on what can be included, see [Custom HTTP headers on page 27](#).

You can include four additional parameters in a JSON call:

- `json` - If set to "true" it returns the response in a JSON format.
- `prettyprint` - If set to "true" it indents and formats the response for easy readability. This should only be used for testing purposes.
- `callback` - This is used for cross-domain calls. It is the JavaScript method to pass the results into.
- `jsonPOST` - Use this parameter for cross-domain calls to allow posting the JSON request using a GET call.

JSON requests are almost always HTTP GET calls. Where a REST call requires POST of XML (such as with the [saveMyList](#) operation) the system now allows a GET when you use the `jsonPOST` parameter containing the escaped post data. If you do a POST with the `json` parameter enabled, the system attempts to unmarshal the data as JSON or XML depending on the content-type.

In a response, all inner-elements and attributes of an element are treated as variables in the current variable.

For example, if the response normally returns XML like the following:

```
<parentElement xmlns="some:URI">
  <child1 att="First">
    <subchild1>Content</subchild1>
  </child1>
</parentElement>
```

Then, the JSON response format would be the following:

```
{
  "child1" : {
    "att" : "First",
    "subchild1" : "Content"
  }
}
```

Notice that the element and attribute are marshaled at the same level. Also, JSON does not serialize namespaces or root element names, so the `parentElement` and the URI are not returned.

JSON also requires some unique fault handling. For information, see [Handling faults on page 33](#).

For more information about JSON, see <http://www.json.org>

**Related topics**

- [Custom HTTP headers](#) on page 27
- [Making a REST version request](#) on page 29
- [Working with Web Services](#) on page 17
- [Handling faults](#) on page 33

## Working with Web Services sessions

A Web Services session allows the Web server to keep track of a specific user through a session token. The session token is a unique identifier for an authenticated user in a Web Services session. You start a new session and get a new session token through the [loginUser](#) operation.

For Web Services, some operations only require that you identify a specific user for the session (these are called [Authenticated operations](#)). Authenticated operations only require a valid session token.

Some operations require that you identify a specific user so that the user's access permissions can be verified (these are called [Privileged operations](#)). For example, the server will not allow one user to access another user's personal information without authorization. These privileged operations require a valid session token and additional SirsiDynix Symphony user rights.

A few operations do not require any user identification (such as [license](#) and [version](#)) and therefore do not need a session token.

**What is a session?**

In Web Services, a session caches information for a specific user with specific privileges and (optionally) a specific workstation policy. This improves performance interacting with the ILS server and eliminates the need for you to specify credentials repeatedly for multiple operations.

The Web Services session is independent of any sessions managed by the application server (Tomcat sessions, for example).

**Session usage**

Your application can open and manage multiple sessions simultaneously. You may open more than one session for the same user. You may use the same session token for multiple operations so long as the associated privileges are sufficient for the operations you request.

The user associated with a session token is known as a session user. It is the session user's privileges that determine if you have sufficient permissions to complete a privileged operation. For example, if the session user is a patron, that user will not have permission to work with data that requires staff privileges.

The session user may be different than the user that is the object of a specific request. For example, you might log in a staff user and use the returned session token in one or more **lookupUserInfo** operations to look up information about different patrons—so long as that staff user has sufficient permissions to perform the requested operations.



In general, if the user that is the object of a request is different than the session user, the session user must have a privilege level of STAFF or higher. (For more information about privilege levels, see **PrivilegeLevel** on page 227.)

For more information about specific permissions, refer to the WorkFlows Java client help or consult with a library system administrator.

You end a session with the **logoutUser** operation. You are strongly encouraged to use the **logoutUser** operation to close sessions that are no longer needed; this frees up resources on the server. If you don't log out, the session remains open on the server until it expires.

### A session example

An example will help to illustrate the sequence of calls required to complete privileged or authenticated operations.

Suppose your application provides a way for a patron to display the number of items he or she currently has checked out. Displaying a user's information is a privileged operation. Because privileged operations require a session token, you must create a session.

- 1 Collect the patron's credentials (login and password).
- 2 Submit the login and password in a **loginUser** request to start a new session.
- 3 If the patron is a valid library user, the server returns **LoginUserResponse** with a new **sessionToken** and the patron's **userID**.
- 4 Submit a **lookupMyAccountInfo** request using the **sessionToken**, with **includePatronInfo** and **includePatronCheckoutInfo** set to true.
- 5 The server returns **LookupMyAccountInfoResponse** with information about the patron, including the number of items currently checked out.
- 6 At this point, you could perform other operations with the same **sessionToken** or submit a **logoutUser** request to terminate the session.



**Related topics**

- [Working with Web Services](#) on page 17
- [Authenticated operations](#) on page 18
- [Privileged operations](#) on page 19

## Handling faults

Web Services sets HTTP status for all responses. Successful operations return HTTP 200. If the operation could not be completed for some reason, Web Services will set HTTP status to something other than 200, typically HTTP 500 (Internal Server Error). (You might also see HTTP 404, but this is an error from the Tomcat server indicating a problem with the URL, not the operation.)

It is good practice for your application to check HTTP status for all responses. If HTTP status is something other than 200, your application can process the content of the response for specific fault details if any.

For example, if you attempt a `loginUser` request with bad credentials, you will get a response like this:

**SOAP example**

```
HTTP/1.1 500 Internal Server Error
Server: Apache-Coyote/1.1
Content-Type: text/xml; charset=utf-8
Transfer-Encoding: chunked
Date: Wed, 29 Sep 2010 20:02:37 GMT
Connection: close
```

```
<?xml version="1.0" ?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><S:Fault
xmlns:ns3="http://www.w3.org/2003/05/soap-envelope"><faultcode
xmlns:ns0="http://schemas.sirsidynix.com/symws/exception">ns0:com.sirsidynix.symws.service.exceptions
to log in.</faultstring></S:Fault></S:Body></S:Envelope>
```

**REST example**

```
HTTP/1.1 500 Internal Server Error
Server: Apache-Coyote/1.1
Content-Type: text/xml; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 29 Sep 2010 20:06:01 GMT
Connection: close
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><Fault
xmlns="http://schemas.sirsidynix.com/rest"><code>com.sirsidynix.symws.service.exceptions.SecurityServ
to log in.</string></Fault>
```

By catching HTTP 500, you can then extract the fault code and string and take some action or display the fault string to the client user.

For information about specific faults, see [Faults](#) on page 235.

## Handling JSON Faults

JSON faults need to be handled in a slightly different manner because faults for JSON are always returned with an OK HTTP 200.

An example fault response might look like the following:

```
{
    faultResponse : {
        "code" :
        "com.sirsidynix.symws.protocol.exceptions.HeaderException.badClientID",
        "string" : "Bad Client ID: SymWSTestClients"
    }
}
```

So, you should check to see if `<responseVar>.faultResponse` is an object type. If it is, then `<responseVar>.faultResponse.code` contains the error code and `<responseVar>.faultResponse.string` contains the error message.

If it is not an object type, then there is no fault and you can continue with the script.

### Related topics

- [Working with Web Services on page 17](#)
- [Debugging and Fault Checking with SOAP in Perl on page 34](#)
- [Fault Checking with REST in Perl on page 35](#)

## Debugging and Fault Checking with SOAP in Perl

For SOAP, you can use `SOAP::Lite` to help you debug in Perl.

For example, in the `SearchSamples` Perl code, the following lines (16-17) provide an easy way to turn on custom debugging:

```
GetOptions ('debug' => \my $DEBUG);
SOAP::Lite->import(trace => 'debug') if $DEBUG;
```

Then, the code does some error checking as part of the `searchCatalogSoap` subroutine.

```
if ($response->fault)
{
    print $response->faultcode, ' ', $response->faultstring, '\n';
    die 'Terminating program';
}
```

This looks for any faults returned in the response and prints out the error code and the error description. Then, the program execution is halted with the `die` command.

**Related topics**

- [Working with Web Services](#) on page 17
- [Handling faults](#) on page 33
- [Fault Checking with REST in Perl](#) on page 35

## Fault Checking with REST in Perl

For REST, it is a good idea to account for failures in the responses.

For example, in the SearchSamples Perl code, line 191 checks for this condition:

```
die $response->message . "\n" . $response->content unless $response->is_success;
```

This line stops the execution of the program if the response does not return with `is_success`. If there is a failure, the message title and the error message display for the user.

**Related topics**

- [Working with Web Services](#) on page 17
- [Handling faults](#) on page 33
- [Debugging and Fault Checking with SOAP in Perl](#) on page 34

## Searching the Catalog using Perl

One of the example applications provided for Perl is SearchingSamples. This application shows examples of [searchCatalog](#), [searchCatalogPaging](#), and [lookupTitleInfo](#) calls for Symphony Web Services.

We'll go through parts of the code for REST ([Searching the Catalog using REST](#) on page 35). For SOAP, refer to the comments in the example application.

### Searching the Catalog using REST

The first REST subroutine in the SearchingSamples code is `searchCatalogRest`. This performs the [searchCatalog](#) operation (using the value assigned to the `searchTerm` variable) and returns the query ID and hits.

In lines 187-189, we make the request.

```
my $response = $ua->post($BASE_URL.'rest/standard/searchCatalog',
    {clientID=>$clientID,
      term1=>$searchTerm});
```

In these lines, we specify the endpoint for the `searchCatalog` request. The [searchCatalog](#) operation is part of the standard service. We also pass in some required data: the `clientID` and the first search term. The results of this call get returned to the response variable.

Also, notice that the parameter “term” has had a number appended to it. This is allowed (and expected) for parameters where more than one are allowed in a request. Check the details of each parameter in the operation to learn more about the cardinality. So, if you wanted to pass in two terms, the first parameter key would be “term1” and the second parameter key would be “term2.”

Next, we convert the result to something Perl can handle.

```
my $data = $xml->XMLin($response->content);
```

XMLin helps us to convert the XML data to an internal Perl representation. You can use the following call to see the internal representation of the XML data:

```
print Dumper($data);
```

The final part of this subroutine returns the queryID:

```
return ($data->{queryID});
```

The next subroutine called is `searchCatalogPagingRest`. The queryID is passed in and it returns titleID. This subroutine only displays hits 21 through 40 of the previously done query.

In lines 210-214, we make the request.

```
my $response = $ua->post($BASE_URL.'rest/standard/searchCatalogPaging',
    {clientID=>$clientID,
      queryID=>$queryID,
      firstHitToDisplay=>'21',
      lastHitToDisplay=>'40'});
```

This call uses the passed in queryID to identify the results list from the previous `searchCatalog` operation. The `searchCatalogPaging` operation allows you to specify beginning and ending ranges of results to display. This is useful for when you only want to display a limited number of results or when you want to allow the user to specify how many results to display.

After converting the XML to a Perl representation, on line 222 the application returns the titleID of the 21st result.

```
return $data->{HitlistTitleInfo}[0]->{titleID};
```

The XML result for the paged search returns a `HitlistTitleInfo` node for each result. Within that node, there is a `titleID` node. This line gets the first element of `HitlistTitleInfo`, which is `titleID`, and returns it as a scalar variable.

The last subroutine, `lookupTitleInfoRest`, is passed the titleID returned from the previous subroutine. This subroutine retrieves the item information for that titleID.

In lines 235-240 of the code, we make the request:

```
my $response = $ua->post($BASE_URL.'rest/standard/lookupTitleInfo',
    {clientID=>$clientID,
      titleID=>$titleID,
      includeAvailabilityInfo=>'true',
      includeItemInfo=>'true',
      marcEntryFilter=>'NONE'});
```

The **lookupTitleInfo** operation is part of the standard service, so we use that for the endpoint. The passed-in titleID is used to identify the requested item. We also have specified that we want to include the availability information (availability would need to be turned on for this to work) and item information, but not MARC data.

The application then displays the returned information.

See the SearchingSamples application for the complete code to achieve this type of functionality.

## Interacting with patron data using Perl

One of the example applications provided for Perl is Patron Samples. This application shows examples of **loginUser**, **logoutUser**, **lookupMyAccountInfo**, and **renewMyCheckout** calls for Symphony Web Services.

We'll go through parts of the code for REST (**Interacting with patron data using REST on page 37**). For SOAP, refer to the comments in the example application.

### Interacting with patron data using REST

For the PatronSamples application, you need to provide values for some of the variables. In addition to the BASE\_URL, you need to specify a valid userID, password, and userLibrary for a patron account on your Symphony system; a couple valid itemIDs - one for an item that the specific patron already has checked out and can renew and one for an item that the specific patron has authority and capability to place a hold on; and the web services clientID.

The first REST subroutine logs in the specified user and returns a session token. On lines 252-253, we set up the variables used for the HTTP POST command and for handling the returned data in XML format. On line 255, we set the application to be read for the UTF-8 encoded data that gets returned.

Then, in lines 256-258, we set up the request.

```
my $response = $ua->post($BASE_URL.'rest/security/loginUser',
    {clientID=>$clientID,
      login=>$userID, password=>$password});
```

This call for **loginUser** uses the login information specified in variables. Typically, you would get this information from the user through a web page form or other type of input to the application.

After checking for success to the request and converting the XML data to an internal representation for Perl, the subroutine returns the session token value so that other privileged operations can use it:

```
return $data->{sessionToken};
```

This session token is then used for the next subroutine, **lookupMyAccountInfoRest**, which returns the itemID of an item that the user has checked out.

On lines 279-281, we make the request:

```
my $response = $ua->post($BASE_URL.'rest/patron/lookupMyAccountInfo',
    {clientId=>$clientId, sessionToken=>$sessionToken,
    includePatronCheckoutInfo=>'ALL'});
```

The **lookupMyAccountInfo** request is part of the patron service, as indicated in the endpoint address. It also requires a client ID and session token, which have been passed through as variables. We have decided to get all the patron checkout information for this request. The `lookupMyAccountInfo` operation can retrieve lots of different kinds of data about the patron account. Since we only care about getting the checkout data for this application, restricting the result to one area is more efficient.

After checking the response and converting the XML, line 289 returns the `itemID` of a checked out item.

```
return $data->{patronCheckoutInfo}->{itemID};
```

This code assumes that there is at least one checked out item. This line finds the value from the `ItemID` node and assigns it to the returned data.

The next subroutine then takes the session token and the item ID and renews the item. This code assumes that the item can be renewed by this patron account. Lines 303-305 perform the request.

```
my $response = $ua->post($BASE_URL.'rest/patron/renewMyCheckout',
    {clientId=>$clientId, sessionToken=>$sessionToken,
    itemID=>$checkoutItemID});
```

The **renewMyCheckout** operation takes either a `titleID` or `itemID`.

The subroutine checks for a success message in the response and then prints the data from the response.

The next subroutine, `createMyHoldRequest`, is passed in the session token so that the patron can create a hold request on an item.

On lines 321-323, we make the hold request.

```
my $response = $ua->post($BASE_URL.'rest/patron/createMyHold',
    {clientId=>$clientId, sessionToken=>$sessionToken,
    itemID=>$hold2ItemID, pickupLibraryID=>$userLibrary});
```

The **createMyHold** operation requires a valid client ID and session token, as well as an item ID or title ID and a pickup location.

After checking for success in the response, the subroutine prints the returned `HoldKey`.

Finally, the last subroutine, `logoutRest`, uses the session token to log out the patron.

The request is made in lines 340-341.

```
my $response = $ua->post($BASE_URL.'rest/security/logoutUser',
    {clientId=>$clientId, sessionToken=>$sessionToken});
```

The **logoutUser** operation requires a valid session token and client ID.

The subroutine completes by displaying the response data.

See the `SearchingSamples` application for the complete code to achieve all this patron functionality.





# Chapter 4: Services and Operations

This section describes the services and operations available in SirsiDynix Symphony Web Services.

Each topic will contain a synopsis about the operation. The synopsis explains the operation namespace and defines any prerequisite information needed to execute the operation. Each topic also outlines information about request parameters and describes what elements will be returned in the operation response.

For each operation request, you will see the following information in a table:

Heading	Description
Parameter	Lists all the request parameters for the operation.
Type	Lists the data type of each request parameter for the operation. These are links to topics that describe the data type in detail (for more information, see <a href="#">Chapter Chapter 5, "Data types reference" on page 163</a> ).
Description	Offers a brief description for each request parameter in the operation.
Required?	Specifies whether or not each request parameter is required for the operation request. <b>Y</b> means the request parameter is required; <b>N</b> means the request parameter is optional.

For each operation response, you will see the following information in a table:

Heading	Description
Element	Lists all the response elements for the operation.
Type	Lists the data type of each response element for the operation. These are links to topics that describe the data type in detail (for more information, see <a href="#">Chapter Chapter 5, "Data types reference" on page 163</a> ).
Occ.	Specifies the minimum and maximum number of times the element can occur (or be returned) in the operation response. This is the format: <b>&lt;minimum&gt;..&lt;maximum&gt;</b> . For example, if the element occurred a minimum of 0 and was returned a maximum of 1, you would see "0..1". If the element always occurs once, but only once, you will see "1".
Description	Offers a brief description for each response element in the operation.

Some request parameters and element responses are a choice pair. This means that in a request, you will need to specify only one of the parameters (not both), and in a response, only one element will be returned (not both). Choices are specified for your convenience.

This section contains these topics:

- [Standard service on page 42](#)
- [Security service on page 69](#)
- [Patron service on page 77](#)
- [Request service on page 106](#)
- [Admin service on page 116](#)

## Standard service

The Standard service provides operations for common library features that do not require authentication or special privileges.

The Standard service includes these operations (listed alphabetically):

Operation	Description
<a href="#">license</a>	Displays license data for the Web Services instance.
<a href="#">lookupBestSellersDisplayList</a>	Returns best seller display lists.
<a href="#">lookupBestSellerTitles</a>	Returns a list of best seller titles.
<a href="#">lookupHotSites</a>	Returns a list of hot sites.
<a href="#">lookupMostPopular</a>	Returns a list of popular titles, authors, or subjects (listType). Most Popular lists are based on circulation statistics.
<a href="#">LookupRecommendedReadingDisplayList</a>	Returns recommended reading display lists.
<a href="#">lookupRecommendedReadingTitles</a>	Returns a list of Recommended Reading titles.
<a href="#">lookupTitleInfo</a>	Displays information about specific titles, such as item availability and, optionally, other item details.
<a href="#">searchCatalog</a>	Lets you search the ILS catalog based on terms and indexes (searchType) that you specify. You may optionally request that the hit list include availability info. Use searchCatalogPaging to page through results.  This is an authenticated request (it accepts but does not require a session token). Data returned may vary depending on session user.

Operation	Description
<a href="#">searchCatalogPaging</a>	Lets you page through results from a specified query. This is an authenticated request (it accepts but does not require a session token). Data returned may vary depending on session user.
<a href="#">searchInfoDesk</a>	Lets you search the ILS catalog for records that match a specified Information Desk policy. User <a href="#">searchCatalogPaging</a> to page through results as needed.
<a href="#">searchInfoDeskPaging</a>	Lets you page through results from a specified query of an Information Desk policy. This is an authenticated request (it accepts but does not require a session token). Data returned may vary depending on session user.
<a href="#">version</a>	Displays the version of Web Services and (if Web Services is currently connected to the ILS) the ILS software.

This section also includes elements that are referenced in these Standard service operation responses (see [Referenced elements](#)).

### Related topics

- [Referenced elements](#)
- [Services and Operations](#) on page 41

## license

The license operation allows you to retrieve the Web Service license level information.

### Synopsis

Service	<a href="#">Standard service</a>
Namespace	<a href="#">http://schemas.sirsiidynix.com/symws/standard</a>
Prerequisites	None
Request	<a href="#">LicenseRequest</a>
Response	<a href="#">LicenseResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LicenseRequest

The LicenseRequest has no request parameters.

### LicenseResponse

The LicenseResponse returns the following elements:

Element	Type	Occ.	Description
attribute	None	0..*	This element is a container that separates key and value element responses in the response output.
key	<a href="#">xs:string</a>	1	Displays the license key.
value	<a href="#">xs:string</a>	1	Displays the license value.

### Related topics

- [Standard service](#)

## lookupBestSellersDisplayList

The lookupBestSellersDisplayList operation allows you to retrieve the best seller lists and a description about each one.

### Synopsis

Service	<a href="#">Standard service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/standard">http://schemas.sirsidynix.com/symws/standard</a>
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">lookupBestSellersDisplayList</a>
Response	<a href="#">lookupBestSellersDisplayList</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupBestSellersDisplayListRequest

The LookupBestSellersDisplayListRequest has the following request parameter:

Parameter	Type	Description	Required?
bestSellerID	<a href="#">xs:string</a>	Specifies the policy ID of a Best Seller Policy.	Y

### LookupBestSellersDisplayListResponse

The LookupBestSellersDisplayListResponse returns the following element:

Element	Type	Occ.	Description
bestSellerDisplayInfo	<b>BestSellerDisplayInfo</b>	0..*	Displays information about each Best Seller list in the policy.

#### Related topics

- [Standard service](#)
- [lookupBestSellersPolicyList](#)
- [getBestSellerPolicy](#)

## lookupBestSellerTitles

The lookupBestSellerTitles operation allows you to retrieve the titles from a specified best seller display list.

#### Synopsis

Service	<a href="#">Standard service</a>
Namespace	http://schemas.sirsidynix.com/symws/standard
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">lookupBestSellerTitles</a>
Response	<a href="#">lookupBestSellerTitles</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### LookupBestSellerTitlesRequest

The LookupBestSellerTitlesRequest has the following request parameter:

Parameter	Type	Description	Required?
bestSellerID	<b>xs:string</b>	Specifies the policy ID of a Best Seller Policy.	Y

#### LookupBestSellerTitlesResponse

The LookupBestSellerTitlesResponse returns the following element:

Element	Type	Occ.	Description
bestSellerTitleInfo	<b>BestSellerTitleInfo</b>	0..*	Displays information about each title in the Best Seller list.

**Related topics**

- [Standard service](#)
- [lookupBestSellersDisplayList](#)

**lookupHotSites**

The lookupHotSites operation allows you to retrieve the list of hot sites and URLs.

**Synopsis**

<b>Service</b>	<a href="#">Standard service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/standard
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupHotSitesRequest</a>
<b>Response</b>	<a href="#">LookupHotSitesResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**LookupHotSitesRequest**

The LookupHotSitesRequest has no request parameters.

**LookupHotSitesResponse**

The LookupHotSitesResponse returns the following element:

Element	Type	Occ.	Description
hotSitesInfo	<a href="#">HotSitesInfo</a>	0..*	Displays information about each hot site.

**Related topics**

- [Standard service](#)
- [lookupHotSites](#)

**lookupMostPopular**

The lookupMostPopular operation allows you to retrieve a list of the most popular authors, subjects, or titles that are checked out from the library.

**Synopsis**

Service	<b>Standard service</b>
Namespace	http://schemas.sirsidynix.com/symws/standard
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <b>clientID</b>)</li> </ul>
Request	<b>LookupMostPopularRequest</b>
Response	<b>LookupMostPopularResponse</b>
Operation faults	None (see also <b>Faults</b> )

### LookupMostPopularRequest

The LookupMostPopularRequest has the following request parameters:

Parameter	Type	Description	Required?
listType	<b>xs:string</b>	Specifies one of the following enumeration values: <ul style="list-style-type: none"> <li><b>AUTHOR</b>: specifies you want the most popular authors to display</li> <li><b>SUBJECT</b>: specifies you want the most popular subjects to display</li> <li><b>TITLE</b>: specifies you want the most popular titles to display</li> </ul>	Y
libraryFilter	<b>xs:string</b>	Specifies an optional Library ID or Library Group ID.	N
sourceFile	<b>xs:string</b>	Specifies an optional alternate file name for source file.	N
linesToDisplay	<b>xs:nonNegativeInteger</b>	Specifies the maximum number of authors, subjects, or titles to display in the response.	N

### LookupMostPopularResponse

The LookupMostPopularResponse returns the following elements:

Element	Type	Occ.	Description
linesDisplayed	<b>xs:int</b>	1	Displays the number of lines that were returned in the response.

This response also references the following elements:

Element	Occ.	Description
<b>MostPopularInfo</b>	0..*	Displays most popular information about the listType you specify in the request.

**Related topics**

- [Standard service](#)
- [MostPopularInfo](#)

**LookupRecommendedReadingDisplayList**

The lookupRecommendedReadingDisplayList operation allows you to retrieve the Recommended Reading lists and a description about each one.

**Synopsis**

<b>Service</b>	<a href="#">Standard service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/standard
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupRecommendedReadingDisplayList</a>
<b>Response</b>	<a href="#">LookupRecommendedReadingDisplayList</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**LookupRecommendedReadingDisplayListRequest**

The LookupRecommendedReadingDisplayListRequest has the following request parameter:

Parameter	Type	Description	Required?
recommendedReadingID	<a href="#">xs:string</a>	Specifies the policy ID of a Recommended Reading Policy.	Y

**LookupRecommendedReadingDisplayListResponse**

The LookupRecommendedReadingDisplayListResponse returns the following element:

Element	Type	Occ.	Description
recommendedReadingDisplayInfo	<a href="#">RecommendedReadingDisplayInfo</a>	0..*	Displays information about each Recommended Reading list in the policy.



**Related topics**

- [Standard service](#)
- [lookupRecommendedReadingPolicyList](#)
- [getRecommendedReadingPolicy](#)

**lookupRecommendedReadingTitles**

The lookupRecommendedReadingTitles operation allows you to retrieve the titles from a specified Recommended Reading display list.

**Synopsis**

Service	<a href="#">Standard service</a>
Namespace	http://schemas.sirsidynix.com/symws/standard
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupRecommendedReadingTitlesRequest</a>
Response	<a href="#">LookupRecommendedReadingTitlesResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

**LookupRecommendedReadingTitlesRequest**

The LookupRecommendedReadingTitlesRequest has the following request parameter:

Parameter	Type	Description	Required?
recommendedReadingID	<a href="#">xs:string</a>	Specifies the policy ID of a Recommended Reading Policy.	Y

**LookupRecommendedReadingTitlesResponse**

The LookupRecommendedReadingTitlesResponse returns the following element:

Element	Type	Occ.	Description
recommendedReadingTitleInfo	<a href="#">RecommendedReadingTitleInfo</a>	0..*	Displays information about each title in the Best Seller list.

**Related topics**

- [Standard service](#)
- [LookupRecommendedReadingDisplayList](#)

## lookupTitleInfo

The lookupTitleInfo operation allows you to retrieve information about the number of items that are available to be checked out for titles you specify. Optionally, you can also retrieve information about a title's items (such as owning library, call number, number of copies, current location, due dates, and so on).

**Synopsis**

Service	<a href="#">Standard service</a>
Namespace	http://schemas.sirsiidynix.com/symws/standard
Prerequisites	<ul style="list-style-type: none"><li>• Valid clientID (see <a href="#">clientID</a>)</li></ul>
Request	<a href="#">LookupTitleInfoRequest</a>
Response	<a href="#">LookupTitleInfoResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

**LookupTitleInfoRequest**

Parameter	Type	Description	Required?
libraryFilter	xs:string	<p>Specifies a library ID or Search Library group ID to use in order to limit returned data to a specific library or group of libraries. When you include libraryFilter in the request, the response includes information only for items owned by the specified library or group of libraries.</p> <p>Values for this element are specific to the ILS system.</p> <p>This element is optional. If you do not specify a library filter, information will be returned for all libraries in the system that own the specified title.</p>	N
titleID and itemID are a choice. Specify a value for one or the other.			
titleID	xs:long	<p>Specifies the catalog key for a specific title in the library catalog. You may include more than one titleID in a request. At least one titleID (or itemID) is required.</p> <p><b>Note:</b> You only need to specify a titleID or an itemID. If you do not specify an itemID, you must specify a titleID.</p>	Y
itemID	xs:string	<p>Specifies the item ID of the specific title. You man include more than one itemID in a request. At least one itemID (or titleID) is required.</p> <p><b>Note:</b> You only need to specify an itemID or a titleID.If you do not specify a titleID you must specify an itemID.</p>	
All the “include” parameters are assumed <b>false</b> by default if you do not specify a value.			
includeAvailabilityInfo	xs:boolean	Specifies whether you want to include availability information for each specified title in the response ( <b>true</b> ) or not ( <b>false</b> ).	N
includeItemInfo	xs:boolean	Specifies if you want to include item information for each specified title in the response ( <b>true</b> ) or not ( <b>false</b> ).	N
includeCatalogingInfo	xs:boolean	Specifies if you want to include cataloging information for each specified title in the response ( <b>true</b> ) or not ( <b>false</b> ).	N
includeOrderInfo	xs:boolean	Specifies if you want to include order information for each specified title in the response ( <b>true</b> ) or not ( <b>false</b> ).	N
includeOPACInfo	xs:boolean	Specifies if you want to include OPAC information for each specified title in the response ( <b>true</b> ) or not ( <b>false</b> ).	N
includeBoundTogether	xs:boolean	Specifies if you want to include items that are bound together in the response ( <b>true</b> ) or not ( <b>false</b> ).	N
includeMarcHoldings	xs:boolean	Specifies if you want to include MARC holdings in the response ( <b>true</b> ) or not ( <b>false</b> ).	N

Parameter	Type	Description	Required?
marcEntryFilter	<b>xs:string</b>	Specifies one of the following enumeration values: <ul style="list-style-type: none"> <li>• <b>ALL</b></li> <li>• <b>BRIEF</b></li> <li>• <b>FULL</b></li> <li>• <b>NONE</b></li> <li>• <b>TEMPLATE</b></li> </ul> <b>Note:</b> If you do not specify a marcEntryFilter value, “NONE” is assumed by default.	N
marcEntryID	<b>xs:string</b>	Specifies the title MARC entry ID.	N
includeRelatedSearchInfo	<b>xs:boolean</b>	Specifies if you want to include related search information ( <b>true</b> ) or not ( <b>false</b> ).	N
includeCallNumberSummary	<b>xs:boolean</b>	Specifies if you want to include call number information ( <b>true</b> ) or not ( <b>false</b> ). This is useful for determining whether a volume-level hold request must be made on the title. For more information, see <a href="#">Volume holds on page 12</a> .	N

### LookupTitleInfoResponse

The LookupTitleInfoResponse references the following elements:

Element	Occ.	Description
<b>TitleInfo</b>	1..*	Displays title information for each title you specify in the request.

The response will include one titleInfo object for each titleID you specify.

The information that returns in LookupTitleInfoResponse may be limited to a specific library or group of libraries. For example, if you specified EAST as a libraryFilter in the request, LookupTitleInfoResponse will only display information and counts for items in the EAST library.

If the library filter or title ID specified in the request is invalid, or if there are no items for a title (for example, because it is on order and the library has not yet received copies), availability counts will be zero and there will be no callInfo.

### Related topics

- [Standard service](#)
- [TitleInfo](#)

## searchCatalog

The searchCatalog operation allows you to execute a keyword search of the library catalog.

### Synopsis

Service	<b>Standard service</b>
Namespace	http://schemas.sirsidynix.com/symws/standard
Prerequisites	<ul style="list-style-type: none"><li>Valid clientID (see <b>clientID</b>)</li></ul>
Request	<b>SearchCatalogRequest</b>
Response	<b>SearchCatalogResponse</b>
Operation faults	None (see also <b>Faults</b> )

### **SearchCatalogRequest**

The SearchCatalogRequest has the following request parameters:

Parameter	Type	Description	Required?
query	None	The query parameter does not need a value. It is a container for different term, searchType, and operator parameters in a request. You can have up to 8 query entries in one request.	
term	<b>xs:string</b>	Specifies your search term. You can have up to 1024 characters in the term parameter value. <b>Important:</b> In REST requests, you must number the term within a query (for example, "term1=<value>").	Y
searchType	<b>xs:string</b>	Specifies the search policy type. If you do not specify a value, the search index will default to "GENERAL" by default. <b>Important:</b> In REST requests, you must number the searchTypes within a query (for example, "searchType1=<value>").	N
operator	<b>xs:string</b>	Specifies one of the following enumeration values: <ul style="list-style-type: none"> <li>• <b>AND</b></li> <li>• <b>OR</b></li> <li>• <b>NOT</b></li> <li>• <b>XOR</b></li> </ul> If you do not specify a value, the system will assume "AND" by default. <b>Important:</b> In REST requests, you must number your search terms within a query (for example, "operator1=<value>").	N
exactMatch	<b>xs:boolean</b>	Specifies if you want searches to return exact results ( <b>true</b> ), or not ( <b>false</b> ). If you do not specify a value, the system will assume "false" by default.	N
hitsToDisplay	<b>xs:nonNegativeInteger</b>	Specifies the maximum amount of search results you want to return in the response. If you do not specify a value, the system will default to 20 results.	N
filters	<b>SearchCatalogFilters</b>	Specifies any filters you want to apply to your catalog search.	N
includeAvailabilityInfo	<b>xs:boolean</b>	Specifies if you want to include availability information for items returned in the response ( <b>true</b> ), or not ( <b>false</b> ).	N

### SearchCatalogResponse

The `SearchCatalogResponse` returns a `SearchCatalogResult` data type (see [SearchCatalogResult](#)).

#### Related topics

- [Standard service](#)
- [SearchCatalogFilters](#)
- [SearchCatalogResult](#)

## searchCatalogPaging

The `searchCatalogPaging` operation allows you to page through a `searchCatalog` hitlist (or search results list) using a specified beginning and ending range within the specified number of found results.

#### Synopsis

Service	<a href="#">Standard service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/standard</code>
Prerequisites	<ul style="list-style-type: none"> <li>• Valid <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">SearchCatalogPagingRequest</a>
Response	<a href="#">SearchCatalogPagingResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### SearchCatalogPagingRequest

The `SearchCatalogPagingRequest` is a `SearchCatalogPaging` (see [SearchCatalogPaging](#)).

#### SearchCatalogPagingResponse

The `SearchCatalogPagingResponse` returns a `SearchCatalogResult` data type (see [SearchCatalogResult](#)).



If you set `includeAvailabilityInfo` to “true”, then `TitleAvailabilityInfo` will be included for each title on the hitlist. (For more information, see [TitleAvailabilityInfo](#).)

**Related topics**

- [Standard service](#)
- [SearchCatalogResult](#)
- [TitleAvailabilityInfo](#)
- [SearchCatalogResponse](#)

**searchInfoDesk**

The searchInfoDesk operation allows you to execute a keyword search of the library catalog using an Info Desk lookup.

**Synopsis**

<b>Service</b>	<a href="#">Standard service</a>
<b>Namespace</b>	http://schemas.sirsiidynix.com/symws/standard
<b>Prerequisites</b>	<ul style="list-style-type: none"><li>• Valid clientID (see <a href="#">clientID</a>)</li></ul>
<b>Request</b>	<a href="#">SearchCatalogRequest</a>
<b>Response</b>	<a href="#">SearchCatalogResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**SearchInfoDeskRequest**

The SearchInfoDeskRequest has the following request parameters:



Parameter	Type	Description	Required?
<b>infoDesk</b> and <b>customInfoDesk</b> are a choice. Specify a value for one or the other.			
infoDesk	<b>InfoDesk</b>	Specifies the enumerated info desk value. <b>Note:</b> You only need to specify an infoDesk or a customInfoDesk value. If you do not specify an infoDesk value, you must specify a customInfoDesk value.	Y
customInfoDesk	<b>xs:string</b>	Specifies any custom info desk value for a search that was previously created and saved in Symphony. <b>Note:</b> You only need to specify a customInfoDesk or an infoDesk value. If you do not specify a customInfoDesk value, you must specify an infoDesk value.	
hitsToDisplay	<b>xs:nonNegativeInteger</b>	Specifies the maximum number of results to return in the response.	N
filters	<b>SearchCatalogFilters</b>	Specifies any filters you want to apply to your catalog search.	N
includeAvailabilityInfo	<b>xs:boolean</b>	Specifies if you want to include title availability information (see <b>TitleAvailabilityInfo</b> ) in the response ( <b>true</b> ), or not ( <b>false</b> ). <b>Note:</b> If you do not specify a value, the system will assume “false” by default.	N

### SearchInfoDeskResponse

The SearchInfoDeskResponse returns a SearchCatalogResult data type (see **SearchCatalogResult**).



If you set includeAvailabilityInfo to “true,” then TitleAvailabilityInfo will be included for each title on the hitlist. (For more information, see **TitleAvailabilityInfo**.)

### Related topics

- **Standard service**
- **InfoDesk**
- **SearchCatalogFilters**
- **TitleAvailabilityInfo**

### searchInfoDeskPaging

The searchCatalogPaging operation allows you to page through a searchInfoDesk hitlist (or search results list) using a specified beginning and ending range within the specified number of found results.

### Synopsis

<b>Service</b>	<b>Standard service</b>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/standard
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">searchInfoDeskPaging</a>
<b>Response</b>	<a href="#">searchInfoDeskPaging</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

### SearchInfoDeskPagingRequest

The SearchInfoDeskPagingRequest is a SearchCatalogPaging (see [SearchCatalogPaging](#)).

### SearchInfoDeskPagingResponse

The SearchInfoDeskPagingResponse returns a SearchCatalogResult data type (see [SearchCatalogResult](#)).



If you set includeAvailabilityInfo to “true”, then TitleAvailabilityInfo will be included for each title on the hitlist. (For more information, see [TitleAvailabilityInfo](#).)

### Related topics

- [Standard service](#)
- [TitleAvailabilityInfo](#)
- [SearchCatalogPaging](#) on page 233
- [searchInfoDesk](#) on page 56

## version

The version operation allows you to retrieve version information of Web Services.

### Synopsis

<b>Service</b>	<b>Standard service</b>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/standard
<b>Prerequisites</b>	None
<b>Request</b>	<a href="#">VersionRequest</a>
<b>Response</b>	<a href="#">VersionResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

### VersionRequest

The VersionRequest has no request parameters.

### VersionResponse

The VersionResponse returns the following elements:

Element	Type	Occ.	Description
version	None	0..*	This element is a container that separates product and version element responses in the response output.
product	<a href="#">xs:string</a>	1	Displays the SirsiDynix product name for this version (for example, “SYMPHONY” or “SYM-WS”).
version	<a href="#">xs:string</a>	1	Displays the version of the product (for example, “3.3.1”).

### Related topics

- [Standard service](#)

## Referenced elements

The following elements are referenced in Standard service operation requests and responses:

- [BibliographicInfo](#)
- [BoundwithLinkInfo](#)
- [CallInfo](#)
- [HitlistTitleInfo](#)
- [ItemInfo](#)
- [MarcEntryInfo](#)
- [MarcHoldingsInfo](#)
- [MostPopularInfo](#)
- [TitleAvailabilityInfo](#)
- [TitleInfo](#)
- [TitleOrderInfo](#)

### BibliographicInfo

#### Referenced from

- [TitleInfo](#)

#### Contents

BibliographicInfo references the following elements:

Element	Occ.	Description
<b>MarcEntryInfo</b>	1..*	Displays the item's MARC entry information (information about an individual tag in the MARC record).

## BoundwithLinkInfo

### Referenced from

- **TitleInfo**

### Contents

BoundwithLinkInfo has the following elements:

Element	Type	Occ.	Description
linkedAsParent	<b>xs:boolean</b>	1	Indicates if the item is linked as the parent item ( <b>true</b> ) or as the child item ( <b>false</b> ).
callNumber	<b>xs:string</b>	1	Displays the item's call number.
libraryID	<b>xs:string</b>	1	Displays the Library ID of the item's owning library.
itemID	<b>xs:string</b>	0..1	Displays the item ID. <b>Note:</b> If the linked title has more than one copy, the itemID element displays the itemID of the first copy of the linked callnum.
linkedTitle	<b>BriefTitleInfo</b>	1	Displays brief title information about the linked item.

## CallInfo

### Referenced from

- **TitleInfo**

### Contents

CallInfo has the following elements:

Element	Type	Occ.	Description
libraryID	<a href="#">xs:string</a>	1	Displays the item's Library ID.
classificationID	<a href="#">xs:string</a>	1	Displays the item's classification number.
callNumber	<a href="#">xs:string</a>	1	Displays the item's call number.
numberOfCopies	<a href="#">xs:int</a>	1	Displays the number of copies of the item.
boundParentAuthor	<a href="#">xs:string</a>	0..1	Displays the author of the parent item. The boundParentAuthor element will be included in the response if you set includeBoundTogether to "true" in the request.
boundParentTitle	<a href="#">xs:string</a>	0..1	Displays the title of the parent item. The boundParentTitle element will be included in the response if you set includeBoundTogether to "true" in the request.

CallInfo also references the following elements:

Element	Occ.	Description
<a href="#">ItemInfo</a>	0..*	Displays general information about the item.

## HitlistTitleInfo

### Referenced from

- [SearchCatalogResult](#)

### Contents

In addition to returning BriefTitleInfo (see [BriefTitleInfo](#)), HitlistTitleInfo has the following elements:

Element	Type	Occ.	Description
documentNumber	<a href="#">xs:long</a>	1	Displays the document's entry number.
callNumber	<a href="#">xs:string</a>	1	Displays the title's call number.

## ItemInfo

### Referenced from

- [CallInfo](#)

### Contents

ItemInfo has the following elements:

Element	Type	Occ.	Description
itemID	<b>xs:string</b>	1	Displays the item's item ID.
itemTypeID	<b>xs:string</b>	1	Displays the item's item type ID.
currentLocationID	<b>xs:string</b>	1	Displays the item's current location Library ID.
homeLocationID	<b>xs:string</b>	1	Displays the item's home Library ID.
dueDate	<b>xs:dateTime</b>	0..1	Displays the item's due date (if it is checked out). <b>Note:</b> dueDate is only returned if the item is currently checked out.
recallDueDate	<b>xs:dateTime</b>	0..1	Displays the item's recall due date (if the item has been recalled). <b>Note:</b> recallDueDate is only returned if the item is currently checked out and has been recalled.
reshelvingLocationID	<b>xs:string</b>	0..1	Displays the item's reshelving location ID. <b>Note:</b> reshelvingLocationID is only returned for copies that are considered to be in the process of being reshelved.
"transit" information is returned only for copies that are in transit from one library to another.			
transitSourceLibraryID	<b>xs:string</b>	0..1	Displays the Library ID of the library the item is being transited from.
transitDestinationLibraryID	<b>xs:string</b>	0..1	Displays the Library ID of the library the item is being transited to.
transitReason	<b>xs:string</b>	0..1	Displays the reason the item was put into transit (for example, "HOLD", "RESERVE", etc.).
transitDate	<b>xs:dateTime</b>	0..1	Displays the date the item was put in transit.
chargeable	<b>xs:boolean</b>	1	Indicates if the item is chargeable ( <b>true</b> ) or not ( <b>false</b> ).
numberOfHolds	<b>xs:int</b>	0..1	Displays the number of holds that are placed on the item.
"reserve" information is returned only for items on academic reserve.			
reserveCollectionID	<b>xs:string</b>	0..1	Displays the reserve desk ID associated with the item.
reserveCirculationRule	<b>xs:string</b>	0..1	Displays the Circulation Rule policy associated with reserves for the item.
Booking information is returned only for items available for booking.			

Element	Type	Occ.	Description
mediaDeskID	<b>xs:string</b>	0..1	Displays the media desk ID associated with the item.
fixedTimeBooking	<b>xs:boolean</b>	0..1	Indicates if the item is booked with a fixed time slot ( <b>true</b> ) or not ( <b>false</b> ).
publicNote	<b>xs:string</b>	0..1	Displays the item's public note.
staffNote	<b>xs:string</b>	0..1	Displays the item's staff note.

## MarcEntryInfo

### Referenced from

- **BibliographicInfo**
- **MarcHoldingsInfo**

### Contents

MarcEntryInfo has the following elements:

Element	Type	Occ.	Description
label	<a href="#">xs:string</a>	0..1	Displays the label of the MARC tag (for example, “Personal Author” for the 100 tag).
alternateLabel	<a href="#">xs:string</a>	0..1	Displays the alternate label of the MARC tag. The alternateLabel element is for entries linked to an “alternate graphic representation”.
entryID	<a href="#">xs:string</a>	1	Displays the MARC tag (for example, “100” for the author tag).
indicators	<a href="#">xs:string</a>	1	Displays the indicator of the MARC tag.
text	<a href="#">xs:string</a>	1	Displays the text for the MARC tag (for example, “Author” for the 100 tag).
unformattedText	<a href="#">xs:string</a>	0..1	Displays any unformatted text for the MARC tag. <b>Note:</b> text is returned only if the label variation indicates that it should be included.
entryTypeCodes	<a href="#">xs:string</a>	0..1	Displays the CAT_ENTRY_TYPE policy ID(s) for the entry.
url	<a href="#">xs:string</a>	0..1	Displays a link to a graphic or web URL.
These last three elements are returned only if you set includeRelatedSearchInfo to “true” in the request.			
relatedLookupText	<a href="#">xs:string</a>	0..1	Displays the text of the related tag (for example, “Mark Twain” for AUTHOR).
relatedLookupIndexID	<a href="#">xs:string</a>	0..1	Displays the tag entry ID of the related record (for example, “100” for AUTHOR).
relatedLookupHeadingTypeID	<a href="#">xs:string</a>	0..1	Displays the tag label policy for the related record (for example, “AUTHOR”).

## MarchHoldingsInfo

### Referenced from

- [TitleInfo](#)

### Contents

MarchHoldingsInfo has the following elements:

Element	Type	Occ.	Description
holdingLibraryID	<a href="#">xs:string</a>	1	Displays the Library ID of the library that is holding the item.

MarchHoldingsInfo also references the following elements:



Element	Occ.	Description
<b>MarcEntryInfo</b>	0..*	Displays the item's MARC entry information (information about an individual tag in the MARC record).

## MostPopularInfo

### Referenced from

- **lookupMostPopular**

### Contents

MostPopularInfo has the following elements:

Element	Type	Occ.	Description
searchableHeading	<b>xs:string</b>	1	Displays the author, subject, or title in a normalized form suitable for use as a search term.
displayableHeading	<b>xs:string</b>	1	Displays the author, subject, or title formatted for public display.

## TitleAvailabilityInfo

TitleAvailabilityInfo contains information about a particular item's availability.

### Referenced from

- **BriefTitleInfo**
- **TitleInfo**

### Contents

TitleAvailabilityInfo has the following elements:

Element	Type	Occ.	Description
totalCopiesAvailable	<b>xs:int</b>	1	Displays the number of available copies of the item.
libraryWithAvailableCopies	<b>xs:string</b>	0..*	Displays the library name where copies of the item are available.
totalResvCopiesAvailable	<b>xs:int</b>	1	Displays the number of available reserve copies of the item.
libraryWithAvailableResvCopies	<b>xs:string</b>	0..*	Displays the library name where reserve copies of the item are available.
locationOfFirstAvailableItem	<b>xs:string</b>	0..1	Displays the library name where the first available item is located.
holdable	<b>xs:boolean</b>	1	Indicates if at least one copy is available for holds ( <b>true</b> ), or if no copies are available for holds ( <b>false</b> ).
bookable	<b>xs:boolean</b>	1	Indicates if at least one copy is available for booking ( <b>true</b> ), or if no copies are available for booking ( <b>false</b> ).

## TitleInfo

### Referenced from

- **lookupTitleInfo**

### Contents

TitleInfo has the following elements:

Element	Type	Occ.	Description
titleID	<b>xs:long</b>	1	Displays the item's title ID.
The next four elements will be included if you set includeCatalogingInfo to "true" in the request.			
titleControlNumber	<b>xs:string</b>	0..1	Displays the item's title control number: the primary Symphony ID number for the complete title record.
catalogFormatID	<b>xs:string</b>	0..1	Displays the Catalog Format policy ID (for example, "MARC").
catalogFormatType	<b>xs:string</b>	0..1	Displays a MARC data flag.
materialType	<b>xs:string</b>	0..1	Displays the item's material type.
The next fifteen elements will be included if you set includeOPACInfo to "true" in the request.			
baseCallNumber	<b>xs:string</b>	0..1	Displays the item's base call number (all characters in a call number preceding the analytic (subfield z).
author	<b>xs:string</b>	0..1	Displays the item's author.
title	<b>xs:string</b>	0..1	Displays the item's title.
ISBN	<b>xs:string</b>	0..*	Displays the item's ISBN.
SICI	<b>xs:string</b>	0..*	Displays the item's SICI.
UPC	<b>xs:string</b>	0..*	Displays the item's UPC.
OCLCControlNumber	<b>xs:string</b>	0..1	Displays the item's OCLC control number.
sisacID	<b>xs:string</b>	0..1	Displays the item's SISAC ID.
publisherName	<b>xs:string</b>	0..1	Displays the item's publisher name.
datePublished	<b>xs:string</b>	0..1	Displays the date the item was published.
yearOfPublication	<b>xs:int</b>	0..1	Displays the year the item was published.
extent	<b>xs:string</b>	0..1	Indicates if the hold is a title-level or copy-level hold.
netLibraryID	<b>xs:string</b>	0..1	Displays information from tag 35, subfield "a".
numberOfCallNumbers	<b>xs:int</b>	0..1	Displays the number of call numbers associated with the title.
numberOfTitleHolds	<b>xs:int</b>	0..1	Displays the number of holds are currently on the title.
The next two elements will be included if you set includeOrderInfo to "true" in the request.			

Element	Type	Occ.	Description
copiesOnOrder	<b>xs:int</b>	0..1	Displays the number of copies of the item that are currently on order.
outstandingCopiesOnOrder	<b>xs:int</b>	0..1	Displays the number of copies of the item that are currently on order.
numberOfBoundwithLinks	<b>xs:int</b>	0..1	Displays the number of boundwith links are applied to the item. This element will be included only if you set includeBoundTogether to “true” in the request.
callSummary	<b>CallSummary</b>	0..*	Displays the call number information for the title. This element will be included only if you set includeCallNumberSummary to “true” in the request.

TitleInfo also references the following elements:

Element	Occ.	Description
<b>TitleAvailabilityInfo</b>	0..1	Displays information about the item’s availability. This element will be included only if you set includeAvailabilityInfo to “true” in the request.
<b>TitleOrderInfo</b>	0..*	Displays information about orders on the item. This element will be included only if you set includeOrderInfo to “true” in the request.
<b>CallInfo</b>	0..*	Displays call information about the item. This element will be included only if you set includeItemInfo to “true” in the request.
<b>BibliographicInfo</b>	0..1	Displays the item’s bibliographic information. This element will be included if the marcEntryFilter parameter is set to “NONE” or if you specify a marcEntryID parameter in the request.
<b>MarcHoldingsInfo</b>	0..*	Displays the item’s MARC holdings information. This element will be included only if you set includeMarcHoldings to “true” in the request.
<b>BoundwithLinkInfo</b>	0..*	Displays information about any boundwith links the item may have. This element will be included only if you set includeBoundTogether to “true” in the request.

## TitleOrderInfo

### Referenced from

- **TitleInfo**

### Contents

TitleOrderInfo has the following elements:

Element	Type	Occ.	Description
orderLibraryID	<b>xs:string</b>	1	Displays the Library ID of the ordering library.
copiesOrdered	<b>xs:int</b>	1	Displays the number of copies that were ordered.
volumesOrdered	<b>xs:string</b>	0..1	Displays which volumes of the item were ordered.
orderNote	<b>xs:string</b>	0..1	Displays a note associated with the order.
orderDateReceived	<b>xs:date</b>	0..1	Displays the date the order was received.

## Security service

The Security service provides operations necessary to authenticate library users and initiate user sessions for privileged operations.

The Security service includes these operations (listed alphabetically):

Operation	Description
<b>authenticateUser</b>	Determines if the specified user is valid based on the user's login and password.
<b>getPassthroughToken</b>	SirsiDynix internal use.
<b>loginUser</b>	Creates a new Web Services session for the specified user. Also returns an actual Symphony userID required for all user-specific operations.
<b>logoutUser</b>	Terminates the specified Web Services session.
<b>lookupUserInfo</b>	Displays information for a specified user. The session user must be an accountability operator to request accountability info. This is a privileged request (it requires sessionToken).

This section also includes elements that are referenced in these Security service operation responses (see [Referenced elements](#)).

**Related topics**

- [Services and Operations on page 41](#)

**authenticateUser**

The `authenticateUser` operation allows you to verify that a given user is a valid library user. If LDAP credentials were specified, then `authenticateUser` also verifies those.

You use this operation to confirm that a user is registered with the library and can use library services. (If you need to establish a Web Services session, use the `loginUser` operation instead; see [loginUser on page 71](#).)

By using this operation, you can validate that a user is a registered user without having to create a Web Services session.

**Synopsis**

<b>Service</b>	<a href="#">Security service</a>
<b>Namespace</b>	<a href="#">http://schemas.sirsiidynix.com/symws/security</a>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Valid <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">AuthenticateUserRequest</a>
<b>Response</b>	<a href="#">AuthenticateUserResponse</a>
<b>Operation faults</b>	None (see <a href="#">Faults</a> )

**AuthenticateUserRequest**

The `AuthenticateUserRequest` has the following request parameters:

Parameter	Type	Description	Required?
<code>login</code>	<a href="#">xs:string</a>	Specifies a user in the library system. The value of <code>login</code> is either the user ID (typically a barcode) or an alternate ID if the library chooses to use alternate IDs. It may also be an LDAP UID if LDAP authentication has been enabled.	Y
<code>password</code>	<a href="#">xs:string</a>	Specifies the LDAP password if LDAP authentication has been enabled.	N

**AuthenticateUserResponse**

The content of `AuthenticateUserResponse` is Boolean (see [xs:boolean](#)). The output will be **true** if the user is valid or **false** if not.

**See Also**

- [loginUser](#) on page 71
- [Security service](#) on page 69

## getPassthroughToken

The `getPassthroughToken` operation is intended for SirsiDynix internal use.

### Related topics

- [Security service](#) on page 69

## loginUser

The `loginUser` operation allows you to create a new Web Services session for a specific user restricted to their particular user privileges. When you begin a new session, the software returns a session token. A session token is required for all privileged operations. See [Working with Web Services sessions on page 31](#) for more information.



Special or reserved characters need to be URL encoded when using REST. These kind of characters are especially common in user passwords (such as in an LDAP environment). For example, if a user may use special or reserved characters in a password, such as "h%llo", you would send something like the following login request for a user named JoeB:

```
http://-
/libraryapps.example.org/symws/rest/security/loginUser?clientID=
SymWSTestClient& login= JoeB&password=h%25llo
```

`loginUser` also returns a Symphony userID.

`loginUser` requires a login for a specific user. If the login is incorrect, `loginUser` will fail.

You end a session using the `logoutUser` operation (see [logoutUser on page 72](#)).

### Synopsis

Service	<a href="#">Security service</a>
Namespace	<a href="http://schemas.sirsiidynix.com/symws/security">http://schemas.sirsiidynix.com/symws/security</a>
Prerequisites	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">Faults</a>)</li> </ul>
Request	<a href="#">LoginUserRequest</a>
Response	<a href="#">LoginUserResponse</a>
Operation faults	<a href="#">SecurityServiceException: Unable to log in</a> (see also <a href="#">Faults</a> )

### LoginUserRequest

The `LoginUserRequest` has the following request parameters:

Parameter	Type	Description	Required?
login	<a href="#">xs:string</a>	Specifies a user in the library system. The value of login is either the user ID (typically a barcode) or an alternate ID if the library chooses to use alternate IDs. It may also be an LDAP UID if LDAP authentication has been enabled.	Y
password	<a href="#">xs:string</a>	Specifies the LDAP password if LDAP authentication has been enabled. password values can also be the user PIN if you are not using LDAP.	N
workstationID	<a href="#">xs:string</a>	Specifies the workstationID if you want workstation settings to override user settings (such as station library).	N

### LoginUserResponse

The LoginUserResponse returns the following elements:

Element	Type	Occ.	Description
userID	<a href="#">xs:string</a>	1	<p>Displays a Symphony userID for a user even if the request supplied an alternate ID.</p> <p>The userID element is the primary identifier for a specific user in the ILS system.</p> <p>The userID is typically a barcode, but, depending on the ILS configuration, could be any string.</p> <p>Values for userID are case-sensitive. However, in most configurations of the ILS server, these values are forced to uppercase.</p> <p>userID is a string (see <a href="#">xs:string on page 165</a>).</p>
sessionToken	<a href="#">xs:string</a>	1	<p>Displays the Web Services session token created by the LoginUserRequest for a specific user with specific privileges and (optionally) a specific station library.</p> <p>This sessionToken may be included in the message header for other privileged operations (see <a href="#">Privileged operations on page 19</a>).</p> <p>sessionToken is a string (see <a href="#">xs:string on page 165</a>).</p>

### See Also

- [authenticateUser on page 70](#)
- [logoutUser on page 72](#)
- [Privileged operations on page 19](#)
- [Working with Web Services sessions on page 31](#)
- [Security service on page 69](#)

## logoutUser

The logoutUser operation allows you to terminate (expire) a specific Web Services session. See [Working with Web Services sessions on page 31](#) for more information.



Sessions will eventually expire automatically, but it is good practice to log out when you are finished performing operations for a specific user in order to free system resources.

Once a Web Services session is expired, subsequent requests using the specified sessionToken are rejected.

### Synopsis

Service	<a href="#">Security service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/security">http://schemas.sirsidynix.com/symws/security</a>
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> <li>sessionToken (see <a href="#">sessionToken</a>)</li> </ul>
Request	<a href="#">LogoutUserRequest</a>
Response	None
Operation faults	None (see also <a href="#">Faults</a> )

### LogoutUserRequest

The LogoutUserRequest has no request parameters.

The server determines the session to end from the value of sessionToken.

### Response

logoutUser is a one-way operation; there is no response defined.

### See Also

- [loginUser](#) on page 71
- [Working with Web Services sessions](#) on page 31
- [Security service](#) on page 69

## lookupSessionInfo

Displays user login and host information, similar to what displays in the Java Workflows client when you select Help > Session Info.

The operation is authenticated and requires a valid session token, however the user does not require a minimum privilege level.

### Synopsis

Service	<a href="#">Security service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/security">http://schemas.sirsidynix.com/symws/security</a>

<b>Service</b>	<b>Security service</b>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> <li>sessionToken (see <a href="#">SdHeader</a>)</li> </ul>
<b>Request</b>	<b>LookupUserInfoRequest</b>
<b>Response</b>	<b>LookupSessionInfoResponse</b>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

### LookupSessionInfoRequest

The LookupSessionInfoRequest has no request parameters.

The server determines the session from the value of sessionToken and the user to display data for from the value of clientID.

### LookupSessionInfoResponse

The LookupSessionInfoResponse returns the following elements:

Element	Type	Occ.	Description
userID	<a href="#">xs:string</a>	1	Displays the actual ILS user identifier. userID is typically a barcode, but depending on the ILS configuration, could be any string.
userAccessID	<a href="#">xs:string</a>	1	Displays the ID of the user's assigned access policy. This designates what access rights the user has been granted.
userAccessDescription	<a href="#">xs:string</a>	0..1	Displays the description of the user's assigned access policy.
userEnvironmentID	<a href="#">xs:string</a>	1	Displays the ID of the user's assigned environment policy.
userEnvironmentDescription	<a href="#">xs:string</a>	0..1	Displays the description of the user's assigned environment policy.
userProfileID	<a href="#">xs:string</a>	1	Displays the ID of the user's assigned user profile.
userProfileDescription	<a href="#">xs:string</a>	0..1	Displays the description of the user's assigned user profile.
userLibraryID	<a href="#">xs:string</a>	1	Displays the ID of the user's assigned library policy. This is referred to in WorkFlows as the Station Library.
userLibraryDescription	<a href="#">xs:string</a>	0..1	Displays the description of the user's assigned library policy.
serverIPAddress	<a href="#">xs:string</a>	1	Displays the IP address of the ILS server.
serverPlatform	<a href="#">xs:string</a>	1	Displays the OS version installed on and the database platform used by the ILS server.
serverVersion	<a href="#">xs:string</a>	1	Displays the Symphony ILS version installed on the host server.

**Related topics**

- [Security service](#)

**lookupUserInfo**

The lookupUserInfo operation is used to retrieve user information for a user that you specify.

Users who request another user's information must have a privilege level of STAFF or higher, and will need to have access command code rights to command "Display 2 User." For sites with accountability turned on, only accountable users can lookup other users with clearance settings.

**Synopsis**

<b>Service</b>	<a href="#">Security service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/security
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> <li>• sessionToken (see <a href="#">sessionToken</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupUserInfoRequest</a>
<b>Response</b>	<a href="#">LookupUserInfoResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**LookupUserInfoRequest**

LookupUserInfoRequest has the following request parameters:

Parameter	Type	Description	Required?
userID	<a href="#">xs:string</a>	Specifies which user's information to pull. userID is typically a barcode, but depending on the ILS configuration, could be any string.	Y
includeAccountability	<a href="#">xs:boolean</a>	Indicates if the LookupUserInfoResponse should include accountability information ( <b>true</b> ) or not ( <b>false</b> ). If you do not specify a value, the system assumes "false" by default.	N
includeAllowedCommands	<a href="#">xs:boolean</a>	Indicates if the LookupUserInfoResponse should include a list of allowed commands ( <b>true</b> ) or not ( <b>false</b> ). If you do not specify a value, the system assumes "false" by default.	N

**LookupUserInfoResponse**

The LookupUserInfoResponse returns the following elements:

Element	Type	Occ.	Description
userID	<a href="#">xs:string</a>	1	Displays the actual ILS user identifier. userID is typically a barcode, but depending on the ILS configuration, could be any string.
displayName	<a href="#">xs:string</a>	1	Displays the user's full name. displayName is a string (see <a href="#">xs:string on page 165</a> ).
privilegeLevel	<a href="#">PrivilegeLevel</a>	1	Displays the enumerated value for the user's user level on the Symphony server.
sharedLogin	<a href="#">xs:boolean</a>	1	The ILS server limits operations that can be performed by a user account with access type SHARED. The sharedLogin element specifies if the user account that was used to start a Web Services session is a shared account ( <b>true</b> ) or not ( <b>false</b> ).
validUser	<a href="#">xs:boolean</a>	1	The validUser element will always return <b>true</b> .
libraryID	<a href="#">xs:string</a>	1	Displays the ID of the user's assigned library policy.
accessID	<a href="#">xs:string</a>	1	Displays the ID of the user's assigned access policy. This designates what access rights the user has been granted.
environmentID	<a href="#">xs:string</a>	1	Displays the ID of the user's assigned environment policy.
allowedCommand	<a href="#">xs:string</a>	0..*	If you used includeAllowedCommands in the LookupUserInfoRequest, then a list of all enabled command codes for the user will be returned in the LookupUserInfoResponse.

This response also references the following elements:

Element	Occ.	Description
<a href="#">AccountabilityInfo</a>	0..1	If you used includeAccountability in the LookupUserInfoRequest, then several AccountabilityInfo elements may be returned in the LookupUserInfoResponse.

#### Related topics

- [Security service](#)
- [AccountabilityInfo](#)

## Referenced elements

The following element is referenced in Security service operation requests and responses:

- [AccountabilityInfo](#)

## AccountabilityInfo

### Referenced from

- [lookupUserInfo](#)

### Contents

AccountabilityInfo has the following elements

Element	Type	Occ.	Description
clearanceID	<a href="#">xs:string</a>	1	Displays the user's clearance policy ID. This designates the user's level of accountability clearance. <b>Note:</b> If you request accountability information, but accountability is either not enabled for the system or for the requested user, the clearanceID will be <b>NONE</b> .
patronType	<a href="#">xs:string</a>	0..1	Displays a value of either <b>INTERNAL</b> or <b>EXTERNAL</b> . An INTERNAL patron is a patron of the library that holds the material. An EXTERNAL patron is typically a patron that is temporarily stationed at one classified facility, but is registered at another one.
clearanceGrantingFacility	<a href="#">xs:string</a>	0..1	Displays the name of the institution that originally granted the user's security clearance.
citizenshipID	<a href="#">xs:string</a>	0..1	Displays the user's assigned citizenship policy ID.
ssn	<a href="#">xs:string</a>	0..1	Displays the user's Social Security Number.
birthDate	<a href="#">xs:string</a>	0..1	Displays the user's birth date.
birthPlace	<a href="#">xs:string</a>	0..1	Displays the user's place of birth.
accessRestrictionID	<a href="#">xs:string</a>	0..*	Displays the access restriction policy IDs assigned to the user.
needToKnowID	<a href="#">xs:string</a>	0..*	Displays the need to know policy IDs assigned to the user.

## Patron service

The Patron service includes operations that a patron can do for themselves. Operations in this service can only be performed on the user record for the patron that is logged in.

The Patron service includes these operations (listed alphabetically):

Operation	Description
<a href="#">activateOnlineRegisteredUser</a>	Activates a user who completed online user registration.

Operation	Description
<b>changeMyAlternativeID</b>	Specifies a new Alternative User ID for the session user.
<b>cancelMyHold</b>	Cancels a hold for the specified item or title for the session user.
<b>changeMyPIN</b>	Specifies a new PIN for the session user.
<b>createMyHold</b>	Creates a hold for the specified item or title for the session user. Returns only a hold key. Compare with placeMyHold.
<b>createSelfRegisteredPatron</b>	Creates a new patron record based on rules in the Online User Registration setup. The Online User Registration feature of the ILS must be enabled and properly configured to use this operation.
<b>deleteMyList</b>	Deletes the specified permanent list for the session user.
<b>enableMyCheckoutHistory</b>	Specifies whether to enable or disable the keeping of circulation history for the patron.
<b>getMyList</b>	Retrieves the specified permanent list in a format suitable for updating data with saveMyList.
<b>lookupMyAccountInfo</b>	Retrieves information about the current session user's account and account activity.
<b>lookupMyList</b>	Displays the specified permanent list for the session user; optionally includes brief title information for each title in the list.
<b>lookupMyLists</b>	Displays a list of the session user's permanent lists.
<b>lookupPatronInfo</b>	Retrieves information about a specified user's account and account activity.
<b>modifyMyHold</b>	Lets you change the expiration date of the specified hold or specify a pickup library or mail service (if holds by mail is enabled in the ILS). If you specify a mail service, the hold is automatically flagged for mail delivery. If you specify a pickup library, the mail flag is cleared.
<b>placeMyHold</b>	Creates a hold for the specified item or title for the session user. Returns a hold key and other hold information. Compare with createMyHold.
<b>renewMyCheckout</b>	Renews the specified item for the session user.
<b>saveMyList</b>	Allows you to save a new or updated My List for the current session user.

Operation	Description
<a href="#">suspendMyHold</a>	Suspends the specified hold for the duration you specify by start and end date.  The ILS validates dates. For example, the start or end date cannot be later than the hold expiration date. Consult with the library system administrator for information about what rules the ILS uses for dates.
<a href="#">unsuspendMyHold</a>	Activates the specified hold if it is currently suspended (it clears the suspension start and end dates).

### Related topics

- [Services and Operations on page 41](#)

## activateOnlineRegisteredUser

The activateOnlineRegisteredUser operation allows an authenticated staff member to activate a user who registered through the online user registration process (see [createSelfRegisteredPatron on page 85](#)). The staff member must have the appropriate rights to use this operation: permission to use the ACTIVUSROL command code.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="#">http://schemas.sirsiidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">ActivateOnlineRegisteredUserRequest</a>
Response	<a href="#">ActivateOnlineRegisteredUserResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

### ActivateOnlineRegisteredUserRequest

The ActivateOnlineRegisteredUserRequest has the following request parameters:

Parameter	Type	Description	Required?
assignedUserID	<a href="#">xs:string</a>	Specifies the ID assigned to the user (patron) during the self-registration process.	Y
newUserID	<a href="#">xs:string</a>	Specifies the new user ID to use for the patron. SirsiDynix Symphony may require a value for this parameter even though Web Services does not.	N
newAltID	<a href="#">xs:string</a>	Specifies the Alternate User ID to use for the patron.	N
newWebAuthID	<a href="#">xs:string</a>	Specifies the patron's single-sign-on ID.	N
webAuthOverride	<a href="#">xs:string</a>	Specifies the override for using the single-sign-on ID. If you submit a value for newWebAuthId, you must also include a value for this parameter.	N
pin	<a href="#">xs:string</a>	Specifies the PIN to use for the patron. SirsiDynix Symphony may require a value for this parameter even though Web Services does not.	N

### ActivateOnlineRegisteredUserResponse

The content of ActivateOnlineRegisteredUserResponse is Boolean (see [xs:boolean](#)). The output will be **true** if the user activation was successful. A fault is thrown if there is a problem with the newUserID, newAltID, webAuthOverride, or pin.

### See Also

- [Patron service](#) on page 77
- [createSelfRegisteredPatron](#) on page 85

## changeMyAlternativeID

The changeMyAlternativeID operation allows the current session user to change the Alternative User ID. If the Symphony instance does not use Alternative User IDs, an exception gets thrown when you use the operation.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/patron">http://schemas.sirsidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> <li>• Valid sessionToken (see <a href="#">sessionToken</a>)</li> </ul>
Request	<a href="#">ChangeMyAlternativeIDRequest</a>
Response	<a href="#">ChangeMyAlternativeIDResponse</a>
Operation faults	None (see <a href="#">Faults</a> )



### ChangeMyAlternativeIDRequest

The ChangeMyAltIDRequest has the following request parameters:

Parameter	Type	Description	Required?
newAlternativeID	<a href="#">xs:string</a>	Specifies the new Alternative User ID to use.	Y

### ChangeMyAlternativeIDResponse

The ChangeMyAltIDResponse is boolean (see [xs:boolean](#)). The output will be **true** if the Alternative User ID was successfully changed. If Symphony does not support Alternative User IDs or if another user already has the requested Alternative User ID, the system throws an Error Response Fault.

#### Related topics

- [Patron service](#)

## cancelMyHold

The cancelMyHold operation allows you to cancel a hold for the current session user.

#### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsiidynix.com/symws/patron">http://schemas.sirsiidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">CancelMyHoldRequest</a>
Response	<a href="#">CancelMyHoldResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

### CancelMyHoldRequest

The CancelMyHoldRequest has the following request parameters:

Parameter	Type	Description	Required?
holdKey	<a href="#">xs:long</a>	Specifies the hold key for the hold you are cancelling.	Y

### CancelMyHoldResponse

The CancelMyHoldResponse is boolean (see [xs:boolean](#)). The output will be **true** if the hold was canceled. If the hold was not cancelled, the system throws an Error Response Fault.

**Related topics**

- [Patron service](#)

**changeMyPIN**

The changeMyPIN operation allows you to change the current session user's PIN to a new PIN (provided that the current PIN is validated).

**Synopsis**

Service	<a href="#">Patron service</a>
Namespace	http://schemas.sirsidynix.com/symws/patron
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">ChangeMyPINRequest</a>
Response	<a href="#">ChangeMyPINResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

**ChangeMyPINRequest**

TheChangeMyPINRequest has the following request parameters:

Parameter	Type	Description	Required?
currentPIN	<a href="#">xs:string</a>	Specifies the user's existing PIN.	Y
newPIN	<a href="#">xs:string</a>	Specifies the user's new PIN.	Y

**ChangeMyPINResponse**

The ChangeMyPINResponse is boolean (see [xs:boolean](#)). The output will be **true** if the PIN change was successful. If the operation fails, an Error Response Exception or a Patron Service Exception will be returned.

**Related topics**

- [Patron service](#)

**createMyHold**

The createMyHold operation allows you to make a hold for the current session user and get a hold key back.

This operation is different than the [placeMyHold](#) operation, which returns the hold key plus additional information about the request.

**Synopsis**

Service	<b>Patron service</b>
Namespace	http://schemas.sirsidynix.com/symws/patron
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <b>clientID</b>)</li> </ul>
Request	<b>CreateMyHoldRequest</b>
Response	<b>CreateMyHoldResponse</b>
Operation faults	None (see <b>Faults</b> )

### CreateMyHoldRequest

The CreateMyHoldRequest has the following request parameters:

Parameter	Type	Description	Required?
<b>itemID</b> and <b>titleKey</b> are a choice. Specify a value for one or the other.			
itemID	<b>xs:string</b>	Specifies the item ID for the item being placed on hold. <b>Note:</b> Only specify an itemID or a titleKey. Do not specify both.	Y
titleKey	<b>xs:long</b>	Specifies the titleKey for the item being placed on hold. <b>Note:</b> Only specify a titleKey or an itemID. Do not specify both.	
callNumber	<b>xs:string</b>	Specifies the call number for the item being placed on hold. This is mainly used for placing volume-level holds. For information about how to determine whether this is needed, see <a href="#">Volume holds on page 12</a> .	N
holdType	<b>HoldType</b>	Specifies the enumerated hold type value. If you do not specify a holdType value, the system assumes <b>TITLE</b> by default.	N
holdRange	<b>HoldRange</b>	Specifies the enumerated hold range value. If you do not specify a holdRange value, the system assumes <b>GROUP</b> by default.	N
<b>pickupLibraryID</b> and <b>mailServiceID</b> are a choice. Specify a value for one or the other.			
pickupLibraryID	<b>xs:string</b>	Specifies the pickup library Library ID if the hold will be picked up by the patron. <b>Note:</b> Do not specify a pickupLibraryID if you specify a mailServiceID.	Y
mailServiceID	<b>xs:string</b>	Specifies the Mail Service Tracking ID if the hold will be delivered by mail. <b>Note:</b> Do not specify a mailServiceID if you specify a pickupLibraryID.	
expiresDate	<b>NillableDate</b>	Specifies the hold expiration date. You can specify a value that will override what is existing, or you can specify to nullify the existing date entirely. If you do not specify an expiresDate value, the expiresDate will be calculated by the ILS.	N
suspendStartDate	<b>xs:date</b>	Specifies the start date of the hold suspension.	N
suspendEndDate	<b>xs:date</b>	Specifies the date the hold suspension ends.	N
comment	<b>xs:string</b>	Specifies optional text to display with the hold information.	N

### CreateMyHoldResponse

The CreateMyHoldResponse is xs:long (see **xs:long**). The output will be the holdKey.

**Related topics**

- [Patron service](#)
- [HoldType](#)
- [HoldRange](#)
- [NillableDate](#)

**createSelfRegisteredPatron**

The createSelfRegisteredPatron operation allows you to create a new patron using On Line User Registration rules.

The user who is used to make this call must have the "Create User OnLine" command code privilege for making Online User Registration calls. For additional information, see [Online User Registration](#) on page 16.

**Synopsis**

<b>Service</b>	<a href="#">Patron service</a>
<b>Namespace</b>	http://schemas.sirsiidynix.com/symws/patron
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">CreateSelfRegisteredPatronRequest</a>
<b>Response</b>	<a href="#">CreateSelfRegisteredPatronResponse</a>
<b>Operation faults</b>	None (see <a href="#">Faults</a> )

**CreateSelfRegisteredPatronRequest**

The CreateSelfRegisteredPatronRequest has the following request parameters:



Depending on the specific library configuration, other fields may be required in the request.

Parameter	Type	Description	Required?
firstName	<a href="#">xs:string</a>	Specifies the patron's first name.	Y
middleName	<a href="#">xs:string</a>	Specifies the patron's middle name.	N
lastName	<a href="#">xs:string</a>	Specifies the patron's last name.	Y
nameSuffix	<a href="#">xs:string</a>	Specifies the patron's name suffix.	N
preferredName	<a href="#">xs:string</a>	Specifies the patron's preferred name.	N
socialSecurityNumber	<a href="#">xs:string</a>	Specifies the patron's Social Security Number.	N
birthDate	<a href="#">xs:string</a>	Specifies the patron's birth date.	N
pin	<a href="#">xs:string</a>	Specifies the users PIN.	N
street	<a href="#">xs:string</a>	Specifies the patron's street address.	Y
apartment	<a href="#">xs:string</a>	Specifies the patron's apartment number.	N
city	<a href="#">xs:string</a>	Specifies the city where the patron resides.	Y
state	<a href="#">xs:string</a>	Specifies the state where the patron resides.	Y
postalCode	<a href="#">xs:string</a>	Specifies the patron's ZIP code.	Y
homePhone	<a href="#">xs:string</a>	Specifies the patron's home phone number.	N
emailAddress	<a href="#">xs:string</a>	Specifies the patron's email address.	N
patronLibraryID	<a href="#">xs:string</a>	Specifies the Library ID of the patron's library.	Y

### CreateSelfRegisteredPatronResponse

The CreateSelfRegisteredPatronResponse is a string (see [xs:string](#)). The output is the patron's newly generated userID.

#### Related topics

- [Patron service](#)

## deleteMyList

The deleteMyList operation allows you to delete My List entries for the current session user.

#### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsiidynix.com/symws/patron">http://schemas.sirsiidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">DeleteMyListRequest</a>
Response	<a href="#">DeleteMyListResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

### DeleteMyListRequest

The DeleteMyListRequest has the following request parameters:

Parameter	Type	Description	Required?
myListID	<a href="#">xs:string</a>	Specifies the user's My List ID.	Y

### DeleteMyListResponse

The DeleteMyListResponse is boolean (see [xs:boolean](#)). The output will be **true** if the user's My List was deleted and **false** if it was not.

### Related topics

- [Patron service](#)

## enableMyCheckoutHistory

The enableMyCheckoutHistory operation allows patrons to set whether to keep their circulation (checkout) history. This must also be enabled in SirsiDynix Symphony. For more information, see [isCheckoutHistoryEnabled](#) on page 141.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsiidynix.com/symws/patron">http://schemas.sirsiidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">enableMyCheckoutHistoryRequest</a>
Response	<a href="#">enableMyCheckoutHistoryResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### enableMyCheckoutHistoryRequest

The enableMyCheckoutHistoryRequest has the following request parameter:

Parameter	Type	Description	Required?
enable	<a href="#">xs:boolean</a>	Specifies whether to enable checkout history (true) or not (false).	Y

### enableMyCheckoutHistoryResponse

The enableMyCheckoutHistoryResponse is boolean (see [xs:boolean](#)). The output will be **true** if the change to enable checkout history is successful. Otherwise, a fault is returned if the change was not allowed or if checkout history is not enabled.

### Related topics

- [Patron service](#) on page 77
- [isCheckoutHistoryEnabled](#) on page 141

## getMyList

The getMyList operation allows you to retrieve a My List entry for the current session user in a format that can then be modified.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsiidynix.com/symws/patron">http://schemas.sirsiidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">GetMyListRequest</a>
Response	<a href="#">GetMyListResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

### GetMyListRequest

The GetMyListRequest has the following parameters:

Parameter	Type	Description	Required?
myListID	<a href="#">xs:string</a>	Specifies the user's My List ID.	Y

### GetMyListResponse

The GetMyListResponse returns a MyList (see [MyList](#) for a list of elements returned in this response).



**Related topics**

- [Patron service](#)
- [MyList](#)

**lookupMyAccountInfo**

The lookupMyAccountInfo operation allows you to retrieve information about the current session user's account and account activity.

**Synopsis**

<b>Service</b>	<a href="#">Patron service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/patron
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupMyAccountInfoRequest</a>
<b>Response</b>	<a href="#">LookupMyAccountInfoResponse</a>
<b>Operation faults</b>	None (see <a href="#">Faults</a> )

**LookupMyAccountInfoRequest**

The LookupMyAccountInfoRequest has the following request parameters:

Parameter	Type	Description	Required?
includePatronInfo	<b>xs:boolean</b>	Specifies if you want to include patron information in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronInfo, the system will assume “false” by default.	N
includePatronCirculationInfo	<b>xs:boolean</b>	Specifies if you want to include patron circulation information in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronCirculationInfo, the system will assume “false” by default.	N
includePatronAddressInfo	<b>xs:boolean</b>	Specifies if you want to include patron address information in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronAddressInfo, the system will assume “false” by default.	N
includePatronCheckoutInfo	<b>CheckoutInfoFilter</b>	Specifies an enumeration value that limits the patron checkout information you receive in the response. For example, specify <b>OVERDUE</b> as the value in order to only include information about checked out items that are overdue.	N
includePatronCheckoutHistoryInfo	<b>xs:boolean</b>	Specifies if you want to include the patron's checkout history in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronCheckoutHistoryInfo, the system will assume “false” by default.	N
includePatronHoldInfo	<b>HoldInfoFilter</b>	Specifies an enumeration value that limits the patron hold information you receive in the response.	N
includeFeeInfo	<b>FeeInfoFilter</b>	Specifies an enumeration value that limits the fee information you receive in the response.	N
includePatronStatusInfo	<b>xs:boolean</b>	Specifies if you want to include patron information in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronStatusInfo, the system will assume “false” by default.	N

Parameter	Type	Description	Required?
includeUserGroupInfo	<b>xs:boolean</b>	Specifies if you want to include user group information in the response ( <b>true</b> ) or not ( <b>false</b> ). Use the <b>isUserGroupsEnabled</b> operation to verify that User Groups are enabled in the ILS. <b>Note:</b> If you do not specify to includeUserGroupInfo, the system will assume “false” by default.	N
includeUserSuspensionInfo	<b>xs:boolean</b>	Specifies if you want to include information pertaining to a user's suspension in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includeUserSuspensionInfo, the system will assume “false” by default.	N
stationLibraryID	<b>xs:string</b>	Specifies the Library ID of the patron's station library.	N

### LookupMyAccountInfoResponse

The LookupMyAccountInfoResponse returns the following elements:

Element	Type	Occ.	Description
patronInfo	<b>PatronInfo</b>	0..1	Displays information about a patron. patronInfo is returned only if includePatronInfo is <b>true</b> .
patronCirculationInfo	<b>PatronCirculationInfo</b>	0..1	Displays specific circulation information for a patron. patronCirculationInfo is returned only if includePatronCirculationInfo is <b>true</b> .
patronAddressInfo	<b>PatronAddressInfo</b>	0..1	Displays address information for a patron. patronAddressInfo is returned only if includePatronAddressInfo is <b>true</b> .
patronCheckoutInfo	<b>PatronCheckoutInfo</b>	0..*	Displays specific checkout information for a patron. patronCheckoutInfo is returned only if you enter an includePatronCheckoutInfo value.
patronCheckoutHistoryInfo	<b>PatronCheckoutHistoryInfo</b>	0..*	Displays specific checkout history information for a patron. patronCheckoutHistoryInfo is returned only if includePatronCheckoutHistoryInfo is <b>true</b> .
patronHoldInfo	<b>PatronHoldInfo</b>	0..*	Displays specific hold information for a patron. patronHoldInfo is returned only if you enter an includePatronHoldInfo value.
feeInfo	<b>FeeInfo</b>	0..*	Displays information about a patron's fees and bills. feeInfo is returned only if you enter an includePatronFeeInfo value.
patronStatusInfo	<b>PatronStatusInfo</b>	0..1	Displays information specific to a patron's status. patronStatusInfo is returned only if includePatronStatusInfo is <b>true</b> .
groupInfo	<b>GroupInfo</b>	0..1	Displays information about a patron's user group, information about user group transactions (including holds for each user, charges/checkouts for each user, and bills for each user) and additional information about the group, such as group permissions, group member information, and other miscellaneous information pertaining to the group. userGroupInfo is returned only if includeUserGroupInfo is <b>true</b> .
userSuspensionInfo	<b>UserSuspensionInfo</b>	0..1	Displays information about a patron's suspension. userSuspensionInfo is returned only if includeSuspensionInfo is <b>true</b> .

### Related topics

- [Patron service](#)

## lookupMyList

The lookupMyList operation allows you to retrieve one of the current session user's My Lists.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/patron">http://schemas.sirsidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>Valid <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupMyListRequest</a>
Response	<a href="#">LookupMyListResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

### LookupMyListRequest

The LookupMyListRequest has the following request parameters:

Parameter	Type	Description	Required?
myListID	<a href="#">xs:string</a>	Specifies the MyList ID.	Y
includeMyListTitleInfo	<a href="#">xs:boolean</a>	<p>Specifies if you want to include MyList title information (<b>true</b>) or not (<b>false</b>).</p> <p><b>Note:</b> If you do not specify a value, the system assumes “false” by default.</p>	N

### LookupMyListResponse

The LookupMyListResponse returns MyListInfo (see [MyListInfo](#) for a list of elements returned in this response).

### Related topics

- [Patron service](#)
- [MyListInfo](#)

## lookupMyLists

The lookupMyLists operation allows you to retrieve all My Lists that belong to the current session user.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/patron">http://schemas.sirsidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>Valid <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupMyListsRequest</a>

Service	<a href="#">Patron service</a>
Response	<a href="#">LookupMyListsResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

### LookupMyListsRequest

The LookupMyListsRequest has no request parameters.

### LookupMyListsResponse

The LookupMyListsResponse returns the following elements:

Element	Type	Occ.	Description
myList	<a href="#">MyList</a>	0..*	<p>The MyList data type is the first Entity object. It is used for CRUD operations dealing with MyList. MyList extends “Entity” which dictates which operation is to be performed on the MyList and MyListEntry elements. (For more information, see <a href="#">Entity</a>.)</p> <p><b>Note:</b> the myList element will only contain identifying information in this response. MyListEntry will be empty.</p>



If you want a list of titles, use the [lookupMyList](#) operation.

### Related topics

- [Patron service](#)
- [MyList](#)

## lookupPatronInfo

The lookupPatronInfo operation allows you to retrieve information about a specified user’s account and account activity.

This is a privileged operation.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/patron">http://schemas.sirsidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> <li>• Valid sessionToken (see <a href="#">Authenticated operations</a>) from user whose user access includes Display2 User Command Code.</li> </ul>
Request	<a href="#">LookupPatronInfoRequest</a>

Service	<a href="#">Patron service</a>
Response	<a href="#">LookupPatronInfoResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

### LookupPatronInfoRequest

The LookupPatronInfoRequest has the following request parameters:

Parameter	Type	Description	Required?
<b>userID</b> , <b>alternateID</b> , and <b>webAuthID</b> are a choice. Specify a value for only one of them.			
userID	<b>xs:string</b>	Specifies the user ID of the patron to look up information about. <b>Note:</b> You only need to specify a userID, alternateID, or webAuthID. If you do not specify a userID, you must specify either an alternateID or webAuthID.	Y
alternateID	<b>xs:string</b>	Specifies the alternate ID of the patron to look up information about. <b>Note:</b> You only need to specify a userID, alternateID, or webAuthID. If you do not specify an alternateID, you must specify either a userID or webAuthID.	
webAuthID	<b>xs:string</b>	Specifies the web auth ID of the patron to look up information about. <b>Note:</b> You only need to specify a userID, alternateID, or webAuthID. If you do not specify a webAuthID, you must specify either a userID or alternateID.	
includePatronInfo	<b>xs:boolean</b>	Specifies if you want to include patron information in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronInfo, the system will assume “false” by default.	N
includePatronCirculationInfo	<b>xs:boolean</b>	Specifies if you want to include patron circulation information in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronCirculationInfo, the system will assume “false” by default.	N
includePatronAddressInfo	<b>xs:boolean</b>	Specifies if you want to include patron address information in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronAddressInfo, the system will assume “false” by default.	N
includePatronCheckoutInfo	<b>CheckoutInfoFilter</b>	Specifies an enumeration value that limits the patron checkout information you receive in the response.  For example, specify <b>OVERDUE</b> as the value in order to only include information about checked out items that are overdue.	N



Parameter	Type	Description	Required?
includePatronCheckoutHistoryInfo	<b>xs:boolean</b>	Specifies if you want to include the patron's checkout history in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronCheckoutHistoryInfo, the system will assume “false” by default.	N
includePatronHoldInfo	<b>HoldInfoFilter</b>	Specifies an enumeration value that limits the patron hold information you receive in the response.	N
includeFeeInfo	<b>FeeInfoFilter</b>	Specifies an enumeration value that limits the fee information you receive in the response.	N
includePatronStatusInfo	<b>xs:boolean</b>	Specifies if you want to include patron information in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includePatronStatusInfo, the system will assume “false” by default.	N
includeUserGroupInfo	<b>xs:boolean</b>	Specifies if you want to include user group information in the response ( <b>true</b> ) or not ( <b>false</b> ). Use the <b>isUserGroupsEnabled</b> operation to verify that User Groups are enabled in the ILS. <b>Note:</b> If you do not specify to includeUserGroupInfo, the system will assume “false” by default.	N
includeUserSuspensionInfo	<b>xs:boolean</b>	Specifies if you want to include information pertaining to a user's suspension in the response ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> If you do not specify to includeUserSuspensionInfo, the system will assume “false” by default.	N

### LookupPatronInfoResponse

The LookupPatronInfoResponse returns the following elements:

Element	Type	Occ.	Description
patronInfo	<b>PatronInfo</b>	0..1	Displays information about a patron. patronInfo is returned only if includePatronInfo is <b>true</b> .
patronCirculationInfo	<b>PatronCirculationInfo</b>	0..1	Displays specific circulation information for a patron. patronCirculationInfo is returned only if includePatronCirculationInfo is <b>true</b> .
patronAddressInfo	<b>PatronAddressInfo</b>	0..1	Displays address information for a patron. patronAddressInfo is returned only if includePatronAddressInfo is <b>true</b> .
patronCheckoutInfo	<b>PatronCheckoutInfo</b>	0..*	Displays specific checkout information for a patron. patronCheckoutInfo is returned only if you enter an includePatronCheckoutInfo value.
patronCheckoutHistoryInfo	<b>PatronCheckoutHistoryInfo</b>	0..*	Displays specific checkout history information for a patron. patronCheckoutHistoryInfo is returned only if includePatronCheckoutHistoryInfo is <b>true</b> .
patronHoldInfo	<b>PatronHoldInfo</b>	0..*	Displays specific hold information for a patron. patronHoldInfo is returned only if you enter an includePatronHoldInfo value.
feeInfo	<b>FeeInfo</b>	0..*	Displays information about a patron's fees and bills. feeInfo is returned only if you enter an includePatronFeeInfo value.
patronStatusInfo	<b>PatronStatusInfo</b>	0..1	Displays information specific to a patron's status. patronStatusInfo is returned only if includePatronStatusInfo is <b>true</b> .

Element	Type	Occ.	Description
groupInfo	<b>GroupInfo</b>	0..1	Displays information about a patron's user group, information about the user group transactions (including holds for each user, charges/checkouts for each user, and bills for each user) and additionally retrieve the group permissions a user has pertaining to their group, group member information, and other miscellaneous information pertaining to the group.  userGroupInfo is returned only if includeUserGroupInfo is <b>true</b> .
userSuspensionInfo	<b>UserSuspensionInfo</b>	0..1	Displays information about a patron's suspension.  userSuspensionInfo is returned only if includeSuspensionInfo is <b>true</b> .

**Related topics**

- [Patron service](#)

**modifyMyHold**

The modifyMyHold operation allows you to modify certain hold values for a current session user's hold.

**Synopsis**

<b>Service</b>	<b><a href="#">Patron service</a></b>
<b>Namespace</b>	<a href="http://schemas.sirsiidynix.com/symws/patron">http://schemas.sirsiidynix.com/symws/patron</a>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<b><a href="#">ModifyMyHoldRequest</a></b>
<b>Response</b>	<b><a href="#">ModifyMyHoldResponse</a></b>
<b>Operation faults</b>	None (see <a href="#">Faults</a> )

**ModifyMyHoldRequest**

The ModifyMyHoldRequest has the following request parameters:

Parameter	Type	Description	Required?
holdKey	<a href="#">xs:long</a>	Specifies the hold key for the hold being modified.	Y
expiresDate	<a href="#">NillableDate</a>	Specifies the hold expiration date. You can specify a value that will override what is existing, or you can specify to nullify the existing date entirely. If you do not specify an expiresDate value, the expiresDate will be calculated by the ILS.	N
<b>holdPickupLibraryID</b> and <b>mailDeliveryID</b> are a choice. Specify a value for one or the other.			
holdPickupLibraryID	<a href="#">xs:string</a>	Specifies the Library ID for the library if the hold will be picked up by the patron. <b>Note:</b> If you enter a holdPickupLibraryID, do not specify a mailDeliveryID. <b>Note:</b> If you enter a holdPickupLibraryID and the ILS is not config.multi (that is, it is a single-library instance), the request will succeed even though no change is made.	N
mailDeliveryID	<a href="#">xs:string</a>	Specifies the Mail Service Tracking ID if the hold will be delivered by mail. <b>Note:</b> If you enter a mailDeliveryID, do not specify a holdPickupLibraryID.	

### ModifyMyHoldResponse

The ModifyMyHoldResponse is boolean (see [xs:boolean](#)). The output will be **true** if the request to modify a hold was successful, and **false** if it was not.



A fault will be thrown if the user attempting to modify a hold does not own the hold reference by the supplied holdKey.

### Related topics

- [Patron service](#)

## placeMyHold

The placeMyHold operation allows you to make a hold for the current session user and to get back information about the request.

This operation is different than the [createMyHold](#) operation, which only returns the hold key.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsiidynix.com/symws/patron">http://schemas.sirsiidynix.com/symws/patron</a>

Service	<a href="#">Patron service</a>
Prerequisites	<ul style="list-style-type: none"><li>Valid clientID (see <a href="#">clientID</a>)</li></ul>
Request	<a href="#">PlaceMyHoldRequest</a>
Response	<a href="#">PlaceMyHoldResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

### **PlaceMyHoldRequest**

The PlaceMyHoldRequest has the following request parameters:

Parameter	Type	Description	Required?
<b>itemID</b> and <b>titleKey</b> are a choice. Specify a value for one or the other.			
itemID	<b>xs:string</b>	Specifies the item ID for the item being placed on hold. <b>Note:</b> Only specify an itemID or a titleKey. Do not specify both.	Y
titleKey	<b>xs:long</b>	Specifies the titleKey for the item being placed on hold. <b>Note:</b> Only specify a titleKey or an itemID. Do not specify both.	
callNumber	<b>xs:string</b>	Specifies the call number for the item being placed on hold. This is mainly used for placing volume-level holds. For information about how to determine whether this is needed, see <a href="#">Volume holds on page 12</a> .	N
holdType	<b>HoldType</b>	Specifies the enumerated hold type value. If you do not specify a holdType value, the system assumes <b>TITLE</b> by default.	N
holdRange	<b>HoldRange</b>	Specifies the enumerated hold range value. If you do not specify a holdRange value, the system assumes <b>GROUP</b> by default.	N
<b>pickupLibraryID</b> and <b>mailServiceID</b> are a choice. Specify a value for one or the other.			
pickupLibraryID	<b>xs:string</b>	Specifies the pickup library Library ID if the hold will be picked up by the patron. <b>Note:</b> Do not specify a pickupLibraryID if you specify a mailServiceID.	Y
mailServiceID	<b>xs:string</b>	Specifies the Mail Service Tracking ID if the hold will be delivered by mail. <b>Note:</b> Do not specify a mailServiceID if you specify a pickupLibraryID.	
expiresDate	<b>NillableDate</b>	Specifies the hold expiration date. You can specify a value that will override what is existing, or you can specify to nullify the existing date entirely. If you do not specify an expiresDate value, the expiresDate will be calculated by the ILS.	N
suspendStartDate	<b>xs:date</b>	Specifies the start date of the hold suspension.	N
suspendEndDate	<b>xs:date</b>	Specifies the date the hold suspension ends.	N
comment	<b>xs:string</b>	Specifies optional text to display with the hold information.	N

### PlaceMyHoldResponse

The PlaceMyHoldResponse returns the following elements:

Element	Type	Occ.	Description
holdKey	<a href="#">xs:long</a>	1	Displays the hold key for the request.
estimatedFee	<a href="#">Money</a>	0..1	Displays the estimated fee assessed, if any, for placing a hold on the item.
title	<a href="#">xs:string</a>	0..1	Displays the title of the hold request.
author	<a href="#">xs:string</a>	0..1	Displays the author of the hold request.
itemAvailable	<a href="#">xs:boolean</a>	1	Displays whether the item requested is currently available.
callNumber	<a href="#">xs:string</a>	1	Displays the call number of the item.
itemID	<a href="#">xs:string</a>	0..1	Displays the item ID for the hold request item.
dateHoldExpires	<a href="#">xs:date</a>	0..1	Displays the date the hold request expires.
dateHoldExpiresOnShelf	<a href="#">xs:date</a>	0..1	Displays the date on or by which the user must pick up the available hold before it expires on the Holds shelf.

#### Related topics

- [Patron service](#)
- [HoldType](#)
- [HoldRange](#)
- [NillableDate](#)

## renewMyCheckout

The `renewMyCheckout` operation allows you to renew a checked out item for the current session user.

#### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/patron">http://schemas.sirsidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>Valid <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">RenewMyCheckoutRequest</a>
Response	<a href="#">RenewMyCheckoutResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

#### RenewMyCheckoutRequest

The `RenewMyCheckoutRequest` has the following request parameters:

Parameter	Type	Description	Required?
itemID	<a href="#">xs:string</a>	Specifies the item ID of the item being renewed.	Y

### RenewMyCheckoutResponse

The RenewMyCheckoutResponse returns the following elements:

Element	Type	Occ.	Description
message	<a href="#">xs:string</a>	1	Displays a renewal message.
userID	<a href="#">xs:string</a>	1	Displays the user ID of the patron renewing the checkout.
userName	<a href="#">xs:string</a>	1	Displays the name of the patron renewing the checkout.
itemID	<a href="#">xs:string</a>	1	Displays the item ID of the item being renewed.
callNumber	<a href="#">xs:string</a>	1	Displays the call number of the item being renewed.
title	<a href="#">xs:string</a>	1	Displays the title of the item being renewed.
author	<a href="#">xs:string</a>	0..1	Displays the author of the item being renewed.
dueDate	<a href="#">xs:dateTime</a>	0..1	Displays the new due date for the item.

### Related topics

- [Patron service](#)

## saveMyList

The saveMyList operation allows you to save a new or updated My List for the current session user.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	<a href="http://schemas.sirsiidynix.com/symws/patron">http://schemas.sirsiidynix.com/symws/patron</a>
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	The SaveMyListRequest and SaveMyListResponse are both MyList data types (for details, see <a href="#">MyList</a> ).
Response	
Operation faults	None (see <a href="#">Faults</a> )

### SaveMyListRequest

The SaveMyListRequest is a MyList data type (for details, see [MyList](#)).



## SaveMyListResponse

The SaveMyListResponse is a MyList data type (for details, see [MyList](#)).

### Related topics

- [Patron service](#)
- [MyList](#)

## suspendMyHold

The suspendMyHold operation allows you to suspend a current session user's hold.

### Synopsis

Service	<a href="#">Patron service</a>
Namespace	http://schemas.sirsiidynix.com/symws/patron
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">SuspendMyHoldRequest</a>
Response	<a href="#">SuspendMyHoldResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

## SuspendMyHoldRequest

The SuspendMyHoldRequest has the following request parameters:

Parameter	Type	Description	Required?
holdKey	<a href="#">xs:long</a>	Specifies the hold key for the item on hold.	Y
suspendStartDate	<a href="#">xs:date</a>	Specifies the start date for the hold suspension. <b>Note:</b> The suspendStartDate must be after the current date, and no later than the hold expiration date.	Y
suspendEndDate	<a href="#">xs:date</a>	Specifies the date the hold suspension ends. <b>Note:</b> The suspendEndDate must not be later than the hold expiration date	N

## SuspendMyHoldResponse

The SuspendMyHoldResponse is boolean (see [xs:boolean](#)). The output will be **true** if the request to suspend a hold was successful, and **false** if it was not.



A fault will be thrown if the user attempting to suspend a hold does not own the hold reference by the supplied holdKey.

**Related topics**

- [Patron service](#)

## unsuspendMyHold

The unsuspendMyHold operation allows you to unsuspend a current session user's hold.

**Synopsis**

Service	<a href="#">Patron service</a>
Namespace	http://schemas.sirsiDynix.com/symws/patron
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">UnsuspendMyHoldRequest</a>
Response	<a href="#">UnsuspendMyHoldResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

**UnsuspendMyHoldRequest**

The UnsuspendMyHoldRequest has the following request parameters:

Parameter	Type	Description	Required?
holdKey	<a href="#">xs:long</a>	Specifies the hold key for the item on hold.	Y

**UnsuspendMyHoldResponse**

The UnsuspendMyHoldResponse is boolean (see [xs:boolean](#)). The output will be **true** if the request to suspend a hold was successful, and **false** if it was not.



A fault will be thrown if the user attempting to unsuspend a hold does not own the hold reference by the supplied holdKey.

**Related topics**

- [Patron service](#)

## Request service

Requests in SirsiDynix Symphony allows users to interact with library staff by sending library-defined or anonymous requests. However, all these request operations (except lookupRequestTemplate) must be accompanied by a sessionToken of an authenticated login of a staff level or higher user.

In addition to these Request service operations, several helper operations have been added to the Admin service. For details, see [Admin service on page 116](#).

The Request service includes these operations (listed alphabetically):

Operation	Description
<a href="#">cancelRequest</a>	Cancels the specified request.
<a href="#">createRequest</a>	Creates a new request.
<a href="#">lookupRequest</a>	Returns detailed information about a specific request.
<a href="#">lookupRequests</a>	Returns a list of requests for a specified user.
<a href="#">lookupRequestTemplate</a>	Returns a Request Template for determining how to display the request, how to qualify entries, and how to populate a new request.

### Related topics

- [Services and Operations on page 41](#)

## cancelRequest

The cancelRequest operation allows you to cancel a single request.

### Synopsis

Service	<a href="#">Request service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/request">http://schemas.sirsidynix.com/symws/request</a>
Prerequisites	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> <li>• Valid sessionToken (see <a href="#">Authenticated operations</a>) from user with staff level or higher privileges</li> </ul>
Request	<a href="#">cancelRequest</a>
Response	<a href="#">cancelRequest</a>
Operation faults	None (see <a href="#">Faults</a> )

### CancelRequestRequest

The CancelRequestRequest has the following request parameters:

Parameter	Type	Description	Required?
requestKey	<a href="#">xs:string</a>	Specifies the request key for the request to cancel. The request key is available from the <a href="#">lookupRequests</a> operation.	Y

**CancelRequestResponse**

The CancelRequestResponse is boolean (see [xs:boolean](#)). The output will be **true** if the request was canceled, **false** if not.

If the requestKey is not valid, a fault is thrown.

**Related topics**

- [Request service](#)

**createRequest**

The createRequest operation allows you to create a request.

**Synopsis**

<b>Service</b>	<a href="#">Request service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/request
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> <li>• Valid sessionToken (see <a href="#">Authenticated operations</a>) from user with staff level or higher privileges</li> </ul>
<b>Request</b>	<a href="#">createRequest</a>
<b>Response</b>	<a href="#">createRequest</a>
<b>Operation faults</b>	None (see <a href="#">Faults</a> )

**CreateRequestRequest**

The CreateRequestRequest has the following request parameters:

Parameter	Type	Description	Required?
requestTypeID	<a href="#">xs:string</a>	Specifies the ID of the request type being created.	Y
userID	<a href="#">xs:string</a>	Specifies the userID of the person making the request.	N
<b>itemID</b> , <b>catalogKey</b> , and <b>callNumber</b> are a choice. You specify a value for only one.			
itemID	<a href="#">xs:long</a>	Specifies the itemID to use for the request, such as for a Place Academic Reserve or ILL Copy request. In order to use itemID, Symphony must have Ascension turned on. <b>Note:</b> If you specify an itemID, do not include a catalogKey or callNumber in the request.	N
catalogKey	<a href="#">xs:long</a>	Specifies the catalog key to use for the request, such as for a Place Academic Reserve or ILL Copy request. <b>Note:</b> If you specify a catalogKey, do not include an itemID or callNumber in the request.	N
callNumber	<a href="#">xs:string</a>	Specifies the call number of the item to use for the request, such as for a Place Academic Reserve or ILL Copy request. <b>Note:</b> If you specify a call number, do not include a catalogKey or itemID in the request.	N
serviceLibraryID	<a href="#">xs:string</a>	Specifies the libraryID of the service library.	N
statusID	<a href="#">xs:string</a>	Specifies the statusID to use for the request.	N
requestEntry	<a href="#">RequestEntry</a>	Specifies the entry for the request template. This information must first be retrieved from the <a href="#">lookupRequestTemplate</a> operation.	N

### CreateRequestResponse

The CreateRequestResponse is a string (see [xs:string](#)) that is the request key for a successfully created request. If required information is not supplied, invalid data is given in the request, or the user has exceeded the limit on requests, a fault is thrown instead.

### Related topics

- [Request service](#)

## lookupRequest

The lookupRequest operation allows you to retrieve detailed information about a single request.

This is different than the lookupRequests operation, which displays only brief information about all requests for a specific patron.

### Synopsis

<b>Service</b>	<a href="#">Request service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/request
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>Valid clientID (see <a href="#">clientID</a>)</li> <li>Valid sessionToken (see <a href="#">Authenticated operations</a>) from user with staff level or higher privileges</li> </ul>
<b>Request</b>	<a href="#">LookupRequestRequest</a>
<b>Response</b>	<a href="#">LookupRequestResponse</a>
<b>Operation faults</b>	None (see <a href="#">Faults</a> )

### LookupRequestRequest

The LookupRequestRequest has the following request parameters:

Parameter	Type	Description	Required?
requestKey	<a href="#">xs:string</a>	Specifies the request key to display detailed information for. The request key is available from the <a href="#">lookupRequests</a> operation.	N

### LookupRequestResponse

The LookupRequestResponse returns one or more RequestInfo types (see [RequestBriefInfo](#)). The RequestInfo type extends RequestBriefInfo (see [RequestBriefInfo](#) for a list of elements returned in this response).

If the specified request is not found, a fault is thrown instead.

### Related topics

- [Request service](#)

## lookupRequests

The lookupRequests operation allows you to retrieve a list of requests for a specific user.

This is different than the lookupRequest operation, which only displays information about a specific request.

### Synopsis

<b>Service</b>	<a href="#">Request service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/request

Service	<b>Request service</b>
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <b>clientID</b>)</li> <li>Valid sessionToken (see <b>Authenticated operations</b>) from user with staff level or higher privileges</li> </ul>
Request	<b>lookupRequests</b>
Response	<b>lookupRequests</b>
Operation faults	None (see <b>Faults</b> )

### LookupRequestsRequest

The LookupRequestsRequest has the following request parameters:

Parameter	Type	Description	Required?
Either userID or userAltID must be submitted as part of the request.			
userID	<b>xs:string</b>	Specifies the ID of the request type being created.	Y
userAltID	<b>xs:string</b>	Specifies the userID of the person making the request.	
requestStatusID	<b>xs:string</b>	Specifies to return only requests with the specified status ID.	N
serviceLibraryID	<b>xs:string</b>	Specifies to return only requests with the specified service library ID.	N
repliedToRequests	<b>xs:boolean</b>	Specifies whether to return only requests that have a reply ( <b>true</b> ) or all requests ( <b>false</b> ).	N

### LookupRequestsResponse

The LookupRequestsResponse returns one or more RequestBriefInfo types (see **RequestBriefInfo**). If the specified user does not have any matching requests, a fault is thrown instead.

### Related topics

- Request service**

## lookupRequestTemplate

The lookupRequestTemplate operation allows you to retrieve a Request Template for determining how to display the request, how to qualify entries, and how to populate a new request.

This is the only Request service operation that does not require a valid sessionToken from a user with staff level or higher privileges.

### Synopsis

Service	<b>Request service</b>
Namespace	http://schemas.sirsidynix.com/symws/request
Prerequisites	<ul style="list-style-type: none"> <li>Valid clientID (see <b>clientID</b>)</li> </ul>
Request	<b>LookupRequestTemplateRequest</b>
Response	<b>LookupRequestTemplateResponse</b>
Operation faults	None (see also <b>Faults</b> )

### LookupRequestTemplateRequest

The LookupRequestTemplateRequest has the following request parameter:

Parameter	Type	Description	Required?
requestTypeID	<b>xs:string</b>	Specifies the policy ID of the Request Template. You can get a list of templates and policy IDs through the <b>lookupRequestTypePolicyList</b> operation.	Y

### LookupRequestTemplateResponse

The LookupRequestTemplateResponse returns the following elements:

Element	Type	Occ.	Description
userIDRequired	<b>RequiredType</b>	1	Displays whether the user ID or alternate ID is optional, required, or ignored.
catalogKeyRequired	<b>RequiredType</b>	1	Displays whether the catalog key is optional, required, or ignored. The catalog key may be an item ID, call number, or catalog key.
defaultStatusID	<b>xs:string</b>	0..1	Displays the default status ID for the request.
defaultStatusDescription	<b>xs:string</b>	0..1	Displays the description of the default status for the request.
specialUse	<b>SpecialUseType</b>	1	Displays whether a request template is used or not.
requestEntryInfo	<b>RequestEntryTemplateInfo</b>	0..*	Displays additional information about each request entry for the template.



**Related topics**

- [Request service](#)
- [lookupRequestTypePolicyList](#) on page 160

## Circulation service

The Circulation service includes operations that help with circulation tasks.

The Circulation service includes these operations (listed alphabetically):

Operation	Description
<a href="#">lookupMailDeliveryEstimatesInfo</a>	Retrieves information about Mail Delivery Estimates.
<a href="#">recordPayments</a>	Pays one or more bills and returns the results.

**Related topics**

- [Services and Operations](#) on page 41

### lookupMailDeliveryEstimatesInfo

The lookupMailDeliveryEstimatesInfo operation allows you to retrieve Mail Delivery Estimates information for a user and one or more Delivery Service policies.

Use [isMailDeliveryEnabled](#) to determine if Mail Delivery has been enabled for the library.

**Synopsis**

Service	<a href="#">Circulation service</a>
Namespace	<a href="#">http://schemas.sirsiidynix.com/symws/circulation</a>
Prerequisites	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupMailDeliveryEstimatesInfoRequest</a>
Response	<a href="#">LookupMailDeliveryEstimatesInfoResponse</a>
Operation faults	None (see <a href="#">Faults</a> )

### LookupMailDeliveryEstimatesInfoRequest

The LookupMailDeliveryEstimatesInfoRequest has the following request parameters:

Parameter	Type	Description	Required?
userID	<b>xs:string</b>	Specifies the user ID to retrieve mail delivery estimates for. <b>Note:</b> If you do not specify a userID, the session user is used by default.	N
deliveryServiceID	<b>xs:string</b>	Specifies the ID of the delivery service to retrieve an estimate for. You can repeat the element to specify additional IDs.	Y

### LookupMailDeliveryEstimatesInfoResponse

The LookupMailDeliveryEstimatesInfoResponse returns the following elements:

Element	Type	Occ.	Description
maiDeliveryEstimateInfo	<b>MailDeliveryEstimatesInfo</b>	1..*	Displays information about the mail delivery estimate for each submitted delivery service.

### Related topics

- [Circulation service](#) on page 113
- [isMailDeliveryEnabled](#) on page 142

## recordPayments

The recordPayments operation allows you to record payments for multiple bills in a single call. Web Services attempts to record payments for all Credit Account payments before the monetary payments since bill reasons may preclude payment from a credit account for certain Bill Reason IDs.

Although it is possible to mix payments with applied credits, SirsiDynix recommends that you record them in separate requests.

### Synopsis

Service	<b>Circulation service</b>
Namespace	http://schemas.sirsiDynix.com/symws/circulation
Prerequisites	<ul style="list-style-type: none"> <li>• Valid clientID (see <a href="#">clientID</a>)</li> <li>• Valid sessionToken (see <a href="#">Authenticated operations</a>) from user whose user access includes Pay User Part A Command Code.</li> </ul>
Request	<b>recordPayments</b>
Response	<b>recordPayments</b>
Operation faults	None (see <a href="#">Faults</a> )

### RecordPaymentsRequest

TheRecordPaymentsRequest has the following request parameters:

Parameter	Type	Description	Required?
billUserID	<b>xs:string</b>	Specifies the user ID of the patron who the bill is for.	Y
billKey	<b>xs:int</b>	Specifies the bill number that was paid. You can include 1 or more billKey elements in the request. The billNumber is returned for each bill in <b>lookupMyAccountInfo</b> or <b>lookupPatronInfo</b> when includeFeeInfo is specified.	Y
feePayment	<b>FeePayment</b>	Specifies the entry or entries for the fee payments.	Y
selfPaid	<b>xs:boolean</b>	Specifies if the payment is made by a patron through an OPAC, Web Services application, or similar( <b>true</b> ) or by the patron directly to a staff member ( <b>false</b> ). <b>Note:</b> If you do not specify selfPaid, the system will assume "false" by default.	N
paidAtBillingLibrary	<b>xs:boolean</b>	Specifies whether the "paid at" library is also the billing library ( <b>true</b> ) or not ( <b>false</b> ). This is used when the client does not have a "Library" context and you want to avoid the problem of selecting a "paid at" library that is not allowed. <b>Note:</b> If you do not specify paidAtBillingLibrary, the system will assume "false" by default.	N

### RecordPaymentsResponse

The RecordPaymentsResponse returns the following elements:

Element	Type	Occ.	Description
feePaymentRecorded	<b>FeePaymentRecorded</b>	0..*	Displays a list of payments that were successfully recorded.
feePaymentProblem	<b>FeePaymentProblem</b>	0..*	Displays a list of payments that had problems, if any.
netAmountRecorded	<b>Money</b>	1	Displays the amount of money recorded as paid toward the bill or bills.
refundAmount	<b>Money</b>	0..1	Displays the amount of money paid toward a bill or bills that was not used.

**Related topics**

- [Circulation service](#) on page 113

## Admin service

The Admin service provides operations to access common ILS configuration data. The Admin service includes these operations (listed alphabetically):

Operation	Description
<a href="#">getAccessLevelPolicy</a>	Displays details for the specified User Access policy.
<a href="#">getAccessRestrictionPolicy</a>	Displays details for the specified Restriction policy, if the Accountability module is enabled.
<a href="#">getBestSellerPolicy</a>	Displays details for the specified Best Sellers policy.
<a href="#">getBillReasonPolicy</a>	Displays details for the specified Bill Reason policy.
<a href="#">getCitizenshipPolicy</a>	Displays details for the specified Citizenship policy, if the Accountability module is enabled.
<a href="#">getClearancePolicy</a>	Displays details for the specified Clearance policy, if the Accountability module is enabled.
<a href="#">getCreditReasonPolicy</a>	Displays details for the specified credit reason policy.
<a href="#">getGroupResponsibilityPolicy</a>	Displays details for the specified group responsibility policy.
<a href="#">getItemTypePolicy</a>	Displays details for the specified Item Type policy.
<a href="#">getLanguagePolicy</a>	Displays details for the specified Language policy.
<a href="#">getLibraryGroupPolicy</a>	Displays details for the specified Library Group policy.
<a href="#">getLibraryPolicy</a>	Displays details for the specified Library policy.
<a href="#">getLocationPolicy</a>	Displays details for the specified Location policy.
<a href="#">getMailServicePolicy</a>	Displays details for the specified Mail Service policy, if Books By Mail is enabled.
<a href="#">getNeedToKnowPolicy</a>	Displays details for the specified Need To Know policy, if the Accountability module is enabled.
<a href="#">getPaymentTypePolicy</a>	Displays details for the specified payment type policy.

Operation	Description
<b>getPolicy</b>	Displays details for the specified policy. You must supply a valid policy type code.
<b>getRecommendedReadingPolicy</b>	Displays details for the specified Recommended Reading policy.
<b>getRequestListPolicy</b>	Displays details for the specified Request List policy.
<b>getRequestStatusPolicy</b>	Displays details for the specified Request Status policy.
<b>getRequestTypePolicy</b>	Displays details for the specified Request Type policy.
<b>getSearchLibraryGroupPolicy</b>	Displays information about a specific Search Library Group policy.
<b>getUserGroupResponsibilityPolicy</b>	Displays information about the Responsibility Policy for the User Group.
<b>getUserLookupPolicy</b>	Displays details for the specified User Lookup policy.
<b>getUserSortPolicy</b>	Displays details for the specified User Sort policy.
<b>isCheckoutHistoryEnabled</b>	Indicates whether patron checkout history is enabled in Symphony (true) or disabled (false).
<b>isPatronSelfRegistrationEnabled</b>	Indicates whether Online User Registration is enabled (true) or disabled (false).
<b>isRestrictedAccess</b>	Indicates whether Always Require Authentication is enabled (true) or disabled (false).
<b>isUserGroupsEnabled</b>	Indicates whether User Groups are enabled (true) or not (false) in the ILS.
<b>lookupAccessLevelPolicyList</b>	Returns a list of User Access policies currently defined in the ILS.
<b>lookupAccessRestrictionPolicyList</b>	Returns a list of Restriction policies currently defined in the ILS, if the Accountability module is enabled.
<b>lookupBestSellersPolicyList</b>	Returns a list of Best Seller policies currently defined in the ILS.
<b>lookupBillReasonPolicyList</b>	Returns a list of Bill Reason policies currently defined in the ILS.

Operation	Description
<b>lookupCitizenshipPolicyList</b>	Returns a list of Citizenship policies currently defined in the ILS, if the Accountability module is enabled.
<b>lookupClearancePolicyList</b>	Returns a list of Clearance policies currently defined in the ILS, if the Accountability module is enabled.
<b>lookupCreditReasonPolicyList</b>	Returns a list of credit reason policies.
<b>lookupGroupResponsibilityPolicyList</b>	Returns a list of User Group Responsibility policies.
<b>lookupILSConfiguration</b>	For SirsiDynix internal use.
<b>lookupItemTypePolicyList</b>	Returns a list of Item Type policies currently defined in the ILS.
<b>lookupLanguagePolicyList</b>	Returns a list of Language policies currently defined in the ILS.
<b>lookupLibraryGroupList</b>	Returns a list of Library Group policies currently defined in the ILS.
<b>lookupLibraryPolicyList</b>	Returns a list of Library policies currently defined in the ILS.
<b>lookupLocationPolicyList</b>	Returns a list of Location policies currently defined in the ILS.
<b>lookupMailServicePolicyList</b>	Returns a list of Mail Service policies currently defined in the ILS, if Books By Mail is enabled.
<b>lookupNativeCurrency</b>	Returns the three-character ISO 4217 currency code mapped to the Currency policy that is designated as the ILS native currency (that is, the system default currency). If a Currency policy has not been mapped in the Web Services Admin, then the software returns the policy ID.
<b>lookupNeedToKnowPolicyList</b>	Returns a list of Need To Know policies currently defined in the ILS, if the Accountability module is enabled.
<b>lookupPaymentTypePolicyList</b>	Returns a list of payment type policies.
<b>lookupPolicyList</b>	Returns a list of policies defined for the specified policy type.

Operation	Description
<a href="#">lookupRecommendedReadingPolicyList</a>	Returns a list of Recommended Reading policies currently defined in the ILS.
<a href="#">lookupRequestListPolicyList</a>	Returns a list of Request List policies currently defined in the ILS.
<a href="#">lookupRequestStatusPolicyList</a>	Returns a list of Request Status policies currently defined in the ILS.
<a href="#">lookupRequestTypePolicyList</a>	Returns a list of Request Type policies currently defined in the ILS.
<a href="#">lookupSearchLibraryGroupList</a>	Returns a list of Search Library Group policies.
<a href="#">lookupUserLookupPolicyList</a>	Returns a list of User Lookup policies currently defined in the ILS.
<a href="#">lookupUserSortPolicyList</a>	Returns a list of User Sort policies currently defined in the ILS.

#### Related topics

- [Services and Operations](#) on page 41

## getAccessLevelPolicy

The `getAccessLevelPolicy` operation allows you to retrieve the access level policy.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="#">http://schemas.sirsiidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> <li>• <a href="#">policyID</a> (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetAccessLevelPolicyRequest</a>
Response	<a href="#">GetAccessLevelPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### GetAccessLevelPolicyRequest

The `GetAccessLevelPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid [policyID](#) (see [policyID](#)).

#### GetAccessLevelPolicyResponse

In addition to [PolicyInfo](#) (see [PolicyInfo](#)), the `GetAccessLevelPolicyResponse` returns the `AccessLevelPolicy` (see [AccessLevelPolicy](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [AccessLevelPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupAccessLevelPolicyList](#)

## getAccessRestrictionPolicy

The `getAccessRestrictionPolicy` operation allows you to retrieve policy information for access restriction.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/admin">http://schemas.sirsidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> <li>• <code>policyID</code> (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetAccessRestrictionPolicyRequest</a>
Response	<a href="#">GetAccessRestrictionPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### GetAccessRestrictionPolicyRequest

The `GetAccessRestrictionPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)).

#### GetAccessRestrictionPolicyResponse

The `GetAccessLevelPolicyResponse` returns `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).



**Related topics**

- [Admin service](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupAccessRestrictionPolicyList](#)

**getBestSellerPolicy**

The getBestSellerPolicy operation allows you to retrieve the Best Seller Policy Object.

**Synopsis**

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> <li>• policyID (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetBestSellerPolicyRequest</a>
Response	<a href="#">GetBestSellerPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

**GetBestSellerPolicyRequest**

The GetBestSellerPolicyRequest is a PolicyRequest (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid policyID (see [policyID](#)).

**GetBestSellerPolicyResponse**

In addition to PolicyInfo (see [PolicyInfo](#)), the GetBestSellerPolicyResponse returns the BestSellerPolicy (see [BestSellerPolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [BestSellerPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupBestSellersPolicyList](#)

**getBillReasonPolicy**

The getBillReasonPolicy operation allows you to retrieve the BillReason Policy Object.

**Synopsis**

<b>Service</b>	<b><a href="#">Admin service</a></b>
<b>Namespace</b>	<a href="http://schemas.sirsidynix.com/symws/admin">http://schemas.sirsidynix.com/symws/admin</a>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> <li>• <a href="#">policyID</a> (see <a href="#">policyID</a>)</li> </ul>
<b>Request</b>	<b><a href="#">GetBillReasonPolicyRequest</a></b>
<b>Response</b>	<b><a href="#">GetBillReasonPolicyResponse</a></b>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**GetBillReasonPolicyRequest**

The [GetBillReasonPolicyRequest](#) is a [PolicyRequest](#) (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid [policyID](#) (see [policyID](#)).

**GetBillReasonPolicyResponse**

In addition to [PolicyInfo](#) (see [PolicyInfo](#)), the [GetBillPaymentTypePolicyResponse](#) returns the [BillReasonPolicy](#) (see [BillReasonPolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [BillReasonPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupBillReasonPolicyList](#)

**getCitizenshipPolicy**

The [getCitizenshipPolicy](#) operation allows you to retrieve citizenship policy information.

**Synopsis**

<b>Service</b>	<b><a href="#">Admin service</a></b>
<b>Namespace</b>	<a href="http://schemas.sirsidynix.com/symws/admin">http://schemas.sirsidynix.com/symws/admin</a>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> <li>• <a href="#">policyID</a> (see <a href="#">policyID</a>)</li> </ul>

Service	<a href="#">Admin service</a>
Request	<a href="#">GetCitizenshipPolicyRequest</a>
Response	<a href="#">GetCitizenshipPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### GetCitizenshipPolicyRequest

The [GetCitizenshipPolicyRequest](#) is a [PolicyRequest](#) (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid [policyID](#) (see [policyID](#)).

### GetCitizenshipPolicyResponse

The [GetCitizenshipPolicyResponse](#) returns [PolicyInfo](#) (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#) on page 116
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupCitizenshipPolicyList](#)

## getClearancePolicy

The [getClearancePolicy](#) operation allows you to retrieve clearance policy information.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="#">http://schemas.sirsidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> <li>• <a href="#">policyID</a> (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetClearancePolicyRequest</a>
Response	<a href="#">GetClearancePolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### GetClearancePolicyRequest

The [GetClearancePolicyRequest](#) is a [PolicyRequest](#) (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid [policyID](#) (see [policyID](#)).

### GetClearancePolicyResponse

In addition to [PolicyInfo](#) (see [PolicyInfo](#)), the `GetClearancePolicyResponse` returns the `ClearancePolicy` (see [ClearancePolicy](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)

See "Admin service"

- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupClearancePolicyList](#)

## getCreditReasonPolicy

The `getCreditReasonPolicy` operation allows you to retrieve the Credit Reason Object.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="#">http://schemas.sirsidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> <li>• <code>policyID</code> (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">getCreditReasonPolicy</a>
Response	<a href="#">getCreditReasonPolicy</a>
Operation faults	None (see also <a href="#">Faults</a> )

### GetCreditReasonPolicyRequest

The `GetCreditReasonPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)).

### GetCreditReasonPolicyResponse

In addition to `PolicyInfo` (see [PolicyInfo](#)), the `GetCreditReasonPolicyResponse` returns the `CreditReasonPolicy` (see [CreditReasonPolicy](#) for a list of elements returned in this response).

Related topics

- [Admin service](#)
- [CreditReasonPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupCreditReasonPolicyList](#)

getGroupResponsibilityPolicy

The getGroupResponsibilityPolicy operation allows you to retrieve the Group Responsibility Object.

Synopsis

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"><li>• clientID (see <a href="#">clientID</a>)</li><li>• policyID (see <a href="#">policyID</a>)</li></ul>
Request	<a href="#">getGroupResponsibilityPolicy</a>
Response	<a href="#">getGroupResponsibilityPolicy</a>
Operation faults	None (see also <a href="#">Faults</a> )

GetGroupResponsibilityPolicyRequest

The GetGroupResponsibilityPolicyRequest is a PolicyRequest (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid policyID (see [policyID](#)).

GetGroupResponsibilityPolicyResponse

In addition to PolicyInfo (see [PolicyInfo](#)), the GetGroupResponsibilityPolicyResponse returns the GroupResponsibilityPolicy (see [GroupResponsibilityPolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [GroupResponsibilityPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupGroupResponsibilityPolicyList](#)

**getItemTypePolicy**

The getItemTypePolicy operation allows you to retrieve the ItemType Policy Object.

**Synopsis**

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> <li>• policyID (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetItemTypePolicyRequest</a>
Response	<a href="#">GetItemTypePolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

**GetItemTypePolicyRequest**

The GetItemTypePolicyRequest is a PolicyRequest (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid policyID (see [policyID](#)).

**GetItemTypePolicyResponse**

In addition to PolicyInfo (see [PolicyInfo](#)), the GetItemTypePolicyResponse returns the ItemTypePolicy (see [ItemTypePolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [ItemTypePolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupItemTypePolicyList](#)

**getLanguagePolicy**

The getLanguagePolicy operation allows you to retrieve the Language Policy Object.

**Synopsis**

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	<a href="http://schemas.sirsidynix.com/symws/admin">http://schemas.sirsidynix.com/symws/admin</a>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> <li>• <a href="#">policyID</a> (see <a href="#">policyID</a>)</li> </ul>
<b>Request</b>	<a href="#">GetLanguagePolicyRequest</a>
<b>Response</b>	<a href="#">GetLanguagePolicyResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**GetLanguagePolicyRequest**

The [GetLanguagePolicyRequest](#) is a [PolicyRequest](#) (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid [policyID](#) (see [policyID](#)).

**GetLanguagePolicyResponse**

In addition to [PolicyInfo](#) (see [PolicyInfo](#)), the [GetLanguagePolicyResponse](#) returns the [LanguagePolicy](#) (see [LanguagePolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [LanguagePolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupLanguagePolicyList](#)

**getLibraryGroupPolicy**

The [getLibraryGroupPolicy](#) operation allows you to retrieve information about a specific Library Group policy.

**Synopsis**

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	<a href="http://schemas.sirsidynix.com/symws/admin">http://schemas.sirsidynix.com/symws/admin</a>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> <li>• <a href="#">policyID</a> (see <a href="#">policyID</a>)</li> </ul>
<b>Request</b>	<a href="#">GetLibraryGroupPolicyRequest</a>

Service	<a href="#">Admin service</a>
Response	<a href="#">GetLibraryGroupPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### GetLibraryGroupPolicyRequest

The `GetLibraryGroupPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)) that references a Library Group.

### GetLibraryGroupPolicyResponse

In addition to `PolicyInfo` (see [PolicyInfo](#)), the `GetLibraryGroupPolicyResponse` returns the `LibraryGroupPolicy` (see [LibraryGroupPolicy](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [LibraryGroupPolicy](#) on page 173
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupLibraryGroupList](#) on page 152

## getLibraryPolicy

The `getLibraryPolicy` operation allows you to retrieve the Library Policy Object.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="http://schemas.sirsiidynix.com/symws/admin">http://schemas.sirsiidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> <li>• <code>policyID</code> (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetLibraryPolicyRequest</a>
Response	<a href="#">GetLibraryPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### GetLibraryPolicyRequest

The `GetLibraryPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)).

### GetLibraryPolicyResponse



In addition to [PolicyInfo](#) (see [PolicyInfo](#)), the `GetLibraryPolicyResponse` returns the `LibraryPolicy` (see [LibraryPolicy](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [LibraryPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupLibraryPolicyList](#)

## getLocationPolicy

The `getLocationPolicy` allows you to retrieve the Location Policy Object.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="#">http://schemas.sirsidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> <li>• <code>policyID</code> (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetLocationPolicyRequest</a>
Response	<a href="#">GetLocationPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### GetLocationPolicyRequest

The `GetLocationPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)).

#### GetLocationPolicyResponse

In addition to `PolicyInfo` (see [PolicyInfo](#)), the `GetLocationPolicyResponse` returns the `LocationPolicy` (see [LocationPolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [LocationPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupLocationPolicyList](#)

**getMailServicePolicy**

The getMailServicePolicy operation allows you to retrieve the Mail Service Policy Object.

**Synopsis**

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/admin
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> <li>• policyID (see <a href="#">policyID</a>)</li> </ul>
<b>Request</b>	<a href="#">GetMailServicePolicyRequest</a>
<b>Response</b>	<a href="#">GetMailServicePolicyResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**GetMailServicePolicyRequest**

The GetMailServicePolicyRequest is a PolicyRequest (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid policyID (see [policyID](#)).

**GetMailServicePolicyResponse**

In addition to PolicyInfo (see [PolicyInfo](#)), the GetMailServicePolicyResponse returns the MailServicePolicy (see [MailServicePolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [MailServicePolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupMailServicePolicyList](#)

**getNeedToKnowPolicy**

The getNeedToKnowPolicy operation allows you to retrieve Need to Know policy information.

**Synopsis**

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/admin
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> <li>• policyID (see <a href="#">policyID</a>)</li> </ul>
<b>Request</b>	<a href="#">GetNeedToKnowPolicyRequest</a>
<b>Response</b>	<a href="#">GetNeedToKnowPolicyResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**GetNeedToKnowPolicyRequest**

The GetNeedToKnowPolicyRequest is a PolicyRequest (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid policyID (see [policyID](#)).

**GetNeedToKnowPolicyResponse**

The GetCitizenshipPolicyResponse returns PolicyInfo (see [PolicyInfo](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupNeedToKnowPolicyList](#)

**getPaymentTypePolicy**

The getPaymentTypePolicy operation allows you to retrieve the Payment Type Object.

**Synopsis**

<b>Service</b>	<b><a href="#">Admin service</a></b>
<b>Namespace</b>	<a href="http://schemas.sirsidynix.com/symws/admin">http://schemas.sirsidynix.com/symws/admin</a>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <b><a href="#">clientID</a></b>)</li> <li>• <a href="#">policyID</a> (see <b><a href="#">policyID</a></b>)</li> </ul>
<b>Request</b>	<b><a href="#">GetPaymentTypePolicyRequest</a></b>
<b>Response</b>	<b><a href="#">GetPaymentTypePolicyResponse</a></b>
<b>Operation faults</b>	None (see also <b><a href="#">Faults</a></b> )

**GetPaymentTypePolicyRequest**

The [GetPaymentTypePolicyRequest](#) is a [PolicyRequest](#) (see **[PolicyRequest](#)**). This request has no request parameters, but it requires a valid [policyID](#) (see **[policyID](#)**).

**GetPaymentTypePolicyResponse**

In addition to [PolicyInfo](#) (see **[PolicyInfo](#)**), the [GetPaymentTypePolicyResponse](#) returns the [PaymentTypePolicy](#) (see **[PaymentTypePolicy](#)** for a list of elements returned in this response).

**Related topics**

- **[Admin service](#)**
- **[PaymentTypePolicy](#)**
- **[PolicyInfo](#)**
- **[PolicyRequest](#)**
- **[lookupPaymentTypePolicyList](#)**

**getPolicy**

The [getPolicy](#) operation allows you to retrieve information about a specific policy of a policy type.

**Synopsis**

<b>Service</b>	<b><a href="#">Admin service</a></b>
<b>Namespace</b>	<a href="http://schemas.sirsidynix.com/symws/admin">http://schemas.sirsidynix.com/symws/admin</a>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <b><a href="#">clientID</a></b>)</li> <li>• <a href="#">policyID</a> (see <b><a href="#">policyID</a></b>)</li> </ul>

Service	<a href="#">Admin service</a>
Request	<a href="#">GetPolicyRequest</a>
Response	<a href="#">GetPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### GetPolicyRequest

The GetPolicyRequest requires a valid policyID (see [policyID](#)) and has the following request parameters:

Parameter	Type	Description	Required?
policyType	<a href="#">xs:string</a>	Specifies the policy type for the policy you are attempting to get.	Y

### GetPolicyResponse

The GetPolicyResponse returns the following element:

Element	Type	Occ.	Description
policyField	<a href="#">PolicyField</a>	0..*	Displays information for the policy you specify in your request.

### Related topics

- [Admin service](#)
- [PolicyField](#)
- [lookupPolicyList](#)

## getRecommendedReadingPolicy

The getRecommendedReadingPolicy operation allows you to retrieve the Recommended Reading Policy Object.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="#">http://schemas.sirsidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> <li>• policyID (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetRecommendedReadingPolicyRequest</a>
Response	<a href="#">GetRecommendedReadingPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### GetRecommendedReadingPolicyRequest

The GetRecommendedReadingPolicyRequest is a PolicyRequest (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid policyID (see [policyID](#)).

### GetRecommendedReadingPolicyResponse

In addition to PolicyInfo (see [PolicyInfo](#)), the GetRecommendedReadingPolicyResponse returns the RecommendedReadingPolicy (see [RecommendedReadingPolicy](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [RecommendedReadingPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupRecommendedReadingPolicyList](#)

## getRequestListPolicy

The getRequestListPolicy operation allows you to retrieve the Request List Policy Object. The policy qualifies the Request types that can be made for a User Access type.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> <li>• policyID (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetRequestListPolicyRequest</a>
Response	<a href="#">GetRequestListPolicyResponse</a>
Operation faults	A fault is thrown when an invalid policy ID is used (see also <a href="#">Faults</a> )

### GetRequestListPolicyRequest

The GetRequestListPolicyRequest is a PolicyRequest (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid policyID (see [policyID](#)).

### GetRequestListPolicyResponse

In addition to [PolicyInfo](#) (see [PolicyInfo](#)), the `GetRequestListPolicyResponse` returns the `RequestListPolicy` (see [RequestListPolicy](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [RequestListPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupRequestListPolicyList](#)

## getRequestStatusPolicy

The `getRequestStatusPolicy` operation allows you to retrieve the Request Status Policy Object. The policy qualifies the different statuses that a request can be, as well as whether replies have a reply date automatically assigned.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/admin</code>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> <li>• <code>policyID</code> (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetRequestStatusPolicyRequest</a>
Response	<a href="#">GetRequestStatusPolicyResponse</a>
Operation faults	A fault is thrown when an invalid policy ID is used (see also <a href="#">Faults</a> )

#### GetRequestStatusPolicyRequest

The `GetRequestStatusPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)).

#### GetRequestStatusPolicyResponse

In addition to [PolicyInfo](#) (see [PolicyInfo](#)), the `GetRequestStatusPolicyResponse` returns the `RequestStatusPolicy` (see [RequestStatusPolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [RequestStatusPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupRequestStatusPolicyList](#)

**getRequestTypePolicy**

The `getRequestTypePolicy` operation allows you to retrieve the Request Type Policy Object. The policy qualifies the kinds of requests that can be made by users.

**Synopsis**

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	<code>http://schemas.sirsidynix.com/symws/admin</code>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> <li>• <code>policyID</code> (see <a href="#">policyID</a>)</li> </ul>
<b>Request</b>	<a href="#">GetRequestTypePolicyRequest</a>
<b>Response</b>	<a href="#">GetRequestTypePolicyResponse</a>
<b>Operation faults</b>	A fault is thrown when an invalid policy ID is used (see also <a href="#">Faults</a> )

**GetRequestTypePolicyRequest**

The `GetRequestTypePolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)).

**GetRequestTypePolicyResponse**

In addition to `PolicyInfo` (see [PolicyInfo](#)), the `GetRequestTypePolicyResponse` returns the `RequestTypePolicy` (see [RequestTypePolicy](#) for a list of elements returned in this response).



**Related topics**

- [Admin service](#)
- [RequestTypePolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupRequestTypePolicyList](#)

**getSearchLibraryGroupPolicy**

The getSearchLibraryGroupPolicy operation allows you to retrieve information about a specific Search Library Group policy.

**Synopsis**

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/admin
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> <li>• policyID (see <a href="#">policyID</a>)</li> </ul>
<b>Request</b>	<a href="#">GetSearchLibraryGroupPolicyRequest</a>
<b>Response</b>	<a href="#">GetSearchLibraryGroupPolicyResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**GetSearchLibraryGroupPolicyRequest**

The GetSearchLibraryGroupPolicyRequest is a PolicyRequest (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid policyID (see [policyID](#)) that references a Search Library Group.

**GetSearchLibraryGroupPolicyResponse**

In addition to PolicyInfo (see [PolicyInfo](#)), the GetSearchLibraryGroupPolicyResponse returns the SearchLibraryGroupPolicy (see [SearchLibraryGroupPolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [SearchLibraryGroupPolicy](#) on page 185
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupSearchLibraryGroupList](#) on page 160

**getUserGroupResponsibilityPolicy**

The `getUserGroupResponsibilityPolicy` operation allows you to retrieve information about a specific User Group's Responsibility policy.

**Synopsis**

Service	<a href="#">Admin service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/admin</code>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> <li>• <code>policyID</code> (see <a href="#">policyID</a>)</li> </ul>
Request	<a href="#">GetUserGroupResponsibilityPolicyRequest</a>
Response	<a href="#">GetUserGroupResponsibilityPolicyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

**GetUserGroupResponsibilityPolicyRequest**

The `GetUserGroupResponsibilityPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)).

**GetUserGroupResponsibilityPolicyResponse**

In addition to `PolicyInfo` (see [PolicyInfo](#)), the `GetUserGroupResponsibilityPolicyResponse` returns the `GroupResponsibilityPolicy` (see [GroupResponsibilityPolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [SearchLibraryGroupPolicy](#) on page 185
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupSearchLibraryGroupList](#) on page 160

**getUserLookupPolicy**

The `getUserLookupPolicy` operation allows you to retrieve the User Lookup Policy Object.

**Synopsis**

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	<code>http://schemas.sirsidynix.com/symws/admin</code>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> <li>• <code>policyID</code> (see <a href="#">policyID</a>)</li> </ul>
<b>Request</b>	<a href="#">GetUserLookupPolicyRequest</a>
<b>Response</b>	<a href="#">GetUserLookupPolicyResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**GetUserLookupPolicyRequest**

The `GetUserLookupPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)).

**GetUserLookupPolicyResponse**

In addition to `PolicyInfo` (see [PolicyInfo](#)), the `GetUserLookupPolicyResponse` returns the `UserLookupPolicy` (see [UserLookupPolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [UserLookupPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupUserLookupPolicyList](#)

**getUserSortPolicy**

The `getUserSortPolicy` operation allows you to retrieve the User Sort Policy Object.

**Synopsis**

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	<code>http://schemas.sirsidynix.com/symws/admin</code>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> <li>• <code>policyID</code> (see <a href="#">policyID</a>)</li> </ul>
<b>Request</b>	<a href="#">GetUserSortPolicyRequest</a>
<b>Response</b>	<a href="#">GetUserSortPolicyResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**GetUserSortPolicyRequest**

The `GetUserSortPolicyRequest` is a `PolicyRequest` (see [PolicyRequest](#)). This request has no request parameters, but it requires a valid `policyID` (see [policyID](#)).

**GetUserSortPolicyResponse**

In addition to `PolicyInfo` (see [PolicyInfo](#)), the `GetUserSortPolicyResponse` returns the `UserSortPolicy` (see [UserSortPolicy](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [UserSortPolicy](#)
- [PolicyInfo](#)
- [PolicyRequest](#)
- [lookupUserSortPolicyList](#)

**isCheckoutHistoryEnabled**

The `isCheckoutHistoryEnabled` operation allows you to determine whether or not patron circulation history has been enabled in Symphony.

Checkout history has three states in Symphony: NOHISTORY, CIRCRULE, and ALLCHARGES. Web Services treats the CIRCRULE state the same as ALLCHARGES. This means that the differences between CIRCRULE and ALLCHARGES are ignored, and only ALLCHARGES is used. Also, if a patron's state is first CIRCRULE, but then the patron disables the history (setting it to NOHISTORY), and later reenables it, the final setting will be ALLCHARGES (instead of CIRCRULE).

**Synopsis**

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	<code>http://schemas.sirsidynix.com/symws/admin</code>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">IsCheckoutHistoryEnabledRequest</a>
<b>Response</b>	<a href="#">IsCheckoutHistoryEnabledResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

**IsCheckoutHistoryEnabledRequest**

The `IsCheckoutHistoryEnabledRequest` has no parameters.

**IsCheckoutHistoryEnabledResponse**

The `IsCheckoutHistoryEnabledResponse` is boolean (see [xs:boolean](#)). The output will be **true** if checkout history is enabled (set to CIRCRULE or ALLCHARGES in Symphony), **false** if not.

**Related topics**

- [Admin service](#)

**isMailDeliveryEnabled**

The isMailDeliveryEnabled operation allows you to determine whether or not Mail Delivery has been enabled for the site.

**Synopsis**

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">IsMailDeliveryEnabledRequest</a>
Response	<a href="#">IsMailDeliveryEnabledResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

**IsMailDeliveryEnabledRequest**

The IsMailDeliveryEnabledRequest has no parameters.

**IsMailDeliveryEnabledResponse**

The IsMailDeliveryEnabledResponse is boolean (see [xs:boolean](#)). The output will be **true** if Mail Delivery is enabled, **false** if not.

**Related topics**

- [Admin service](#)

**isPatronSelfRegistrationEnabled**

The isPatronSelfRegistrationEnabled operation allows you to determine whether or not On Line User Registration has been enabled.

**Synopsis**

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">IsPatronSelfRegistrationEnabledRequest</a>

Service	<a href="#">Admin service</a>
Response	<a href="#">IsPatronSelfRegistrationEnabledResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### IsPatronSelfRegistrationEnabledRequest

The IsPatronSelfRegistrationEnabledRequest has no parameters.

### IsPatronSelfRegistrationEnabledResponse

The IsPatronSelfRegistrationEnabledResponse is boolean (see [xs:boolean](#)). The output will be **true** if patron self-registration is enabled, **false** if not.

### Related topics

- [Admin service](#)

## isRestrictedAccess

The isRestrictedAccess operation allows you to determine whether the **Always Require Authentication** option has been enabled in the Web Services Admin.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="#">http://schemas.sirsidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">IsRestrictedAccessRequest</a>
Response	<a href="#">IsRestrictedAccessResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### IsRestrictedAccessRequest

The IsRestrictedAccessRequest has no parameters. For REST requests, the clientID is also not required.

### IsRestrictedAccessResponse

The IsRestrictedAccessResponse is boolean (see [xs:boolean](#)). The output will be **true** if restricted access is enabled, **false** if not. When restricted access is enabled, most requests need a valid sessionToken (for more information about how it works, see [Privileged operations](#) on page 19).

**Related topics**

- [Admin service](#)
- [Privileged operations](#) on page 19

## isUserGroupsEnabled

The isUserGroupsEnabled operation allows you to determine whether or not User Groups have been enabled in Symphony.

**Synopsis**

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">IsUserGroupsEnabledResponse</a>
Response	<a href="#">IsUserGroupsEnabledRequest</a>
Operation faults	None (see also <a href="#">Faults</a> )

**IsUserGroupsEnabledRequest**

The IsUserGroupsEnabledRequest has no parameters.

**IsUserGroupsEnabledResponse**

The IsUserGroupsEnabledResponse is boolean (see [xs:boolean](#)). The output will be **true** if User Groups are enabled, **false** if not.

**Related topics**

- [Admin service](#)

## lookupAccessLevelPolicyList

The lookupAccessLevelPolicyList operation allows you to retrieve a list of Access Level policies.

**Synopsis**

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> </ul>



Service	<a href="#">Admin service</a>
Request	<a href="#">LookupAccessLevelPolicyListRequest</a>
Response	<a href="#">LookupAccessLevelPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupAccessLevelPolicyListRequest

The `LookupAccessLevelPolicyListRequest` has no request parameters.

### LookupAccessLevelPolicyListResponse

The `LookupAccessLevelPolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [getAccessLevelPolicy](#)
- [PolicyList](#)

## lookupAccessRestrictionPolicyList

The `lookupAccessRestrictionPolicyList` operation allows you to retrieve a list of Access Restriction policies.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/admin</code>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupAccessRestrictionPolicyListRequest</a>
Response	<a href="#">LookupAccessRestrictionPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupAccessRestrictionPolicyListRequest

The `LookupAccessRestrictionPolicyListRequest` has no request parameters.

### LookupAccessRestrictionPolicyListResponse

The `LookupAccessRestrictionPolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getAccessRestrictionPolicy](#)
- [PolicyList](#)

## lookupBestSellersPolicyList

The `lookupBestSellersPolicyList` operation allows you to retrieve a list of Best Seller policies.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/admin</code>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupBestSellersPolicyListRequest</a>
Response	<a href="#">LookupBestSellersPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### LookupBestSellersPolicyListRequest

The `LookupBestSellersPolicyListRequest` has no request parameters.

#### LookupBestSellersPolicyListResponse

The `LookupBestSellersPolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getBestSellerPolicy](#)
- [PolicyList](#)

## lookupBillReasonPolicyList

The `lookupBillReasonPolicyList` operation allows you to retrieve a list of Bill Reason policies.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupBillReasonPolicyListRequest</a>
Response	<a href="#">LookupBillReasonPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupBillReasonPolicyListRequest

The LookupBillReasonPolicyListRequest has no request parameters.

### LookupBillReasonPolicyListResponse

The LookupBillReasonPolicyListResponse returns a PolicyList (see [PolicyList](#)). The PolicyList contains PolicyInfo (see [PolicyInfo](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [getBillReasonPolicy](#)
- [PolicyList](#)

## lookupCitizenshipPolicyList

The lookupCitizenshipPolicyList operation allows you to retrieve a list of Citizenship policies.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupCitizenshipPolicyListRequest</a>
Response	<a href="#">LookupCitizenshipPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupCitizenshipPolicyListRequest

The LookupCitizenshipPolicyListRequest has no request parameters.

### LookupCitizenshipPolicyListResponse

The `LookupCitizenshipPolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getCitizenshipPolicy](#)
- [PolicyList](#)

## lookupClearancePolicyList

The `lookupClearancePolicyList` operation allows you to retrieve a list of Clearance policies.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/admin</code>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupClearancePolicyListRequest</a>
Response	<a href="#">LookupClearancePolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### LookupClearancePolicyListRequest

The `LookupClearancePolicyListRequest` has no request parameters.

#### LookupClearancePolicyListResponse

The `LookupClearancePolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getClearancePolicy](#)
- [PolicyList](#)

## lookupCreditReasonPolicyList

The `lookupCreditReasonPolicyList` operation allows you to retrieve a list of Credit Reason Policies that can be used with a Credit Account.

When recording credit account payments, you can only use the Credit Reason Policies that have a type of WITHDRAWAL. So, use the [getCreditReasonPolicy](#) to get the policy details to determine if the type is WITHDRAWAL.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">lookupCreditReasonPolicyList</a>
Response	<a href="#">lookupCreditReasonPolicyList</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupCreditReasonPolicyListRequest

The LookupCreditReasonPolicyListRequest has no request parameters.

### LookupCreditReasonPolicyListResponse

The LookupCreditReasonPolicyListResponse returns a PolicyList (see [PolicyList](#)). The PolicyList contains PolicyInfo (see [PolicyInfo](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [recordPayments](#)
- [PolicyList](#)

## lookupGroupResponsibilityPolicyList

The lookupGroupResponsibilityPolicyList operation allows you to retrieve a list of User Group Responsibility policies.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupGroupResponsibilityPolicyListRequest</a>
Response	<a href="#">LookupGroupResponsibilityPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

**LookupGroupResponsibilityPolicyListRequest**

The LookupGroupResponsibilityPolicyListRequest has no request parameters.

**LookupGroupResponsibilityPolicyListResponse**

The LookupGroupResponsibilityPolicyListResponse returns a PolicyList (see [PolicyList](#)). The PolicyList contains PolicyInfo (see [PolicyInfo](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [getUserLookupPolicy](#)
- [PolicyList](#)

**lookupILSConfiguration**

The lookupILSConfiguration operation is intended for SirsiDynix internal use.

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"><li>• clientID (see <a href="#">clientID</a>)</li></ul>
Request	<a href="#">lookupILSConfiguration</a>
Response	<a href="#">lookupILSConfiguration</a>
Operation faults	None (see also <a href="#">Faults</a> )

Parameter	Type	Description	Required?
configGroup	<a href="#">xs:string</a>	Specifies one of the following enumerated values: <ul style="list-style-type: none"><li>• <b>ALL</b> - Specify this value to return a list of all ILS configurations in the response. (If you do not specify a configGroup value, the system will assume ALL by default.)</li><li>• <b>MODULES</b> - Specify this value to return a list of only module configurations in the response.</li><li>• <b>SYSTEM</b> - Specify this value to return a list of only system configurations in the response.</li></ul>	N
configName	<a href="#">xs:string</a>	Specifies the name of the configuration you are looking up. <b>Note:</b> If you specify a configGroup, the entire group will display, regardless of whether you enter a configName parameter.	N

Element	Type	Occ.	Description
configInfo	<b>Element</b>	1..*	Displays information about your ILS configuration. The result is a list of <b>key/value</b> pairs that represent the data in the config file on the ILS server.

## lookupItemTypePolicyList

The lookupItemTypePolicyList operation allows you to retrieve a list of Item Type policies.

### Synopsis

Service	<b>Admin service</b>
Namespace	http://schemas.sirsiidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>clientID (see <b>clientID</b>)</li> </ul>
Request	<b>LookupItemTypePolicyListRequest</b>
Response	<b>LookupItemTypePolicyListResponse</b>
Operation faults	None (see also <b>Faults</b> )

### LookupItemTypePolicyListRequest

The LookupItemTypePolicyListRequest has no request parameters.

### LookupItemTypePolicyListResponse

The LookupItemTypePolicyListResponse returns a PolicyList (see **PolicyList**). The PolicyList contains PolicyInfo (see **PolicyInfo** for a list of elements returned in this response).

### Related topics

- **Admin service**
- **getItemTypePolicy**
- **PolicyList**

## lookupLanguagePolicyList

The lookupLanguagePolicyList operation allows you to retrieve a list of Language policies.

### Synopsis

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/admin
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupLanguagePolicyListRequest</a>
<b>Response</b>	<a href="#">LookupLanguagePolicyListResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

### LookupLanguagePolicyListRequest

The LookupLanguagePolicyListRequest has no request parameters.

### LookupLanguagePolicyListResponse

The LookupLanguagePolicyListResponse returns a PolicyList (see [PolicyList](#)). The PolicyList contains PolicyInfo (see [PolicyInfo](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [getLanguagePolicy](#)
- [PolicyList](#)

## lookupLibraryGroupList

The lookupLibraryGroupList operation allows you to retrieve a list of Library Group policies.

### Synopsis

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/admin
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupLibraryGroupListRequest</a>
<b>Response</b>	<a href="#">LookupLibraryGroupListResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

### LookupLibraryGroupListRequest

The LookupLibraryGroupListRequest has no request parameters.

### LookupLibraryGroupListResponse



The `LookupLibraryGroupListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getLibraryGroupPolicy](#)
- [PolicyList](#)

## lookupLibraryPolicyList

The `lookupLibraryPolicyList` operation allows you to retrieve a list of Library policies.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/admin</code>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupLibraryPolicyListRequest</a>
Response	<a href="#">LookupLibraryPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### LookupLibraryPolicyListRequest

The `LookupLibraryPolicyListRequest` has no request parameters.

#### LookupLibraryPolicyListResponse

The `LookupLibraryPolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getLibraryPolicy](#)
- [PolicyList](#)

## lookupLocationPolicyList

The `lookupLocationPolicyList` operation allows you to retrieve a list of Location policies.

#### Synopsis

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/admin
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupLocationPolicyListRequest</a>
<b>Response</b>	<a href="#">LookupLocationPolicyListResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

### LookupLocationPolicyListRequest

The LookupLocationPolicyListRequest has no request parameters.

### LookupLocationPolicyListResponse

The LookupLocationPolicyListResponse returns a PolicyList (see [PolicyList](#)). The PolicyList contains PolicyInfo (see [PolicyInfo](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [getLocationPolicy](#)
- [PolicyList](#)

## lookupMailServicePolicyList

The lookupMailServicePolicyList operation allows you to retrieve a list of Mail Service policies.

### Synopsis

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/admin
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupMailServicePolicyListRequest</a>
<b>Response</b>	<a href="#">LookupMailServicePolicyListResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

### LookupMailServicePolicyListRequest

The LookupMailServicePolicyListRequest has no request parameters.

### LookupMailServicePolicyListResponse

The `LookupMailServicePolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getMailServicePolicy](#)
- [PolicyList](#)

## lookupNativeCurrency

The `lookupNativeCurrency` operation allows you to retrieve the ILS's native currency code.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/admin</code>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupNativeCurrencyRequest</a>
Response	<a href="#">LookupNativeCurrencyResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### LookupNativeCurrencyRequest

The `LookupNativeCurrencyRequest` has no request parameters.

#### LookupNativeCurrencyResponse

The `LookupNativeCurrencyResponse` is a string (see [xs:string](#)). The output will be the library's native currency. If currencies have been mapped in the Admin, then the value returned in the response will be the mapped value.

#### Related topics

- [Admin service](#)

## lookupNeedToKnowPolicyList

The `lookupNeedToKnowPolicyList` operation allows you to retrieve a list of Need To Know policies.

#### Synopsis

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/admin
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupNeedToKnowPolicyListRequest</a>
<b>Response</b>	<a href="#">LookupNeedToKnowPolicyListResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

### LookupNeedToKnowPolicyListRequest

The LookupNeedToKnowPolicyListRequest has no request parameters.

### LookupNeedToKnowPolicyListResponse

The LookupNeedToKnowPolicyListResponse returns a PolicyList (see [PolicyList](#)). The PolicyList contains PolicyInfo (see [PolicyInfo](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [getNeedToKnowPolicy](#)
- [PolicyList](#)

## lookupPaymentTypePolicyList

The lookupPaymentTypePolicyList operation allows you to retrieve a list of Payment Type Policies that can be used when recording a payment.

### Synopsis

<b>Service</b>	<a href="#">Admin service</a>
<b>Namespace</b>	http://schemas.sirsidynix.com/symws/admin
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
<b>Request</b>	<a href="#">LookupPaymentTypePolicyListRequest</a>
<b>Response</b>	<a href="#">LookupPaymentTypePolicyListResponse</a>
<b>Operation faults</b>	None (see also <a href="#">Faults</a> )

### LookupPaymentTypePolicyListRequest

The LookupPaymentTypePolicyListRequest has no request parameters.

### LookupPaymentTypePolicyListResponse

The `LookupPaymentTypePolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [recordPayments](#)
- [PolicyList](#)
- [getPaymentTypePolicy](#) on page 131

## lookupPolicyList

The `lookupPolicyList` operation allows you to retrieve a list of policies for a specific policy type.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="#">http://schemas.sirsidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupPolicyListRequest</a>
Response	<a href="#">LookupPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### LookupPolicyListRequest

The `LookupPolicyListRequest` has the following request parameters:

Parameter	Type	Description	Required?
<code>policyType</code>	<a href="#">xs:string</a>	Specifies the policy type for the policy you are attempting to look up.	Y

#### LookupPolicyListResponse

The `LookupPolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [getPolicy](#)
- [PolicyList](#)

**lookupRecommendedReadingPolicyList**

The lookupRecommendedReadingPolicyList operation allows you to retrieve a list of Recommended Reading policies.

**Synopsis**

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• clientID (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupRecommendedReadingPolicyListRequest</a>
Response	<a href="#">LookupRecommendedReadingPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

**LookupRecommendedReadingPolicyListRequest**

The LookupRecommendedReadingPolicyListRequest has no request parameters.

**LookupRecommendedReadingPolicyListResponse**

The LookupRecommendedReadingPolicyListResponse returns a PolicyList (see [PolicyList](#)). The PolicyList contains PolicyInfo (see [PolicyInfo](#) for a list of elements returned in this response).

**Related topics**

- [Admin service](#)
- [getRecommendedReadingPolicy](#)
- [PolicyList](#)

**lookupRequestListPolicyList**

The lookupRequestListPolicyList operation allows you to retrieve a list of Request List policies.

**Synopsis**

Service	<a href="#">Admin service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/admin">http://schemas.sirsidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupRequestListPolicyListResponse</a>
Response	<a href="#">LookupRequestListPolicyListRequest</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupRequestListPolicyListRequest

The [LookupRequestListPolicyListRequest](#) has no request parameters.

### LookupRequestListPolicyListResponse

The [LookupRequestListPolicyListResponse](#) returns a [PolicyList](#) (see [PolicyList](#)). The [PolicyList](#) contains [PolicyInfo](#) (see [PolicyInfo](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [getRequestListPolicy](#)
- [PolicyList](#)

## lookupRequestStatusPolicyList

The [lookupRequestStatusPolicyList](#) operation allows you to retrieve a list of Request Status policies.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<a href="http://schemas.sirsidynix.com/symws/admin">http://schemas.sirsidynix.com/symws/admin</a>
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupRequestStatusPolicyListRequest</a>
Response	<a href="#">LookupRequestStatusPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupRequestStatusPolicyListRequest

The [LookupRequestStatusPolicyListRequest](#) has no request parameters.

### LookupRequestStatusPolicyListResponse

The `LookupRequestStatusPolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getRequestStatusPolicy](#)
- [PolicyList](#)

## lookupRequestTypePolicyList

The `lookupRequestTypePolicyList` operation allows you to retrieve a list of Request Type policies.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/admin</code>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupRequestTypePolicyListRequest</a>
Response	<a href="#">LookupRequestTypePolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### LookupRequestTypePolicyListRequest

The `LookupRequestTypePolicyListRequest` has no request parameters.

#### LookupRequestTypePolicyListResponse

The `LookupRequestTypePolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getRequestTypePolicy](#)
- [PolicyList](#)

## lookupSearchLibraryGroupList

The `lookupSearchLibraryGroupList` operation allows you to retrieve a list of Search Library Group policies.

#### Synopsis



Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupSearchLibraryGroupListRequest</a>
Response	<a href="#">LookupSearchLibraryGroupListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupSearchLibraryGroupListRequest

The `LookupSearchLibraryGroupListRequest` has no request parameters.

### LookupSearchLibraryGroupListResponse

The `LookupSearchLibraryGroupListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

### Related topics

- [Admin service](#)
- [getSearchLibraryGroupPolicy](#)
- [PolicyList](#)

## lookupUserLookupPolicyList

The `lookupUserLookupPolicyList` operation allows you to retrieve a list of User Lookup policies.

### Synopsis

Service	<a href="#">Admin service</a>
Namespace	http://schemas.sirsidynix.com/symws/admin
Prerequisites	<ul style="list-style-type: none"> <li>• <a href="#">clientID</a> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupUserLookupPolicyListRequest</a>
Response	<a href="#">LookupUserLookupPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

### LookupUserLookupPolicyListRequest

The `LookupUserLookupPolicyListRequest` has no request parameters.

### LookupUserLookupPolicyListResponse

The `LookupUserLookupPolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getUserLookupPolicy](#)
- [PolicyList](#)

## lookupUserSortPolicyList

The `lookupUserSortPolicyList` operation allows you to retrieve a list of User Sort policies.

#### Synopsis

Service	<a href="#">Admin service</a>
Namespace	<code>http://schemas.sirsidynix.com/symws/admin</code>
Prerequisites	<ul style="list-style-type: none"> <li>• <code>clientID</code> (see <a href="#">clientID</a>)</li> </ul>
Request	<a href="#">LookupUserSortPolicyListRequest</a>
Response	<a href="#">LookupUserSortPolicyListResponse</a>
Operation faults	None (see also <a href="#">Faults</a> )

#### LookupUserSortPolicyListRequest

The `LookupUserLookupPolicyListRequest` has no request parameters.

#### LookupUserSortPolicyListResponse

The `LookupUserLookupPolicyListResponse` returns a `PolicyList` (see [PolicyList](#)). The `PolicyList` contains `PolicyInfo` (see [PolicyInfo](#) for a list of elements returned in this response).

#### Related topics

- [Admin service](#)
- [getUserSortPolicy](#)
- [PolicyList](#)

# Chapter 5: Data types reference

This section contains information about the different data types that are used in the SirsiDynix Symphony Web Services 3.1 service operations.

The following topics are described in this section:

- [XSL Schema data types](#)
- [Web Services data types](#)

## XSL Schema data types

XSL Schemas contain the “xs” common namespace for an xml document that identifies the document as an XMLSchema. This common namespace has the following definition (found at the top of .xsd documents):

**xmlns:xs="http://www.w3.org/2001/XMLSchema"**

Any data type referenced in this chapter and in [Chapter Chapter 4, "Services and Operations" on page 41](#) that is preceded with “xs:” (such as **xs:string**) identifies the type as coming from the XMLSchema namespace.

This section explains these XSL Schema data types:

- **xs:boolean**
- **xs:date**
- **xs:dateTime**
- **xs:decimal**
- **xs:int**
- **xs:long**
- **xs:nonNegativeInteger**
- **xs:string**

### Related topics

- [Data types reference on page 163](#)

### xs:boolean

The xs:boolean type specifies one of two choices, true (“true” or “1”) or false (“false” or “0”). The values “true” and “false” are case-sensitive.

### Related topics

- [XSL Schema data types on page 163](#)
- [Data types reference on page 163](#)

## xs:date

The xs:date type represents a Gregorian calendar date in the format YYYY-MM-DD where *YYYY* is the four-digit year (0001-9999), *MM* is the month (01-12), *DD* is the day (01-31).

The XML standard allows for an optional “-” (minus sign) before the date to signify B.C. (For example, “-100” is equivalent to 100 B.C.)

SirsiDynix Symphony Web Services does not use time zone information for xs:date. Dates are adjusted for the time zone configured in the Admin tool.

For more information about this data type, see the XML Schema document at <http://www.w3.org/TR/xmlschema-2/>

### Related topics

- [XSL Schema data types on page 163](#)
- [Data types reference on page 163](#)

## xs:dateTime

The xs:dateTime type combines a Gregorian calendar date with a time in the format YYYY-MM-DDThh:mm:ss where *YYYY* is the year (0001-9999), *MM* is the month (01-12), *DD* is the day (01-31), *hh* is hours (00-23), *mm* is minutes (00-59) and *ss* is seconds (00-59). T divides the date and time parts.

The XML standard allows for an optional “-” (minus sign) before the date to signify B.C. (For example, “-100” is equivalent to 100 B.C.)

xs:dateTime data may also include time zone information relative to Coordinated Universal Time (UCT), in the format ±hh:mm, for example, 2007-04-23T09:43:23+06:00 or 2008-05-09T15:03:37Z (for 00:00, that is, no offset).

For more information about this data type, see the XML Schema document at <http://www.w3.org/TR/xmlschema-2/>

### Related topics

- [XSL Schema data types on page 163](#)
- [Data types reference on page 163](#)

## xs:decimal

The xs:decimal type represents a base-10 number with any number of digits to the right of a decimal point.

**Related topics**

- [XSL Schema data types](#) on page 163
- [Data types reference](#) on page 163

**xs:int**

The xs:int type is a 32-bit signed integer between -2147483648 and 2147483647.

**Related topics**

- [XSL Schema data types](#) on page 163
- [Data types reference](#) on page 163

**xs:long**

The xs:long type is a 64-bit integer between -9223372036854775808 and 9223372036854775807.

**Related topics**

- [XSL Schema data types](#) on page 163
- [Data types reference](#) on page 163

**xs:nonNegativeInteger**

The xs:nonNegativeInteger type includes zero or some integer greater than 0.

**Related topics**

- [XSL Schema data types](#) on page 163
- [Data types reference](#) on page 163

**xs:string**

The xs:string type represents a string of any valid XML characters, with special characters (such as <, >, and &) properly escaped to ensure well-formedness.

**Related topics**

- [XSL Schema data types](#) on page 163
- [Data types reference](#) on page 163

## Web Services data types

This section explains the SirsiDynix Symphony Web Services data types that are used in version 3.1 service operations.

For each data type, you will see the following information in a table:

Heading	Description
Element	Lists all the elements in the data type.
Type	Lists the data type of each element in the data type. These are links to topics that describe the data type in detail.
Occ.	Specifies the minimum and maximum number of times the element can occur (or be returned) in the data type. This is the format: <b>&lt;minimum&gt;..&lt;maximum&gt;.</b> For example, if the element occurred a minimum of 0 and was returned a maximum of 1, you would see "0..1". If the element always occurs once, but only once, you will see "1".
Description	Offers a brief description for each response element in the data type.

These data types are organized by service, and corresponding information can be found in the appropriate schema (.xsd) document and in this guide in [Appendix 7, "Service Schemas" on page 243](#).

- [Admin service](#)
- [Circulation service](#)
- [Common service](#)
- [Patron service](#)
- [Security service](#)
- [Standard service](#)

#### Related topics

- [Data types reference on page 163](#)

## Admin service

The Admin service has these data types:

- [AccessLevelPolicy](#)
- [BestSellerPolicy](#)
- [BillReasonPolicy](#)
- [ClearancePolicy](#)
- [CreditReasonPolicy](#)
- [Element](#)
- [ItemTypePolicy](#)
- [GroupResponsibilityPolicy](#)
- [LanguagePolicy](#)
- [LibraryGroupPolicy](#)
- [LibraryPolicy](#)

- [LocationPolicy](#)
- [MailServicePolicy](#)
- [PaymentTypePolicy](#)
- [PolicyField](#)
- [PolicyInfo](#)
- [PolicyList](#)
- [PolicyRequest](#)
- [RecommendedReadingPolicy](#)
- [RequestListPolicy](#)
- [RequestStatusPolicy](#)
- [RequestTypePolicy](#)
- [RequiredType](#)
- [SearchLibraryGroupPolicy](#)
- [SpecialUseType](#)
- [UserLookupPolicy](#)
- [UserSortPolicy](#)

### **AccessLevelPolicy**

The AccessLevelPolicy type displays information about the specified Access Level policy.

#### **Ancestors**

None

#### **Extends**

- [PolicyInfo](#)

#### **Relevant operations**

- [getAccessLevelPolicy](#)

#### **Contents**

This type has the following elements:

Element	Type	Occ.	Description
navigateAllowed	<a href="#">xs:boolean</a>	1	<p>Descriptions for the AccessLevelPolicy elements can be found by referring to the list of policy attributes under the “Access Level Wizard” topic in the Java WorkFlows help.</p> <p>Each of these elements will display <b>true</b> if the user is allowed to perform the respective task (navigate, search, etc.), and <b>false</b> if the user is not allowed.</p>
searchAllowed	<a href="#">xs:boolean</a>	1	
printAllowed	<a href="#">xs:boolean</a>	1	
downloadAllowed	<a href="#">xs:boolean</a>	1	
createAllowed	<a href="#">xs:boolean</a>	1	
modifyAllowed	<a href="#">xs:boolean</a>	1	
deleteAllowed	<a href="#">xs:boolean</a>	1	

### BestSellerPolicy

The BestSellerPolicy type displays information about the Best Seller policy.

#### Ancestors

None

#### Extends

- [PolicyInfo](#)

#### Relevant operations

- [getBestSellerPolicy](#)

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
bestSellerDisplay	<a href="#">xs:string</a>	0..1	Displays a comma-separated list of best seller lists that are configured to display for the specified Best Seller policy.

### BillReasonPolicy

The BillReasonPolicy type displays information about the specified Bill Reason policy.

#### Ancestors

None

#### Extends

- [PolicyInfo](#)



**Relevant operations**

- [getBillReasonPolicy](#)

**Contents**

This type has the following elements:

Element	Type	Occ.	Description
associatedItem	<a href="#">xs:boolean</a>	1	Indicates if the bill reason pertains to an item ( <b>true</b> ) or not ( <b>false</b> ).
billReasonType	<a href="#">xs:string</a>	0..1	Descriptions for the BillReasonPolicy elements can be found by referring to the list of policy attributes under the “Bill Reason Wizard” topic in the Java WorkFlows help.
taxRate	<a href="#">xs:string</a>	1	
taxBase	<a href="#">xs:string</a>	0..1	

**ClearancePolicy**

The ClearancePolicy type displays information about the specified Clearance policy.

**Ancestors**

None

**Extends**

- [PolicyInfo](#)

**Relevant operations**

- [getClearancePolicy](#)

**Contents**

This type has the following elements:

Element	Type	Occ.	Description
clearanceLevel	<a href="#">xs:int</a>	1	Descriptions for the ClearancePolicy elements can be found by referring to the list of policy attributes under the “Clearance Wizard” topic in the Java WorkFlows help.
purgePeriod	<a href="#">xs:int</a>	1	
chargeReceiptRequired	<a href="#">xs:boolean</a>	1	
certificateOfDestructionRequired	<a href="#">xs:boolean</a>	1	
classified	<a href="#">xs:boolean</a>	1	

## CreditReasonPolicy

The CreditReasonPolicy type displays information about the specified credit reason policy.

### Ancestors

None

### Extends

- [PolicyInfo](#)

### Relevant operations

- [getCreditReasonPolicy](#)

### Contents

The CreditReasonPolicy type has the following element.

Element	Type	Occ.	Description
type	<a href="#">xs:string</a>	1	Displays the type for the credit reason policy. For payments to work in Web Services, the type must be <b>WITHDRAWAL</b> .

Element	Type	Occ.	Description
name	<a href="#">xs:string</a>	1	Displays the name of the configuration (for example, "LOAN_HISTORY").
value	<a href="#">xs:string</a>	1	Displays a numeric value of <b>0</b> or <b>1</b> . "0" means that the configuration is not enabled on the ILS. "1" means that the configuration is enabled on the ILS. <b>Note:</b> Depending on the configuration setup, other numeric values can also be returned.

## ItemTypePolicy

The ItemTypePolicy type displays information about the specified Item Type policy.

### Ancestors

None

### Extends

- [PolicyInfo](#)

### Relevant operations

- [getItemTypePolicy](#)

### Contents

The ItemTypePolicyInfo type has the following elements:

Element	Type	Occ.	Description
floating	<a href="#">xs:boolean</a>	0..1	Descriptions for the ItemTypePolicy elements can be found by referring to the list of policy attributes under the “Item Type Wizard” topic in the Java WorkFlows help.
holdThreshold	<a href="#">xs:int</a>	0..1	
bookingProfileID	<a href="#">xs:string</a>	0..1	
reservationProfileID	<a href="#">xs:string</a>	0..1	
multiCirc	<a href="#">xs:boolean</a>	1	

## GroupResponsibilityPolicy

The GroupResponsibilityPolicy type displays information about the specified User Group's Responsibility policy.

### Ancestors

None

### Extends

- [PolicyInfo](#)

### Relevant operations

- [getGroupResponsibilityPolicy](#)
- [getUserGroupResponsibilityPolicy](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
responsibilityLevel	<a href="#">xs:string</a>	1	Displays the ID of the level of responsibility for delinquencies. The level can be either SELF or LINKED. The value is determined by the Responsibility Policy.
responsibilityType	<a href="#">xs:string</a>	1	Displays the type field used to indicate if transactions such as checkouts are handled differently. This value is determined by the Responsibility Policy.
allowedDisplayGroupCheckouts	<a href="#">xs:boolean</a>	1	Displays whether the user is allowed to see charges/checkouts for other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedDisplayGroupHolds	<a href="#">xs:boolean</a>	1	Displays whether the user can see holds that belong to other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedCheckoutGroupHolds	<a href="#">xs:boolean</a>	1	Displays whether the user can check out items on hold for other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedCancelGroupHolds	<a href="#">xs:boolean</a>	1	Displays whether the user can remove holds for other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedDisplayGroupFees	<a href="#">xs:boolean</a>	1	Displays whether the user can see bills that belong to other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedPayGroupFees	<a href="#">xs:boolean</a>	1	Displays whether the user can pay bills for other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedGroupChoice	<a href="#">xs:boolean</a>	1	Displays whether WorkFlows displays a prompt ( <b>true</b> ) or not ( <b>false</b> ) for Checkout, Place Hold, or Bill User that asks whether the transaction is for the group or individual. A value of true indicates that a prompt displays. A value of false indicates that a prompt does not display.

## LanguagePolicy

The LanguagePolicy type displays information about the specified Language policy.

### Ancestors

None

### Extends

- [PolicyInfo](#)

### Relevant operations

- [getLanguagePolicy](#)

### Contents

The LanguagePolicy type has the following elements:

Element	Type	Occ.	Description
marcLanguageCode	<a href="#">xs:string</a>	1	Descriptions for the LanguagePolicy elements can be found by referring to the list of policy attributes under the “Language Wizard” topic in the Java WorkFlows help.
defaultLanguage	<a href="#">xs:boolean</a>	1	
writeDirection	<a href="#">xs:string</a>	1	
directory	<a href="#">xs:string</a>	0..1	
locale	<a href="#">xs:string</a>	1	
article	<a href="#">xs:string</a>	0..*	
charset	<a href="#">xs:string</a>	0..1	

### LibraryGroupPolicy

The LibraryGroupPolicy type displays information about the specified Library Group policy.

#### Ancestors

None

#### Extends

- [PolicyInfo](#)

#### Relevant operations

- [getLibraryGroupPolicy](#)

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
libraryID	<a href="#">xs:string</a>	0..*	Displays the name of each library in the Library Group.
holdFill	<a href="#">xs:boolean</a>	1	See the description for the <b>Honor Order of Libraries in Group for Holds</b> element under the "Library Group Wizard" topic in the Java WorkFlows help. Displays <b>true</b> if the Ordered Hold Fill policy is used for selecting items to fill holds, or <b>false</b> if it is not.

### LibraryPolicy

The LibraryPolicy type displays information about the specified Library policy.

#### Ancestors

None

### Extends

- [PolicyInfo](#)

### Relevant operations

- [getLibraryPolicy](#)

### Contents

The LibraryPolicy type has the following elements.



Descriptions for the LibraryPolicy elements can be found by referring to the list of policy attributes under the “Library Wizard” topic in the Java WorkFlows help.

Element	Type	Occ.
userID	<a href="#">xs:string</a>	0..1
loanPeriod	<a href="#">xs:string</a>	0..1
accrueFine	<a href="#">xs:boolean</a>	1
closedDate	<a href="#">xs:date</a>	0..*
closedEndDate	<a href="#">xs:date</a>	0..*
holdLocationID	<a href="#">xs:string</a>	1
closedSunday	<a href="#">xs:boolean</a>	1
closedMonday	<a href="#">xs:boolean</a>	1
closedTuesday	<a href="#">xs:boolean</a>	1
closedWednesday	<a href="#">xs:boolean</a>	1
closedThursday	<a href="#">xs:boolean</a>	1
closedFriday	<a href="#">xs:boolean</a>	1
closedSaturday	<a href="#">xs:boolean</a>	1
textIndexName	<a href="#">xs:string</a>	0..1
NUCSymbol	<a href="#">xs:string</a>	0..1
acquisitionsDisplayLibraryID	<a href="#">xs:string</a>	0..*
acquisitionsMaintenanceLibraryID	<a href="#">xs:string</a>	0..*
holdPermissionLibraryGroupID	<a href="#">xs:string</a>	0..1
holdGroupLibraryGroupID	<a href="#">xs:string</a>	0..1
libraryID	<a href="#">xs:string</a>	0..1
serialDisplayLibraryID	<a href="#">xs:string</a>	0..*
serialMaintenanceLibraryID	<a href="#">xs:string</a>	0..*
circDisplayLibraryGroupID	<a href="#">xs:string</a>	0..1
circMaintenanceLibraryID	<a href="#">xs:string</a>	0..1
catDisplayLibraryGroupID	<a href="#">xs:string</a>	0..1

Element	Type	Occ.
catMaintenanceLibraryGroupID	<a href="#">xs:string</a>	0..1
userDisplayLibraryGroupID	<a href="#">xs:string</a>	0..1
userMaintenanceLibraryGroupID	<a href="#">xs:string</a>	0..1
billDisplayLibraryGroupID	<a href="#">xs:string</a>	0..1
billMaintenanceLibraryGroupID	<a href="#">xs:string</a>	0..1
holdAvailableLibraryID	<a href="#">xs:string</a>	0..*
SAN	<a href="#">xs:string</a>	0..1
closedDatesReport	<a href="#">xs:boolean</a>	1
libraryNumber	<a href="#">xs:int</a>	1
callNumberMaintenanceLibraryGroupID	<a href="#">xs:string</a>	0..1
itemMaintenanceLibraryGroupID	<a href="#">xs:string</a>	0..1
marcHoldingsMaintenanceLibraryGroupID	<a href="#">xs:string</a>	0..1
holdExpireDays	<a href="#">xs:int</a>	1
holdAvailableExpireDays	<a href="#">xs:int</a>	1
OCLCCode	<a href="#">xs:string</a>	0..1
libraryHourMaintenanceLibraryGroupID	<a href="#">xs:string</a>	0..1
openUrlResolver	<a href="#">xs:string</a>	0..1
patronCreditMaintenanceLibraryGroupID	<a href="#">xs:string</a>	0..1
mailCenterLibraryID	<a href="#">xs:string</a>	0..1
mailCenterLocationID	<a href="#">xs:string</a>	0..1
mailHoldLibraryGroupID	<a href="#">xs:string</a>	0..1
floatingLibraryGroupID	<a href="#">xs:string</a>	0..1

### LocationPolicy

The LocationPolicy type displays information about the specified Location policy.

#### Ancestors

None



**Extends**

- [PolicyInfo](#)

**Relevant operations**

- [getLocationPolicy](#)

**Contents**

This type has the following elements:

Element	Type	Occ.	Description
holdable	<a href="#">xs:boolean</a>	1	Descriptions for the LocationPolicy elements can be found by referring to the list of policy attributes under the “Location Wizard” topic in the Java WorkFlows help.
type	<a href="#">xs:string</a>	1	
shadow	<a href="#">xs:boolean</a>	1	
reshelvingPeriod	<a href="#">xs:string</a>	0..1	
reshelvingLocation	<a href="#">xs:string</a>	0..1	
available	<a href="#">xs:boolean</a>	1	

**MailServicePolicy**

The MailServicePolicy type displays information about the specified Mail Service policy.

**Ancestors**

None

**Extends**

- [PolicyInfo](#)

**Relevant operations**

- [getMailServicePolicy](#)

**Contents**

The MailServicePolicy type has the following elements:

Element	Type	Occ.	Description
shipTime	<a href="#">xs:int</a>	0..1	Descriptions for this MailServicePolicy element can be found by referring to the list of policy attributes under the “Mail Service Wizard” topic in the Java WorkFlows help.

## PaymentTypePolicy

The PaymentTypePolicy type displays information about the specified Payment Type policy.

### Ancestors

None

### Extends

- [PolicyInfo](#)

### Relevant operations

- [getPaymentTypePolicy](#)

### Contents

The PaymentTypePolicy type has the following element.

Element	Type	Occ.	Description
kind	<a href="#">xs:string</a>	1	Displays information about the kind of Payment Type (for example, "CASH", "CHECK", "CREDIT CARD"). For more information, see the "Payment Type Wizard" topic in the Java WorkFlows help.

## PolicyField

The PolicyField type displays the defined value of a policy's data codes.

### Ancestors

None

### Relevant operations

- [getPolicy](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
policyFieldCode	<a href="#">xs:string</a>	1	Displays the policy's data code (for example, "ue" for the Bill Payment Type policy).
policyFieldValue	<a href="#">xs:string</a>	1	Displays the value for the policy's data code.

## PolicyInfo

The PolicyInfo type includes the basic information fields that are common to all policies. PolicyInfo is returned with the PolicyList data type (see [PolicyList](#)) for all “lookup...PolicyList” operation responses. It is also returned in the response for all “get...Policy” operations.

### Ancestors

- [PolicyList](#)

### Relevant operations

- [getAccessLevelPolicy](#)
- [getAccessRestrictionPolicy](#)
- [getBestSellerPolicy](#)
- [getBillReasonPolicy](#)
- [getCitizenshipPolicy](#)
- [getClearancePolicy](#)
- [getCreditReasonPolicy](#)
- [getItemTypePolicy](#)
- [getLanguagePolicy](#)
- [getLibraryGroupPolicy](#)
- [getLibraryPolicy](#)
- [getLocationPolicy](#)
- [getMailServicePolicy](#)
- [getNeedToKnowPolicy](#)
- [getPaymentTypePolicy](#)
- [getPolicy](#)
- [getRecommendedReadingPolicy](#)
- [getRequestListPolicy](#)
- [getRequestStatusPolicy](#)
- [getRequestTypePolicy](#)
- [getSearchLibraryGroupPolicy](#)
- [getUserGroupResponsibilityPolicy](#)
- [getUserLookupPolicy](#)
- [getUserSortPolicy](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
policyNumber	<a href="#">xs:int</a>	1	Displays the policy number.
policyID	<a href="#">xs:string</a>	1	Displays the policy ID.
policyDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the policy.

## PolicyList

The PolicyList type displays policy information (see [PolicyInfo](#)) about the policy you specify in your request. This data type will be returned in the response for all the “lookup...PolicyList” operations.

### Ancestors

None

### Relevant operations

- [lookupAccessLevelPolicyList](#)
- [lookupAccessRestrictionPolicyList](#)
- [lookupBestSellersPolicyList](#)
- [lookupBillReasonPolicyList](#)
- [lookupCitizenshipPolicyList](#)
- [lookupClearancePolicyList](#)
- [lookupCreditReasonPolicyList](#)
- [lookupGroupResponsibilityPolicyList](#)
- [lookupItemTypePolicyList](#)
- [lookupLanguagePolicyList](#)
- [lookupLibraryGroupList](#)
- [lookupLibraryPolicyList](#)
- [lookupLocationPolicyList](#)
- [lookupMailServicePolicyList](#)
- [lookupNeedToKnowPolicyList](#)
- [lookupPaymentTypePolicyList](#)
- [lookupRecommendedReadingPolicyList](#)
- [lookupRequestListPolicyList](#)
- [lookupRequestStatusPolicyList](#)
- [lookupRequestTypePolicyList](#)
- [lookupSearchLibraryGroupList](#)
- [lookupUserLookupPolicyList](#)

- [lookupUserSortPolicyList](#)
- [getAccessLevelPolicy](#)
- [getAccessRestrictionPolicy](#)
- [getBillReasonPolicy](#)
- [getCitizenshipPolicy](#)
- [getClearancePolicy](#)
- [getItemTypePolicy](#)
- [getLanguagePolicy](#)
- [getLibraryGroupPolicy](#)
- [getLibraryPolicy](#)
- [getLocationPolicy](#)
- [getMailServicePolicy](#)
- [getNeedToKnowPolicy](#)
- [getPaymentTypePolicy](#)
- [getPolicy](#)
- [getRequestTypePolicy](#)
- [getSearchLibraryGroupPolicy](#)
- [getUserGroupResponsibilityPolicy](#)
- [getUserLookupPolicy](#)
- [getUserSortPolicy](#)

## Contents

This type has the following elements:

Element	Type	Occ.	Description
policyInfo	<a href="#">PolicyInfo</a>	0..1	Displays information about the policy.

## PolicyRequest

The PolicyRequest type is used in all “get...Policy” operation requests. It simply requires that when using any “get...Policy” operations, you include a policyID in the operation request.

## Ancestors

None

## Relevant operations

The PolicyRequest type is used in the following operation requests:

- [getAccessLevelPolicy](#)

- [getAccessRestrictionPolicy](#)
- [getBestSellerPolicy](#)
- [getBillReasonPolicy](#)
- [getCitizenshipPolicy](#)
- [getClearancePolicy](#)
- [getCreditReasonPolicy](#)
- [getGroupResponsibilityPolicy](#)
- [getItemTypePolicy](#)
- [getLanguagePolicy](#)
- [getLibraryGroupPolicy](#)
- [getLibraryPolicy](#)
- [getLocationPolicy](#)
- [getMailServicePolicy](#)
- [getNeedToKnowPolicy](#)
- [getPaymentTypePolicy](#)
- [getRecommendedReadingPolicy](#)
- [getRequestListPolicy](#)
- [getRequestStatusPolicy](#)
- [getRequestTypePolicy](#)
- [getSearchLibraryGroupPolicy](#)
- [getUserGroupResponsibilityPolicy](#)
- [getUserLookupPolicy](#)
- [getUserSortPolicy](#)

### Contents

This type has the following request parameter. You must use this parameter in the requests of each of the operations listed above.

Parameter	Type	Description	Required?
policyID	<a href="#">xs:string</a>	Specifies the ID of the policy you are attempting to get in the call.	Y

### RecommendedReadingPolicy

The RecommendedReadingPolicy type displays information about the Recommended Reading policy.

### Ancestors

None

### Extends

- [PolicyInfo](#)

#### Relevant operations

- [getRecommendedReadingPolicy](#)

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
recommendedReadingDisplay	<a href="#">xs:string</a>	0..1	Displays a comma-separated list of recommended reading lists that are configured to display for the specified Recommended Reading policy.

### RequestListPolicy

The RequestListPolicy type displays information about the specified Request List policy.

#### Ancestors

None

#### Extends

- [PolicyInfo](#)

#### Relevant operations

- [getRequestListPolicy](#)

#### Contents

The RequestListPolicy type has the following elements:

Element	Type	Occ.	Description
requestTypeID	<a href="#">xs:string</a>	0..*	Specifies the IDs of the allowed request types for the list.

### RequestStatusPolicy

The RequestStatusPolicy type displays information about the specified Request Status policy.

#### Ancestors

None

#### Extends

- [PolicyInfo](#)

#### Relevant operations

- [getRequestStatusPolicy](#)

### Contents

The RequestStatusPolicy type has the following element:

Element	Type	Occ.	Description
autoUpdateReplyDate	<a href="#">xs:boolean</a>	1	Specifies whether replies have a reply date automatically assigned ( <b>true</b> ) or not ( <b>false</b> ).

### RequestTypePolicy

The RequestTypePolicy type displays information about the specified Request Type policy.

### Ancestors

None

### Extends

- [PolicyInfo](#)

### Relevant operations

- [getRequestTypePolicy](#)

### Contents

The RequestTypePolicy type has the following elements;

Element	Type	Occ.	Description
requestFormatID	<a href="#">xs:string</a>	1	Descriptions for these RequestTypePolicy elements can be found by referring to the list of policy attributes under the “Request Type Wizard” topic in the Java WorkFlows help.
responseFormatID	<a href="#">xs:string</a>	1	
userIDRequired	<a href="#">RequiredType</a>	1	
titleKeyRequired	<a href="#">RequiredType</a>	1	
requestTextRequired	<a href="#">RequiredType</a>	1	
defaultStatusID	<a href="#">xs:string</a>	1	
specialUse	<a href="#">SpecialUseType</a>	1	

### RequiredType

The RequiredType type displays an enumeration value for specific elements in the RequestTypePolicy.

### Ancestors



- [RequestTypePolicy](#)

#### Relevant operations

- [getRequestTypePolicy](#)

#### Contents

This type is a string (see [xs:string](#)) with one of the following enumeration values:

- **MANDATORY** - Use this value to specify that the element must be supplied. If it is not, Web Services will throw a fault.
- **OPTIONAL** - Use this value to specify that the element may or may not be supplied. If it is not, no fault is thrown. If it is, it is used.
- **NEVER** - Use this value to specify that the element will be ignored if supplied.

### SearchLibraryGroupPolicy

The SearchLibraryGroupPolicy type displays information about the specified Search Library Group policy.

#### Ancestors

None

#### Extends

- [PolicyInfo](#)

#### Relevant operations

- [getSearchLibraryGroupPolicy](#)

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
libraryID	<a href="#">xs:string</a>	0..*	Displays the name of each library in the Search Library Group.

### SpecialUseType

The SpecialUseType type displays an enumeration value for the RequestTypePolicy or lookupRequestTemplate operation.

#### Ancestors

- [RequestTypePolicy](#)

#### Relevant operations

- [getRequestTypePolicy](#)
- [lookupRequestTemplate](#)

### Contents

This type is a string (see [xs:string](#)) with one of the following enumeration values:

- **NONE** - Indicates that the request type policy or template should use an associated request format template.
- **HOLD** - Do not use, this value has been deprecated.
- **MESSAGE** - Indicates that the request type policy or template should not use a request template.
- **ENHANCED\_HOLD** - Do not use, this value has been deprecated.

### UserLookupPolicy

The UserLookupPolicy type displays information about the specified User Lookup policy.

#### Ancestors

None

#### Extends

- [PolicyInfo](#)

#### Relevant operations

- [getUserLookupPolicy](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
searchID	<a href="#">xs:string</a>	1	Displays the user lookup search ID.

### UserSortPolicy

The UserSortPolicy type displays information about the specified User Sort policy.

#### Ancestors

None

#### Extends

- [PolicyInfo](#)

#### Relevant operations

- [getUserSortPolicy](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
searchID	<a href="#">xs:string</a>	1	Displays the user sort search ID.

## Circulation service

The Circulation service has these data types:

- [CheckoutHistoryRule](#)
- [CheckoutInfoFilter](#)
- [FeeInfoFilter](#)
- [FeePayment](#) on page 189
- [FeePaymentProblem](#)
- [FeePaymentRecorded](#)
- [HoldInactiveType](#)
- [HoldInfoFilter](#)
- [HoldRange](#)
- [HoldType](#)
- [MailDeliveryEstimatesInfo](#)
- [StatusType](#)

### CheckoutHistoryRule

The CheckoutHistoryRule type displays information about how charge history is kept for a specific user. For more information, see the “Basic Info Tab: Users” topic in the Java WorkFlows help.

#### Ancestors

- [PatronCirculationInfo](#)

#### Relevant operations

- [lookupMyAccountInfo](#)

#### Contents

This type is a string (see [xs:string](#)) with the following enumeration values:

- **NOHISTORY**
- **CIRCRULE**
- **ALLCHARGES**

## CheckoutInfoFilter

The CheckoutInfoFilter type contains enumeration values that you can use in the [lookupMyAccountInfo](#) operation request to limit your results when you retrieve checkout information.

### Ancestors

None

### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

### Contents

This type is a string (see [xs:string](#)) with the following enumeration values:

- ALL
- ALLEXCEPTLOST
- ALLEXCEPTCLAIMEDRETURNED
- CLAIMEDRETURNED
- LOST
- OVERDUE
- RENEWED
- RECALLED

## FeeInfoFilter

The FeeInfoFilter type contains enumeration values that you can use in the [lookupMyAccountInfo](#) operation request to limit your results when you retrieve fee information.

### Ancestors

None

### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

### Contents

This type is a string (see [xs:string](#)) with the following enumeration values:

- UNPAID\_FEES
- UNPAID\_FEES\_AND\_PAYMENTS
- PAID\_FEES

- **PAID\_FEES\_AND\_PAYMENTS**
- **ALL\_FEES**
- **ALL\_FEES\_AND\_PAYMENTS**

## FeePayment

The FeePayment type specifies details for recording the payment of a fee.

### Ancestors

None

### Relevant operations

- [recordPayments](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
amount	<a href="#">Money</a>	1	Specifies the amount that was recorded as paid.
paymentTypeID	<a href="#">xs:string</a>	1	Specifies the payment type ID. <b>Note:</b> Use the <a href="#">lookupPaymentTypePolicyList</a> or <a href="#">getPaymentTypePolicy</a> operation to retrieve the ID for and information about Payment Type policies.
vendorTransactionNumber	<a href="#">xs:string</a>	0..1	Specifies the transaction number to record with the payment. This number was supplied by the vendor (such as PayPal) at the time the payment was made.
creditReasonID	<a href="#">xs:string</a>	0..1	Specifies the ID of the Credit Reason policy. If the payment type ID is <b>CREDITACCT</b> , creditReasonID is required. Otherwise, if the payment type ID is something else, the system ignores any value submitted in this element. <b>Note:</b> Use <a href="#">lookupCreditReasonPolicyList</a> to get the list of valid creditReasonIDs.

## FeePaymentProblem

The FeePaymentProblem type displays information about unsuccessful recording of fee payments.

### Ancestors

None

### Relevant operations

- [recordPayments](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
billKey	<a href="#">xs:int</a>	1	Displays the billKey that had a problem.
problem	<a href="#">xs:string</a>	1	Displays details about the reason for the problem.

## FeePaymentRecorded

The FeePaymentRecorded type displays information about successful fee payments.

### Ancestors

None

### Relevant operations

- [recordPayments](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
billKey	<a href="#">xs:int</a>	1	Displays the billKey that was recorded as partially or fully paid.
paymentTypeID	<a href="#">xs:string</a>	1	Displays the payment type ID. <b>Note:</b> Use the <a href="#">lookupPaymentTypePolicyList</a> or <a href="#">getPaymentTypePolicy</a> operation to retrieve the ID for and information about Payment Type policies.
paymentTypeDescription	<a href="#">xs:string</a>	0..1	Displays a description of the payment type.
amountPaid	<a href="#">Money</a>	1	Displays the amount recorded that was applied to the bill.
amountRemaining	<a href="#">Money</a>	0..1	Displays the amount remaining on the bill after payment.

## HoldInactiveType

The HoldInactiveType type contains enumeration values that you can use in the [lookupMyAccountInfo](#) operation request to limit your results when you retrieve inactive hold information.

### Ancestors

- [PatronHoldInfo](#)

### Relevant operations

- [lookupMyAccountInfo](#)

### Contents

This type is a string (see [xs:string](#)) with the following enumeration values:

- **HOLD\_INACTIVE\_REASON\_NONE**
- **HOLD\_INACTIVE\_REASON\_FILLED**
- **HOLD\_INACTIVE\_REASON\_CANCELLED**
- **HOLD\_INACTIVE\_REASON\_EXPIRED**
- **HOLD\_INACTIVE\_REASON\_CANCELLED\_ON\_RESERVE**
- **HOLD\_INACTIVE\_REASON\_BLANKET\_NONEED**
- **HOLD\_INACTIVE\_REASON\_CANCELLED\_ON\_ORDER**
- **HOLD\_INACTIVE\_REASON\_MODIFIED\_ORDER**
- **HOLD\_INACTIVE\_REASON\_EXP\_ONSHELF**

### HoldInfoFilter

The HoldInfoFilter type contains enumeration values that you can use in the [lookupMyAccountInfo](#) operation request to limit your results when you retrieve hold information.

#### Ancestors

None

#### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

#### Contents

This type is a string (see [xs:string](#)) with the following enumeration values:

- **ALL**
- **AVAILABLE**
- **COUNT\_ONLY**
- **ACTIVE**
- **INACTIVE**
- **RECALLED**
- **RESERVE**
- **UNAVAILABLE**

### HoldRange

The HoldRange type values correspond to the Symphony server “Hold Range” enumeration.



**HoldRange** only applies when the **HoldType** is “TITLE” (for more information, see [HoldType](#)).

#### Ancestors

None

#### Relevant operations

- [createMyHold](#)

#### Contents

This type is a string (see [xs:string](#)) with the following enumeration values:

- **LIBRARY**
- **GROUP**
- **SYSTEM**

#### HoldType

The **HoldType** type values correspond to the Symphony server “Hold Type” enumeration.

#### Ancestors

None

#### Relevant operations

- [createMyHold](#)

#### Contents

This type is a string (see [xs:string](#)) with the following enumeration values:

- **COPY**
- **TITLE**
- **CALL**
- **SYSTEM**

#### MailDeliveryEstimatesInfo

The **MailDeliveryEstimatesInfo** type displays information about the delivery estimate for a mail delivery type.

#### Ancestors

None

#### Relevant operations

- [lookupMailDeliveryEstimatesInfo](#)



## Contents

This type has the following elements:

Element	Type	Occ.	Description
deliveryServiceID	<a href="#">xs:string</a>	1	Displays the ID of the delivery service
deliveryServiceDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the delivery service.
estimatedShippingFee	<a href="#">Money</a>	0..1	Displays the estimated cost for the delivery type for the specified user.
estimatedShippingTime	<a href="#">xs:int</a>	0..1	Displays the estimated number of days for shipping using the delivery type.
valid	<a href="#">xs:boolean</a>	1	Displays <b>false</b> if there is no shipping cost mapping for the delivery service associated with the user's library and profile settings, or if there was no match for the delivery service ID. Otherwise, displays <b>true</b> .

## StatusType

The StatusType type displays the user's status.

### Ancestors

- [PatronStatusInfo](#)

### Relevant operations

- [lookupMyAccountInfo](#)

### Contents

This type is a string (see [xs:string](#)) with the following enumeration values:

- **BARRED**
- **BLOCKED**
- **DELINQUENT**
- **OK**

## Common service

The Common service has these data types:

- [AddressInfo](#)
- [Entity](#)
- [EntityOperation](#)
- [Money](#)
- [NillableDate](#)

## AddressInfo

The AddressInfo type displays patron-specific address information for each of the addressInfo elements in the PatronAddressInfo data type.

### Ancestors

- [PatronAddressInfo](#)

### Relevant operations

- [lookupMyAccountInfo](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
addressPolicyID	<a href="#">xs:string</a>	1	Displays the address policy ID.
addressPolicyDescription	<a href="#">xs:string</a>	1	Displays a text description of the address policy.
addressValue	<a href="#">xs:string</a>	0..1	Displays information related to the address policy. For example, policy "STREET" may have an addressValue of "34 South Main". Policy "ZIP" may have an addressValue of 84604.

## Entity

The Entity type is a persistent object that will be saved in the database.



SirsiDynix Engineering defines both the object and the individual elements within an object that can be persisted as entities.

### Ancestors

- [MyList](#)
- [MyListEntry](#)

### Relevant operations

- [getMyList](#)
- [lookupMyLists](#)
- [saveMyList](#)

### Contents

This type consists of the following:

Attribute	Type
operation	<a href="#">EntityOperation</a>

## EntityOperation

The EntityOperation type provides persistence instructions to the web service specifying what operation to do on a specific entity.

### Ancestors

- [Entity](#)

### Relevant operations

- [getMyList](#)
- [lookupMyLists](#)
- [saveMyList](#)

### Contents

EntityOperation contains the following valid enumerations for the **operation** attribute of the Entity data type (see [Entity](#)).

- "NOOP"
- "CREATE"
- "UPDATE"
- "DELETE"

## Money

The Money type is a decimal value with a required attribute specifying the ISO-4217 value (for example, "USD").

### Ancestors

- [FeeInfo](#)
- [FeePaymentInfo](#)
- [GroupMemberInfo](#)
- [PatronCheckoutInfo](#)
- [PatronCirculationInfo](#)

### Relevant operations

- [lookupMyAccountInfo](#)

### Contents

The Money data type consists of the following:

Attribute	Type	Use
<a href="#">currency</a>	<a href="#">xs:string</a>	required

## NillableDate

The NillableDate type is used in the [createMyHold](#) and [modifyMyHold](#) operations to specify the hold expiresDate. You can specify a value that will override what is existing, or you can specify to nullify the existing date entirely.

### Ancestors

None

### Relevant operations

- [createMyHold](#)
- [modifyMyHold](#)

### Contents

The NillableDate type has the following elements:

Element	Type	Description
value	<a href="#">xs:date</a>	Specifies the expiration date value that will be used for the hold (will override any existing date). <b>Note:</b> These two elements are a choice. If you specify a new date for this" element, do not enter a "nil" value. If there is a value, then nil will not be displayed.
nil	<a href="#">xs:boolean</a>	Specifies if you want to nullify the existing date ( <b>true</b> ), or not ( <b>false</b> ). <b>Note:</b> These two elements are a choice. Only specify a value for this element if you do not specify a new date in the "value" element.

## Patron service

The Patron service has these data types:

- [FeelInfo](#)
- [FeelItemInfo](#)
- [FeePaymentInfo](#)
- [GroupInfo](#) on page 200
- [GroupMemberCheckoutInfo](#)
- [GroupMemberFeeInfo](#)
- [GroupMemberHoldInfo](#)
- [GroupMemberInfo](#)
- [MyList](#)
- [MyListEntry](#)
- [MyListEntryInfo](#)
- [MyListInfo](#)

- [MyListTitleInfo](#)
- [PatronAddressInfo](#)
- [PatronCheckoutHistoryInfo](#)
- [PatronCheckoutInfo](#)
- [PatronCirculationInfo](#)
- [PatronHoldInfo](#)
- [PatronInfo](#)
- [PatronStatusInfo](#)
- [UserSuspensionInfo](#) on page 222

### **FeeInfo**

The FeeInfo type displays information about a particular patron's fees and bills.

This type may also be extended by the [GroupMemberFeeInfo](#) type.

#### **Ancestors**

None

#### **Relevant operations**

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

#### **Contents**

This type has the following elements. (For more information, see “Bills Tab” in the Java WorkFlows help.)

Element	Type	Occ.	Description
billNumber	<b>xs:int</b>	1	Displays a unique number that identifies the bill.
billReasonID	<b>xs:string</b>	1	Displays the ID for the Bill Reason policy. (For more info see “Bill Reason Wizard” in the Java WorkFlows help.)
billReasonDescription	<b>xs:string</b>	0..1	Displays a text description of the Bill Reason policy. (For more information, see “Bill Reason Wizard” in the Java WorkFlows help.)
amount	<b>Money</b>	1	Displays the total monetary amount billed for an item or group of items.
taxAmount	<b>Money</b>	0..1	Displays the monetary amount of tax added to a bill.
amountOutstanding	<b>Money</b>	1	Displays the portion of amount + taxAmount that is not yet paid.
billLibraryID	<b>xs:string</b>	0..1	Displays the Library ID for the library that created the bill. <b>Note:</b> If <code>config.single</code> is configured, the default station library ID is returned.
billLibraryDescription	<b>xs:string</b>	0..1	Displays a text description of the bill library.
dateBilled	<b>xs:date</b>	1	Displays the date an item was billed to the patron.
dateCreated	<b>xs:date</b>	0..1	Displays the date the bill was recorded on the database.
numberOfPaymentsMade	<b>xs:int</b>	1	Displays the number of payments a patron has made on a bill (can be up to 50).
lastPaymentDate	<b>xs:date</b>	0..1	Displays the date the last payment was made on the bill. <b>Note:</b> <code>lastPayment</code> date is only returned if <code>numberOfPaymentsMade</code> is greater than 0 (zero).
feePaymentInfo	<b>FeePaymentInfo</b>	0..50	Displays fee payment information. <b>Note:</b> <code>feePaymentInfo</code> is only returned if <code>includeFeeInfo</code> includes “AND_PAYMENTS”.
feeItemInfo	<b>FeeItemInfo</b>	0..1	Displays fee information if the fee is associated with an item. <code>feeItemInfo</code> is only returned if the fee is associated with an item (for example, an overdue fine).

### FeeItemInfo

The `FeeItemInfo` type displays any fee information that is associated with an item (for example, an overdue book).

### Ancestors

- [FeeInfo](#)

### Relevant operations

- [lookupMyAccountInfo](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
titleKey	<a href="#">xs:long</a>	1	Displays a database key to the title record.
itemID	<a href="#">xs:string</a>	0..1	Displays the itemID. <b>Note:</b> itemID is returned only if the library is configured for accession.
itemTypeDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the item type. itemTypeDescription is returned only if the library is configured for accession.
callNumber	<a href="#">xs:string</a>	1	Displays the items call number.
copyNumber	<a href="#">xs:int</a>	0..1	Displays which copy of the item if multiple copies exist.
itemLibraryID	<a href="#">xs:string</a>	1	Displays the Library ID of the library the item belongs to.
itemLibraryDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the library the item belongs to.
title	<a href="#">xs:string</a>	0..1	Displays the item's title.
author	<a href="#">xs:string</a>	0..1	Displays the item's author.
checkoutDate	<a href="#">xs:dateTime</a>	0..1	Displays the date the item was checked out.
dueDate	<a href="#">xs:dateTime</a>	0..1	Displays the item's due date.
recallDate	<a href="#">xs:date</a>	0..1	Displays the date a recall notice was sent.
renewedDate	<a href="#">xs:dateTime</a>	0..1	Displays the date the item was renewed.
circulationRuleID	<a href="#">xs:string</a>	0..1	Displays the ID of the circulation rule that was applied to the item.
circulationRuleDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the circulation rule that was applied to the item.

### FeePaymentInfo

The FeePaymentInfo type displays information about payments that a patron has paid on fees and fines.



FeePaymentInfo is only displayed if the includeFeeInfo parameter of the lookupMyAccountInfo operation request includes “AND\_PAYMENTS”.

#### Ancestors

- [FeeInfo](#)

#### Relevant operations

- [lookupMyAccountInfo](#)

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
transactionID	<a href="#">xs:string</a>	1	Displays the ID of the payment transaction.
paymentLibraryID	<a href="#">xs:string</a>	0..1	Displays the ID of the library where payment was made.
paymentLibraryDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the library where payment was made.
paymentDate	<a href="#">xs:date</a>	0..1	Displays the date that payment was made.
paymentAmount	<a href="#">Money</a>	1	Displays the monetary amount that was paid.
paymentTypeID	<a href="#">xs:string</a>	1	Displays the ID (code) of the payment type. (For more information, see “Payment Type Wizard” in the Java WorkFlows help.
paymentTypeDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the payment type.

#### GroupInfo

The GroupInfo type displays information about a particular user group.

#### Ancestors

None

#### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

#### Contents

This type has the following elements:



Element	Type	Occ.	Description
groupName	<b>xs:long</b>	1	Displays the name of the user group.
numberOfGroupMembers	<b>xs:int</b>	1	Displays the number of users in the user group.
dateGroupCreated	<b>xs:date</b>	1	Displays the date the group was created.
groupStatusID	<b>xs:string</b>	0..1	Displays the status ID for the entire User Group.
groupStatusDescription	<b>xs:string</b>	0..1	Displays a description about the status for the entire User Group.
groupStatusMessage	<b>xs:string</b>	0..1	Displays a message about the status for the entire User Group.
userGroupResponsibilityPolicyID	<b>xs:string</b>	1	Displays the ID of the Responsibility Policy.
userGroupResponsibilityPolicyDescription	<b>xs:string</b>	1	Displays the description of the Responsibility Policy.
userGroupResponsibilityLevelID	<b>xs:string</b>	1	Displays the ID of the level of responsibility for delinquencies. The level can be either SELF or LINKED. The value is determined by the Responsibility Policy.
userGroupResponsibilityTypeID	<b>xs:string</b>	1	Displays the type field used to indicate if transactions such as checkouts are handled differently. This value is determined by the Responsibility Policy.
noticeMaster	<b>xs:boolean</b>	1	Displays whether all notices for the group are sent to this user ( <b>true</b> ) or not ( <b>false</b> ). There can only be one Notice Master per group.
allowedDisplayGroupCheckouts	<b>xs:boolean</b>	1	Displays whether the user is allowed to see charges for other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedDisplayGroupHolds	<b>xs:boolean</b>	1	Displays whether the user can see holds that belong to other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedCheckoutGroupHolds	<b>xs:string</b>	1	Displays whether the user can check out items on hold for other members of the group ( <b>true</b> ) or not ( <b>false</b> ).

Element	Type	Occ.	Description
allowedCancelGroupHolds	<b>xs:boolean</b>	1	Displays whether the user can remove holds for other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedDisplayGroupFees	<b>xs:boolean</b>	1	Displays whether the user can see bills that belong to other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedPayGroupFees	<b>xs:boolean</b>	1	Displays whether the user can pay bills for other members of the group ( <b>true</b> ) or not ( <b>false</b> ).
allowedGroupChoice	<b>xs:boolean</b>	1	Displays whether WorkFlows displays a prompt ( <b>true</b> ) or not ( <b>false</b> ) for Checkout, Place Hold, or Bill User that asks whether the transaction is for the group or individual. A value of true indicates that a prompt displays. A value of false indicates that a prompt does not display.
groupMemberCheckoutInfo	<b>GroupMemberCheckoutInfo</b>	0..*	Displays checkout information for the group member if <b>includePatronCheckoutInfo</b> is set to <b>true</b> and the user has appropriate permissions.
groupMemberHoldInfo	<b>GroupMemberHoldInfo</b>	0..*	Displays hold information for the group member if <b>includePatronHoldInfo</b> is set to <b>true</b> and the user has appropriate permissions.
groupMemberFeeInfo	<b>GroupMemberFeeInfo</b>	0..*	Displays fee and fine information for the group member if <b>includeFeeInfo</b> is set to <b>true</b> and the user has appropriate permissions.
groupMemberInfo	<b>GroupMemberInfo</b>	0..*	Displays information about the group members.

## GroupMemberCheckoutInfo

The GroupMemberCheckoutInfo type displays checkout information for each User Group member.

### Extends

- [PatronCheckoutInfo](#)

### Ancestors

- **GroupInfo**

**Relevant operations**

- [lookupMyAccountInfo](#)

**Contents**

This type has the following elements:

Element	Type	Occ.	Description
userID	<a href="#">xs:string</a>	1	Displays the ID of the User Group member.
userName	<a href="#">xs:string</a>	1	Displays the name of the User Group member.

**GroupMemberFeeInfo**

The GroupMemberFeeInfo type displays fine and fee information for each User Group member.

**Extends**

- [FeeInfo](#)

**Ancestors**

- [GroupInfo](#)

**Relevant operations**

- [lookupMyAccountInfo](#)

**Contents**

This type has the following elements:

Element	Type	Occ.	Description
userID	<a href="#">xs:string</a>	1	Displays the ID of the User Group member.
userName	<a href="#">xs:string</a>	1	Displays the name of the User Group member.

**GroupMemberHoldInfo**

The GroupMemberHoldInfo type displays hold information for each User Group member.

**Extends**

- [PatronHoldInfo](#)

**Ancestors**

- [GroupInfo](#)

**Relevant operations**

- [lookupMyAccountInfo](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
userID	<a href="#">xs:string</a>	1	Displays the ID of the User Group member.
userName	<a href="#">xs:string</a>	1	Displays the name of the User Group member.

### GroupMemberInfo

The GroupMemberInfo type displays information for each User Group member.

### Ancestors

- [GroupInfo](#)

### Relevant operations

- [lookupMyAccountInfo](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
userID	<b>xs:string</b>	1	Displays the ID of the User Group member.
userName	<b>xs:string</b>	1	Displays the name of the User Group member.
userGroupResponsibilityPolicyID	<b>xs:string</b>	0..1	Displays the ID of the Responsibility Policy for the group member.
userGroupResponsibilityPolicyDescription	<b>xs:string</b>	0..1	Displays the description of the Responsibility Policy for the group member.
noticeMaster	<b>xs:boolean</b>	1	Displays whether all notices for the group are sent to this group member ( <b>true</b> ) or not ( <b>false</b> ). There can only be one Notice Master per group.
userStatusID	<b>xs:string</b>	0..1	Displays the status ID for the group member, such as DELINQUENT or OK.
userStatusDescription	<b>xs:string</b>	0..1	Displays a description of the status of the group member.
userStatusMessage	<b>xs:string</b>	0..1	Displays a message relating to the status of the group member.
userNumberOfBills	<b>xs:int</b>	0..1	Displays the number of outstanding bills for the group member.
userEstimatedFines	<b>Money</b>	0..1	Displays the current estimated fines for the group member.
userNumberOfOverdues	<b>xs:int</b>	0..1	Displays the number of overdue items for the group member.

## MyList

The MyList data type is the first Entity object. It is used for CRUD operations dealing with MyList. MyList extends “Entity” which dictates which operation is to be performed on the MyList and MyListEntry elements. (For more information, see [Entity](#).)

### Ancestors

None

### Relevant operations

- [getMyList](#)
- [lookupMyLists](#)
- [saveMyList](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
myListID	<a href="#">xs:string</a>	0..1	Displays an ID that is generated when a new MyList is created.
description	<a href="#">xs:string</a>	0..1	Displays a text description that the patron specifies to describe the MyList.
activeList	<a href="#">xs:boolean</a>	0..1	Indicates whether the MyList is active ( <b>true</b> ), or not ( <b>false</b> ). <b>Note:</b> Only one MyList can be active at a given time.
myListEntry	<a href="#">MyListEntry</a>	0..*	Extends “Entity” (see <a href="#">Entity</a> ). <b>Note:</b> Only “CREATE”, “DELETE”, and “NOOP” are valid entity operations.

### MyListEntry

MyListEntry extends “Entity” (see [Entity](#)); however, only “CREATE”, “DELETE”, and “NOOP” are valid entity operations. “UPDATE” is treated as “NOOP”.

#### Ancestors

- [MyList](#)

#### Relevant operations

- [getMyList](#)
- [lookupMyLists](#)
- [saveMyList](#)

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
titleKey	<a href="#">xs:long</a>	1	Displays the key to the title record for the particular entry. <b>Note:</b> The “UPDATE” entity operation is treated as “NOOP” for the MyListEntry data type.

### MyListEntryInfo

The MyListEntryInfo type displays MyList information that is not part of the entity object (such as title information).

#### Ancestors

- [MyListInfo](#)

#### Relevant operations

- [lookupMyList](#)

## Contents

MyListEntryInfo has the following elements:

Element	Type	Occ.	Description
titleKey	<a href="#">xs:long</a>	1	Displays the key to the title record for the particular entry.
myListTitleInfo	<a href="#">MyListTitleInfo</a>	0..1	Displays title information about a MyListEntry object (not an “Entity”).

## MyListInfo

The MyListInfo type displays information about a particular patron’s MyList.

### Ancestors

None

### Relevant operations

- [lookupMyList](#)

## Contents

This type has the following elements:

Element	Type	Occ.	Description
myListID	<a href="#">xs:string</a>	1	Displays the ID of the MyList.
description	<a href="#">xs:string</a>	1	Displays a text description of the MyList.
activeList	<a href="#">xs:boolean</a>	1	Indicates whether the MyList is active ( <b>true</b> ) or not ( <b>false</b> ).
myListEntryInfo	<a href="#">MyListEntryInfo</a>	0..*	Displays MyList information that is not part of the entity object.

## MyListTitleInfo

The MyListTitleInfo type displays MyList title information that is not part of the entity object.

### Ancestors

- [MyListEntryInfo](#) (that is a child type of [MyListInfo](#))

### Relevant operations

- [lookupMyList](#)

## Contents

MyListTitleInfo has the following elements:

Element	Type	Occ.	Description
author	<a href="#">xs:string</a>	0..1	Displays the item's author.
title	<a href="#">xs:string</a>	1	Displays the item's title.
thirdBriefEntry	<a href="#">xs:string</a>	0..1	Displays whatever generic information was assigned to field 3 on the Symphony server.
fourthBriefEntry	<a href="#">xs:string</a>	0..1	Displays whatever generic information was assigned to field 4 on the Symphony server.

### PatronAddressInfo

The PatronAddressInfo type displays address information for a particular patron.

#### Ancestors

None

#### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
primaryAddress	<a href="#">xs:int</a>	0..1	Displays the position of the primary address ( <b>1, 2, or 3</b> ).
Address1Info	<a href="#">AddressInfo</a>	0..*	Displays the first AddressInfo.
Address2Info	<a href="#">AddressInfo</a>	0..*	Displays the second AddressInfo.
Address3Info	<a href="#">AddressInfo</a>	0..*	Displays the third AddressInfo.

### PatronCheckoutHistoryInfo

The PatronCheckoutHistoryInfo type displays specific checkout (charge) history information for a particular patron.

#### Ancestors

None

#### Relevant operations

- [lookupMyAccountInfo](#)

#### Contents



This type has the following elements:

Element	Type	Occ.	Description
itemID	<a href="#">xs:string</a>	1	Displays the item ID.
callNumber	<a href="#">xs:string</a>	1	Displays the item call number.
displayableCallNumber	<a href="#">xs:string</a>	1	Displays the formatted call number.
copyNumber	<a href="#">xs:int</a>	1	Displays which copy of the item if multiple copies exist.
title	<a href="#">xs:string</a>	0..1	Displays the title of the item.
author	<a href="#">xs:string</a>	0..1	Displays the item's author.
checkoutLibraryID	<a href="#">xs:string</a>	0..1	Displays the library where the item was checked out. This element is not returned in a single-library configuration.
checkoutLibraryDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the library where the item was checked out. This element is not returned in a single-library configuration.
itemLibraryID	<a href="#">xs:string</a>	0..1	Displays the item's owning library Library ID. This element is not returned in a single-library configuration.
itemLibraryDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the item's owning library. This element is not returned in a single-library configuration.
itemTypeID	<a href="#">xs:string</a>	0..1	Displays the item's item type ID.
itemTypeDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the item's item type ID.
reserveCollectionID	<a href="#">xs:string</a>	0..1	Displays the reserve collection ID. This element is only returned if the item is on academic reserve.
reserveCollectionDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the reserve collection. This element is only returned if the item is on academic reserve.
checkoutDate	<a href="#">xs:dateTime</a>	1	Displays the date the item was checked out.
dueDate	<a href="#">xs:dateTime</a>	1	Displays the item's due date.

Element	Type	Occ.	Description
circulationRuleID	<b>xs:string</b>	0..1	Displays the the policy name of the primary Circulation Rule policy for the item.
circulationRuleDescription	<b>xs:string</b>	0..1	Displays the description of the circulation rule for the item.
recallDate	<b>xs:date</b>	0..1	Displays the date a recall notice was sent. This element is returned only if the checkout has been recalled.
recallDueDate	<b>xs:dateTime</b>	0..1	Displays the item due date after it is recalled.
recallNoticesSent	<b>xs:int</b>	0..1	Displays the number of recall notices that have been sent.
lastRenewedDate	<b>xs:dateTime</b>	0..1	Displays the date the item was last renewed.
renewals	<b>xs:int</b>	1	Displays the number of times the item has been renewed.
wasOverdue	<b>xs:boolean</b>	1	Indicates whether the item is overdue ( <b>true</b> ) or not ( <b>false</b> ).
overdueNoticesSent	<b>xs:int</b>	1	Displays the number of overdue notices the system has sent for an item.
dateOfLastOverdueNotice	<b>xs:date</b>	0..1	Displays the date that the last overdue notice was sent. This element is returned only if the setting is not "NEVER".
fine	<b>Money</b>	0..1	Displays the monetary amount of fine associated with the item. This amount includes both any outstanding bills on the checkout and the estimated fine that would accrue if the item was returned at the time of the operation request. This element is only returned if the fine is greater than 0.
dateLoanHistoryDischarged	<b>xs:dateTime</b>	0..1	Displays the date and time on which the item was most recently discharged from the user.
loanHistoryDischargingLibraryID	<b>xs:string</b>	0..1	Displays the ID of the library where the item was discharged.
loanHistoryDischargingLibraryDescription	<b>xs:string</b>	0..1	Displays the name of the library where the item was discharged.

## PatronCheckoutInfo

The PatronCheckoutInfo type displays specific checkout information for a particular patron.

This type may also be extended by the [GroupMemberCheckoutInfo](#) type.

### Ancestors

None

### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
titleKey	<b>xs:long</b>	0..1	Displays a database key to the title record.
itemID	<b>xs:string</b>	1	Displays the item ID.
callNumber	<b>xs:string</b>	1	Displays the item call number.
copyNumber	<b>xs:int</b>	1	Displays which copy of the item if multiple copies exist.
pieces	<b>xs:int</b>	1	Displays the number of pieces or parts that are included with the item (for multi-part items such as a book with a CD).
numberOfPiecesReturned	<b>xs:int</b>	0..1	Displays the number of pieces or parts that were returned at check in (for multi-part items such as a book with a CD).
title	<b>xs:string</b>	0..1	Displays the title of the item.
author	<b>xs:string</b>	0..1	Displays the item's author.
checkoutLibraryID	<b>xs:string</b>	0..1	Displays the library where the item was checked out.
checkoutLibraryDescription	<b>xs:string</b>	0..1	Displays a text description of the library where the item was checked out.
itemLibraryID	<b>xs:string</b>	0..1	Displays the item's owning library Library ID.
itemLibraryDescription	<b>xs:string</b>	0..1	Displays a text description of the item's owning library.
itemTypeID	<b>xs:string</b>	0..1	Displays the item's item type ID.
itemTypeDescription	<b>xs:string</b>	0..1	Displays a text description of the item's item type ID.
reserveCollectionID	<b>xs:string</b>	0..1	Displays the reserve collection ID. This element is only returned if the item is on academic reserve.
reserveCollectionDescription	<b>xs:string</b>	0..1	Displays a text description of the reserve collection. This element is only returned if the item is on academic reserve.
checkoutDate	<b>xs:dateTime</b>	1	Displays the date the item was checked out.
dueDate	<b>xs:dateTime</b>	1	Displays the item's due date.

Element	Type	Occ.	Description
recallDate	xs:date	0..1	Displays the date a recall notice was sent. This element is returned only if the checkout has been recalled.
recallDueDate	xs:dateTime	0..1	Displays the item due date after it is recalled.
recallNoticesSent	xs:int	0..1	Displays the number of recall notices that have been sent.
lastRenewedDate	xs:dateTime	0..1	Displays the date the item was last renewed.
renewals	xs:int	1	Displays the number of times the item has been renewed.
unseenRenewals	xs:int	1	Number of times the item was renewed without being seen by library staff (i.e. patron self-renewed).
renewalsRemaining and renewalsRemainingUnlimited are a choice. Values will be returned for one element or the other.			
renewalsRemaining	xs:int	0..1	Displays the number of times an item can be renewed. <b>Note:</b> The renewalsRemaining elements are a choice. If there is a number of renewals remaining, renewalsRemainingUnlimited will not display.
renewalsRemainingUnlimited	xs:boolean		Indicates if the number of remaining renewals is unlimited. <b>Note:</b> The renewalsRemaining elements are a choice. If renewalsRemainingUnlimited is “true”, then renewalsRemaining will not display.
unseenRenewalsRemaining and unseenRenewalsRemainingUnlimited are a choice. Values will be returned for one element or the other.			

Element	Type	Occ.	Description
unseenRenewalsRemaining	<b>xs:int</b>	0..1	Displays the number of times an item can be renewed unseen. <b>Note:</b> The unseenRenewalsRemaining elements are a choice. If there is a number of unseen renewals remaining, unseenRenewalsRemainingUnlimited will not display.
unseenRenewalsRemainingUnlimited	<b>xs:boolean</b>		Indicates if the number of remaining renewals is unlimited. <b>Note:</b> The renewalsRemaining elements are a choice. If renewalsRemainingUnlimited is “true”, then renewalsRemaining will not display.
overdue	<b>xs:boolean</b>	1	Indicates whether the item is overdue ( <b>true</b> ) or not ( <b>false</b> ).
overdueNoticesSent	<b>xs:int</b>	1	Displays the number of overdue notices the system has sent for an item.
dateOfLastOverdueNotice	<b>xs:date</b>	0..1	Displays the date that the last overdue notice was sent. This element is returned only if the setting is not “NEVER”.
fine	<b>Money</b>	0..1	Displays the monetary amount of fine associated with the item. This amount includes both any outstanding bills on the checkout and the estimated fine that would accrue if the item was returned at the time of the operation request.

## PatronCirculationInfo

The PatronCirculationInfo type displays specific circulation information for a particular patron.

### Ancestors

None

### Relevant operations

- **lookupMyAccountInfo**
- **lookupPatronInfo**

### Contents

This type has the following elements:

Element	Type	Occ.	Description
numberOfCheckouts	<b>xs:int</b>	1	Displays the number of items the patron has checked out.
numberOfClaimsReturned	<b>xs:int</b>	1	Displays the number of items the patron has claimed returned.
numberOfBookings	<b>xs:int</b>	1	Displays the number of bookings the patron has made.
numberOfRequests	<b>xs:int</b>	1	Displays the number of requests the patron has made.
numberOfUnansweredRequests	<b>xs:int</b>	1	Displays the number of unanswered requests the patron has made.
numberOfHolds	<b>xs:int</b>	1	Displays the number of holds the patron has made.
numberOfAvailableHolds	<b>xs:int</b>	1	Displays the number of holds that are available for pickup.
numberOfFees	<b>xs:int</b>	1	Displays the number of fees the patron has.
estimatedFines	<b>Money</b>	1	Displays an estimated monetary amount fined to the patron.
estimatedOverdues	<b>xs:int</b>	1	Displays an estimated number of items that are overdue.
creditBalance	<b>Money</b>	0..1	Displays the patron's credit balance. This element is only returned if there is a credit balance.
checkoutHistoryRule	<b>CheckoutHistoryRule</b>	0..1	Displays information about how charge history is kept for a specific user. checkoutHistoryRule is returned only if loan history is configured.
<b>numberOfCheckoutsAllowed</b> and <b>numberOfCheckoutsAllowedUnlimited</b> are a choice. Values will be returned for one element or the other.			



Element	Type	Occ.	Description
numberOfCheckoutsAllowed	<a href="#">xs:int</a>	1	Displays the number of checkouts allowed to the patron. <b>Note:</b> The numberOfCheckoutsAllowed elements are a choice. If there is a value for numberOfCheckoutsAllowed, then numberOfCheckoutsAllowedUnlimited will not display.
numberOfCheckoutsAllowedUnlimited	<a href="#">xs:boolean</a>		Displays “true” if the patron is allowed an unlimited amount of checkouts. <b>Note:</b> The numberOfCheckoutsAllowed elements are a choice. If there is a value for numberOfCheckoutsAllowed, then numberOfCheckoutsAllowedUnlimited will not display.

### PatronHoldInfo

The PatronHoldInfo type displays specific hold information for a particular patron.

This type may also be extended by the [GroupMemberHoldInfo](#) type.

#### Ancestors

None

#### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
holdKey	<b>xs:long</b>	1	Displays a key assigned to the particular hold.
titleKey	<b>xs:long</b>	1	Displays a database key to the title record.
itemID	<b>xs:string</b>	0..1	Displays the item ID. <b>Note:</b> itemID may not be present if a hold was placed on an order.
callNumber	<b>xs:string</b>	0..1	Displays the item's call number. This is especially important for multi-volume titles so that the Analytic Z value can display. For more information, see <b>Volume holds</b> .
displayableCallNumber	<b>xs:string</b>	0..1	Displays the item's formatted call number.
blanketHoldID	<b>xs:string</b>	0..1	Displays the blanket hold ID. For more information about blanket holds, see “Place Blanket Hold Wizard” in the Java WorkFlows help.
blanketHoldCopiesNeeded	<b>xs:int</b>	0..1	Displays the number of copies needed to fill the blanket hold.
blanketHoldCopiesReceived	<b>xs:int</b>	0..1	Displays the number of copies actually received for the blanket hold.
expiresDate	<b>xs:date</b>	0..1	Displays the date the hold expires.
holdStatus	<b>xs:int</b>	1	Displays an integer of <b>1</b> or <b>2</b> . 1 means the hold is active. 2 means the hold is inactive.
holdInactiveReasonID	<b>xs:string</b>	0..1	Displays the hold reason ID. Each holdInactiveReasonID will be associated with a <b>holdInactiveType</b> . For more information on holdInactive elements, see “Hold Reason Wizard” in the Java WorkFlows help.
holdInactiveReasonDescription	<b>xs:string</b>	0..1	Displays a text description of the hold reason.
holdInactiveType	<b>HoldInactiveType</b>	0..1	Displays the HoldInactiveType information.
itemTypeID	<b>xs:string</b>	0..1	Displays the item's item type ID. <b>Note:</b> itemTypeID may not be present if a hold was placed on an order.
itemTypeDescription	<b>xs:string</b>	0..1	Displays a text description of the item's item type.
title	<b>xs:string</b>	1	Displays the title of the item.

Element	Type	Occ.	Description
author	<a href="#">xs:string</a>	0..1	Displays the item's author.
itemLibraryID	<a href="#">xs:string</a>	0..1	Displays the item's owning library Library ID.
itemLibraryDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the item's owning library
pickupLibraryID	<a href="#">xs:string</a>	0..1	Displays the item's pickup library Library ID.
pickupLibraryDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the item's pickup library.
placedDate	<a href="#">xs:date</a>	1	Displays the date a hold was made.
mailServiceID	<a href="#">xs:string</a>	0..1	Displays the mail service ID (if the item is to be delivered by mail).
mailServiceDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the mail service (if the item is to be delivered by mail).
reserve	<a href="#">xs:boolean</a>	1	Indicates if the hold is for an academic reserve ( <b>true</b> ) or not ( <b>false</b> ).
recallStatus	<a href="#">xs:string</a>	1	Displays the recall status assigned to the hold. (For more information, see "Frequently Used Hold Fields" in the Java WorkFlows help.)
available	<a href="#">xs:boolean</a>	1	Indicates if an item is ready at the pickup library ( <b>true</b> ) or not ( <b>false</b> ).
intransit	<a href="#">xs:boolean</a>	1	Indicates if an item is in transit to the pickup library ( <b>true</b> ) or not ( <b>false</b> ).
availableDate	<a href="#">xs:date</a>	0..1	Displays the date the hold was made available for pickup. availableDate is only returned if the setting is not "NEVER".
availableExpiresDate	<a href="#">xs:date</a>	0..1	Displays the date the hold available date expires. availableExpiresDate is only returned if the setting is not "NEVER".
queuePosition	<a href="#">xs:int</a>	1	Displays the patron's position in the hold queue.
queLength	<a href="#">xs:int</a>	1	Displays the total number of patrons in the hold queue.
suspendStartDate	<a href="#">xs:date</a>	0..1	Displays the date a hold suspension starts.
suspendEndDate	<a href="#">xs:date</a>	0..1	Displays the date a hold suspension ends.

Element	Type	Occ.	Description
holdPlacedWithOverride	<b>xs:boolean</b>	0..1	Indicates if the hold was placed with an override ( <b>true</b> ) or not ( <b>false</b> ). <b>Note:</b> holdPlacedWithOverride will not be returned in a “traditional holds” configuration.
orderFiscalCycle	<b>xs:int</b>	0..1	Displays the fiscal cycle associated with the order. For more information about order elements, see “Orders Tab (Search by Order Data)” in the Java WorkFlows help. <b>Note:</b> orderFiscalCycle will only be present if the hold was placed on “on order” materials.
orderId	<b>xs:string</b>	0..1	Displays the order ID. <b>Note:</b> orderId will only be present if the hold was placed on “on order” materials.
orderLibraryID	<b>xs:string</b>	0..1	Displays the Library ID of the library where the item was ordered. <b>Note:</b> orderLibraryID will only be present if the hold was placed on “on order” materials.
orderLibraryDescription	<b>xs:string</b>	0..1	Displays a text description of the library where the item was ordered. <b>Note:</b> orderLibraryDescription will only be present if the hold was placed on “on order” materials.
orderLine	<b>xs:string</b>	0..1	Displays the item’s orderline number. <b>Note:</b> orderLine will only be present if the hold was placed on “on order” materials.

## PatronInfo

The PatronInfo type displays information about a particular patron.

### Ancestors

None

### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
userKey	<a href="#">xs:long</a>	1	Displays the database key to the patron's user record.
userID	<a href="#">xs:string</a>	1	Displays the patron's assigned user ID.
alternativeID	<a href="#">xs:string</a>	0..1	Displays the patron's alternative ID (if they have one).
webAuthID	<a href="#">xs:string</a>	0..1	Displays the Web authentication ID used for LDAP.
groupID	<a href="#">xs:string</a>	0..1	Displays the group ID if the patron is part of a user group.
groupDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the group if the patron is part of a user group.
displayName	<a href="#">xs:string</a>	0..1	Displays the patron's display name.
birthDate	<a href="#">xs:date</a>	0..1	Displays the patron's date of birth.
patronLibraryID	<a href="#">xs:string</a>	0..1	Displays the Library ID of the patron's library.
patronLibraryDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the patron's library.
department	<a href="#">xs:string</a>	0..1	Displays the department name if the patron is assigned a department (see "User Wizards Properties Fields" in the Java WorkFlows help).
preferredLanguage	<a href="#">xs:string</a>	0..1	Displays the language code of the patron's preferred language. preferredLanguage will be returned as the ISO 639-1 language code.

## PatronStatusInfo

The PatronStatusInfo type displays information specific to the patron's status. For more information, see "Privilege Tab" in the Java WorkFlows help.

### Ancestors

None

### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
datePrivilegeExpires	<a href="#">xs:date</a>	0..1	Displays the date the patron's circulation privileges expire.
statusID	<a href="#">xs:string</a>	0..1	Displays the patron's status ID.
statusDescription	<a href="#">xs:string</a>	0..1	Displays a text description of the patron's status.
statusType	<a href="#">StatusType</a>	0..1	Displays the patron's status type.
statusMessage	<a href="#">xs:string</a>	0..1	Displays the message that corresponds to the patron's status.
routingAllowed	<a href="#">xs:boolean</a>	1	Indicates if routing is allowed ( <b>true</b> ) or not ( <b>false</b> ).
userHasAlerts	<a href="#">xs:boolean</a>	1	Indicates if alerts will occur for the patron ( <b>true</b> ) or not ( <b>false</b> ).

## UserSuspensionInfo

The UserSuspensionInfo type displays information about a patron's suspension.

### Ancestors

None

### Relevant operations

- [lookupMyAccountInfo](#)
- [lookupPatronInfo](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
dateSuspended	<a href="#">xs:date</a>	1	Displays the date that the suspension started.
dateSuspensionEnds	<a href="#">xs:date</a>	1	Displays the date on which the patron can resume borrowing materials.
suspensionKey	<a href="#">xs:long</a>	1	Displays the internal key assigned to the suspension record.
linkedSuspensionKey	<a href="#">xs:string</a>	1	Displays the internal key that links the user to the suspension record.
suspensionLibraryID	<a href="#">xs:string</a>	1	Displays the ID of the library in which the suspension was generated.
suspensionLibraryDescription	<a href="#">xs:string</a>	1	Displays the descriptive name of the library in which the suspension was generated.
suspensionReasonID	<a href="#">xs:string</a>	1	Displays the reason ID for the temporary suspension of a user's borrowing privileges. These reasons may include OVERDUE, LOST, or DAMAGE.
suspensionReasonDescription	<a href="#">xs:string</a>	1	Displays the reason description for the temporary suspension of a user's borrowing privileges.
suspensionType	<a href="#">xs:string</a>	1	Displays the type of suspension. The suspension type is determined by policies. Examples of suspension types include Overdue or Damage.
numberOfUnits	<a href="#">xs:int</a>	1	Displays the number of suspension units in effect for this suspension.
title	<a href="#">xs:string</a>	0..1	Displays the title of the item, if any, for which the suspension was assessed.
itemID	<a href="#">xs:string</a>	0..1	Displays the item ID of the item, if any, for which the suspension was assessed.

## Request service

The Request service has these data types:

- [RequestBriefInfo](#)
- [RequestEntry](#) on page 224
- [RequestEntryInfo](#) on page 225
- [RequestEntryTemplateInfo](#) on page 225
- [RequestEntryType](#) on page 226
- [RequestInfo](#) on page 227

## RequestBriefInfo

The RequestBriefInfo type includes some information about a request.

### Ancestors

None

### Relevant operations

- [lookupRequests](#)

### Contents

This type has the following elements:

Parameter	Type	Occ.	Description
requestKey	<a href="#">xs:int</a>	1	Displays the request key.
requestTypeID	<a href="#">xs:string</a>	1	Displays the ID of the request type.
requestTypeDescription	<a href="#">xs:string</a>	0..1	Displays a description of the request type.
userID	<a href="#">xs:string</a>	0..1	Displays the user ID of the requesting patron.
itemID	<a href="#">xs:string</a>	0..1	Displays an item ID if one is associated with the request.
statusID	<a href="#">xs:string</a>	0..1	Displays the status ID of the request.
statusDescription	<a href="#">xs:string</a>	0..1	Displays a description of the request's status.

## RequestEntry

The RequestEntry type specifies details for the request.

### Ancestors

None

### Relevant operations

- [createRequest](#)

### Contents

This type has the following request parameters:



Parameter	Type	Descriptions	Required?
absoluteEntryNumber	<b>xs:int</b>	Specifies the absolute entry number. This is retrieved from the <b>lookupRequestTemplate</b> operation.	Y
entryID	<b>xs:string</b>	Specifies the entry ID for the entry. This is retrieved from the <b>lookupRequestTemplate</b> operation.	Y
entryData	<b>xs:string</b>	Specifies the data to include for the entry.	Y

### RequestEntryInfo

The RequestEntryInfo type displays the content of a request or reply.

#### Ancestors

- **RequestInfo**

#### Relevant operations

- **lookupRequest**

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
entryDescription	<b>xs:string</b>	1	Displays a description of the entry.
entryData	<b>xs:string</b>	1	Displays the content of the entry.

### RequestEntryTemplateInfo

The RequestEntryTemplateInfo type includes general information about each request entry in a Request template.

#### Ancestors

None

#### Relevant operations

- **lookupRequestTemplate**

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
absoluteEntryNumber	<a href="#">xs:int</a>	1	Displays the entry number.
entryFormat	<a href="#">RequestEntryType</a>	0..1	Displays the entry format.
entryID	<a href="#">xs:string</a>	1	Displays the entry ID.
entryDescription	<a href="#">xs:string</a>	1	Displays a text description of the entry.
internalEntryNumber	<a href="#">xs:int</a>	1	Displays the internal entry number used by the system. This number ranges from 1000 to 9999 and is assigned by SirsiDynix Symphony.
entryName	<a href="#">xs:string</a>	0..1	Displays a description of the entry ID.
allowedEditCreate	<a href="#">xs:boolean</a>	1	Specifies whether the entry may be edited or created ( <b>true</b> ) or not ( <b>false</b> ).
required	<a href="#">xs:boolean</a>	1	Displays whether data must be typed for the entry ( <b>true</b> ) or not ( <b>false</b> ).
dataType	<a href="#">xs:string</a>	1	Specifies whether the data in this entry must be alphabetic, alphanumeric, numeric, printable (ASCII codes 33-126), or whether the data may be any type of character.
minimumSize	<a href="#">xs:int</a>	0..1	Displays the minimum number of characters that must be typed for the entry.
maximumSize	<a href="#">xs:int</a>	0..1	Displays the maximum number of characters that must be typed for the entry.
defaultValue	<a href="#">xs:string</a>	0..1	Specifies default data that appears in the entry when the request is created.
entryClass	<a href="#">xs:string</a>	1	Displays the entry class assigned to the entry. Possible classes are LEADER, FIXED, CONTROL, CODED, or VARIABLE.
displayValue	<a href="#">xs:string</a>	0..1	Displays a text description of the entry.

## RequestEntryType

The RequestEntryType type displays the type of request entry.

### Ancestors

- [RequestEntryTemplateInfo](#)

### Relevant operations

- [lookupRequestTemplate](#)

### Contents

This type is a string (see [xs:string](#)) with the following enumeration values:

- **COMMENT**
- **EMAIL**
- **PHONE**
- **STAFFONLY**
- **USERREPLY**
- **Z3950**
- **ZIPCODE**

## RequestInfo

The RequestInfo type displays additional information about a request.

### Extends

- [RequestBriefInfo](#)

### Ancestors

None

### Relevant operations

- [lookupRequest](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
requestEntry	<a href="#">RequestEntryInfo</a>	0..*	Displays the content of the request and any subsequent requests.
replyEntry	<a href="#">RequestEntryInfo</a>	0..*	Displays the content of replies to the request.

## Security service

The Security service has this data type:

- [PrivilegeLevel](#)

## PrivilegeLevel

The PrivilegeLevel type displays the enumerated value for the user's user level on the Symphony server.

### Ancestors

None

### Relevant operations

- [lookupUserInfo](#)

### Contents

The PrivilegeLevel type is a string (see [xs:string](#)) that enumerates the expected values for user level on the Symphony server. The content for an element of this type will be one of these values:

- EXTENDED\_PUB
- PUBLIC
- STAFF
- SUPERVISOR
- SYSADMIN

## Standard service

The Standard service has these data types:

- [BestSellerDisplayInfo](#)
- [BestSellerTitleInfo](#) on page 229
- [BriefTitleInfo](#) on page 229
- [CallSummary](#) on page 230
- [HotSitesInfo](#) on page 231
- [InfoDesk](#) on page 231
- [RecommendedReadingDisplayInfo](#) on page 232
- [RecommendedReadingTitleInfo](#)
- [SearchCatalogFilters](#) on page 233
- [SearchCatalogPaging](#)
- [SearchCatalogResult](#) on page 234

### BestSellerDisplayInfo

The BestSellerDisplayInfo type displays information about a best seller list.

### Ancestors

None

### Relevant operations

- [lookupBestSellersPolicyList](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
bestSellerDisplayID	<a href="#">xs:string</a>	1	Displays the ID of the list.
description	<a href="#">xs:string</a>	1	Displays a description of the best seller list.
copyright	<a href="#">xs:string</a>	0..1	Displays copyright information about the list.

### BestSellerTitleInfo

The BestSellerTitleInfo type displays information about the titles in a best seller list.

#### Ancestors

None

#### Relevant operations

- [lookupBestSellerTitles](#)

#### Contents

This type has the following elements:

Element	Type	Occ.	Description
ISBN	<a href="#">xs:string</a>	0..1	Displays the ISBN of the title.
title	<a href="#">xs:string</a>	0..1	Displays the title information.
author	<a href="#">xs:string</a>	0..1	Displays the author information.
searchString	<a href="#">xs:string</a>	0..1	Displays a string that can be used for searching for the title in an OPAC or Web Services. This string replaces spaces with a plus sign (+) and removes unsupported characters (such as apostrophes).

### BriefTitleInfo

The BriefTitleInfo type displays basic title information for an item.

BriefTitleInfo is used as the base for other info objects, such as [BoundwithLinkInfo](#) and [HitlistTitleInfo](#).

#### Ancestors

None

#### Relevant operations

- [lookupTitleInfo](#)

#### Contents

The BriefTitleInfo type has the following elements:

Element	Type	Occ.	Description
titleID	<a href="#">xs:long</a>	0..1	Displays the item's title ID.
materialType	<a href="#">xs:string</a>	0..1	Displays what material type the item is.
title	<a href="#">xs:string</a>	1	Displays the title of the item.
author	<a href="#">xs:string</a>	0..1	Displays the item's author.
line3	<a href="#">xs:string</a>	0..1	Displays whatever generic information was assigned to field 3 on the Symphony server.
line4	<a href="#">xs:string</a>	0..1	Displays whatever generic information was assigned to field 4 on the Symphony server.
yearOfPublication	<a href="#">xs:int</a>	0..1	Displays the year the item was published.
edition	<a href="#">xs:string</a>	0..1	Displays the item's edition (if applicable).
copiesOnOrder	<a href="#">xs:int</a>	0..1	Displays how many copies of the item are on order.
ISBN	<a href="#">xs:string</a>	0..*	Displays the item's ISBN.
SICI	<a href="#">xs:string</a>	0..*	Displays the item's SICI (serial issue and contribution identifier).
UPC	<a href="#">xs:string</a>	0..*	Displays the item's UPC (if present).
OCLCControlNumber	<a href="#">xs:string</a>	0..1	Displays the item's OCLC control number.
url	<a href="#">xs:string</a>	0..1	Displays any URL link that is associated with the item.
message	<a href="#">xs:string</a>	0..1	Displays any catalog message associated with the item.

This type also references the following elements:

Element	Occ.	Description
<a href="#">TitleAvailabilityInfo</a>	0..1	Displays information about the item's availability.



The [TitleAvailability](#) element is returned only if you set [includeAvailabilityInfo](#) to "true" in the request.

### CallSummary

The [CallSummary](#) type displays call number information for a title.

#### Ancestors

- [TitleInfo](#)

**Relevant operations**

- [lookupTitleInfo](#)

**Contents**

The CallSummary type has the following elements:

Element	Type	Occ.	Description
itemID	<a href="#">xs:string</a>	1	Displays the item's ID.
callNumber	<a href="#">xs:string</a>	0..1	Displays the item's call number.
analyticZ	<a href="#">xs:string</a>	0..1	Displays the Analytic Z portion of the item's call number. This is applicable to <a href="#">Volume holds</a> .
heldAtLibrariesDisplay	<a href="#">xs:string</a>	0..1	Displays the descriptive name of the library or libraries where the item is held.

**HotSitesInfo**

The HotSitesInfo type displays information about a hot site.

**Ancestors**

None

**Relevant operations**

- [lookupHotSites](#)

**Contents**

This type has the following elements:

Element	Type	Occ.	Description
title	<a href="#">xs:string</a>	1	Displays a the title of the hot site.
url	<a href="#">xs:string</a>	1	Displays the URL of the hot site.

**InfoDesk**

The InfoDesk type contains enumerated values for a predefined searches in Symphony.

**Ancestors**

None

**Relevant operations**

- [searchInfoDesk](#)

**Contents**

This type is a string (see [xs:string](#)) with the following enumeration values:

- **ANNOUNCE**
- **HOURS**
- **NEW**

**RecommendedReadingDisplayInfo**

The RecommendedReadingDisplayInfo type displays information about a recommended reading list.

**Ancestors**

None

**Relevant operations**

- [LookupRecommendedReadingDisplayList](#)

**Contents**

This type has the following elements:

Element	Type	Occ.	Description
recommendedReadingDisplayID	<a href="#">xs:string</a>	1	Displays the ID of the list.
description	<a href="#">xs:string</a>	1	Displays a description of the recommended reading list.
note	<a href="#">xs:string</a>	0..1	Displays note data about the list.

**RecommendedReadingTitleInfo**

The RecommendedReadingInfo type displays information about the titles in a Recommended Reading list.

**Ancestors**

None

**Relevant operations**

- [lookupRecommendedReadingTitles](#)

**Contents**

This type has the following elements:



Element	Type	Occ.	Description
title	<a href="#">xs:string</a>	0..1	Displays the title information.
author	<a href="#">xs:string</a>	0..1	Displays the author information.
searchString	<a href="#">xs:string</a>	0..1	Displays a string that can be used for searching for the title in an OPAC or Web Services. This string replaces spaces with a plus sign (+) and removes unsupported characters (such as apostrophes).

## SearchCatalogFilters

The SearchCatalogFilters type contains filter elements that you can apply to catalog searches to limit the data you receive in the search results.

### Ancestors

None

### Relevant operations

- [searchCatalog](#)
- [searchInfoDesk](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
libraryFilter	<a href="#">xs:string</a>	0..1	Specifies you only want to return results from a particular library.
languageFilter	<a href="#">xs:string</a>	0..1	Specifies you only want to return results in a particular language.
formatFilter	<a href="#">xs:string</a>	0..1	Specifies you only want to return results of a particular format.
pubyearFilter	<a href="#">xs:string</a>	0..1	Specifies you only want to return results within a particular publication year.
homelocationFilter	<a href="#">xs:string</a>	0..1	Specifies you only want to return results from a particular home location.
itemtypeFilter	<a href="#">xs:string</a>	0..1	Specifies you only want to return results of a particular item type.
itemcat1Filter	<a href="#">xs:string</a>	0..1	Specifies you only want to return results of an Item Cat1 policy type.
itemcat2Filter	<a href="#">xs:string</a>	0..1	Specifies you only want to return results of an Item Cat2 policy type.

## SearchCatalogPaging

- [searchInfoDeskPaging](#)

Delete this text and replace it with your own content.

## SearchCatalogResult

The SearchCatalogResult type displays the results of searches made with the [searchCatalog](#), [searchCatalogPaging](#), [searchInfoDesk](#), and [searchInfoDeskPaging](#) operations.

### Ancestors

None

### Relevant operations

- [searchCatalog](#)
- [searchCatalogPaging](#)
- [searchInfoDesk](#)
- [searchInfoDeskPaging](#)

### Contents

This type has the following elements:

Element	Type	Occ.	Description
queryID	<a href="#">xs:string</a>	1	Displays the ID of your search query.
totalHits	<a href="#">xs:long</a>	1	Displays the total number of search results available.
firstHitDisplayed	<a href="#">xs:long</a>	1	Displays the first search result record number returned in the response.
lastHitDisplayed	<a href="#">xs:long</a>	1	Displays the last search result record number returned in the response.

This type also references the following elements:

Element	Occ.	Description
<a href="#">HitlistTitleInfo</a>	0..*	Displays title information for items on the hitlist.

# Chapter 6: Faults

This section describes some of the common fault messages you may encounter while using Web Services.

- **HatIOException: Cannot connect to host** on page 235
- **HatIOException: Connection to server has been closed** on page 236
- **HatIOException: Read of response timed out** on page 236
- **HatSessionException: Unable to acquire session with server** on page 237
- **HeaderException: Symphony Soap Header is required** on page 237
- **HeaderException: Bad Client ID** on page 237
- **HTTP 404 File not found** on page 238
- **HTTP 500 Internal server error** on page 238
- **HTTP 501 Not Implemented** on page 238
- **InvalidRequestException: Entity Operation is required** on page 239
- **InvalidRequestException: Invalid Operation for this request** on page 239
- **LicenseException.notLicensed** on page 240
- **RestInvocationException.missingMethod** on page 240
- **SecurityServiceException: Access Denied** on page 240
- **SecurityServiceException: Session has timed out** on page 241
- **SecurityServiceException: Unable to log in** on page 241

## HatIOException: Cannot connect to host

This fault indicates that there is a problem with the network connection to the ILS server.

### Cause

Web Services cannot connect to the netserver or the ILS server is down. For example, the server may have been halted and restarted for routine maintenance or configuration updates.

### Action

Retry the request later.

If you continue to receive the fault, contact the library system administrator to verify that the ILS server is running and that Web Services is properly configured.

### See Also

- [Faults on page 235](#)

## HatIOException: Connection to server has been closed

This fault indicates a problem communicating with the ILS server.

### Cause

Web Services is unable to communicate with the ILS.

The ILS server was temporarily unavailable (because of a restart to initialize configuration changes, for example).

### Action

Retry the request later.

If you continue to receive the fault, contact the library system administrator to verify that the ILS server is running and that Web Services is properly configured.

### See Also

- [Faults on page 235](#)

## HatIOException: Read of response timed out

This error indicates that a request to the ILS did not get a response before the socket read timed out.

### Cause

The request took longer to complete on the ILS than the socket read timeout allows, possibly because of the amount of data you are requesting (for example, if you use `lookupTitleInfo` with multiple title or item IDs and many `Include` options).

### Action

Retry the request.

Retry the request with different options. For example, if you are requesting hold info for a patron, request only active holds, not all holds. If you are searching the catalog with a large number of hits to display, reduce the number of hits to display.

If requests continue to result in a read timeout, talk to the library system administrator about increasing the socket read timeout value for the Web Services instance.

### See Also

- [Faults on page 235](#)

## HatSessionException: Unable to acquire session with server

This fault indicates a problem establishing a communication session with the ILS server.

### Cause

Maximum number of ILS sessions has been reached and all sessions are busy.

A socket read is timing out before a response is received.

Web Services is unable to establish a connection with the ILS, possibly because of a dropped connection.

### Action

Retry the request later.

If you continue to receive the fault, contact the library system administrator to verify that the ILS server is running and that Web Services is properly configured.

### See Also

- [Faults on page 235](#)

## HeaderException: Symphony Soap Header is required

This fault indicates that the required SirsiDynix header is missing from a request (SOAP only).

### Cause

The SdHeader element is missing or invalid.

### Action

Make sure you include an SdHeader with a valid client ID. Verify that element names are spelled correctly.

### See Also

- [SdHeader on page 25](#)
- [Faults on page 235](#)

## HeaderException: Bad Client ID

This fault indicates that the client ID in the request header is invalid or missing.

### Cause

The client ID specified in the request header is invalid.

**Action**

Use a valid client ID. Verify that the client ID is spelled correctly. Client ID is case-sensitive.

**See Also**

- [Faults on page 235](#)

## HTTP 404 File not found

This HTTP status code indicates a problem with the URL you used to access a service.

**Cause**

The path you specified is not a valid endpoint.

**Action**

Use a valid endpoint. Verify that all parts of the path are spelled correctly. Paths are case-sensitive.

**See Also**

- [Service endpoints on page 4](#)
- [Faults on page 235](#)

## HTTP 500 Internal server error

This HTTP status code indicates that the server was unable to complete the request.

**Cause**

This is the expected HTTP status code for any fault.

**Action**

Inspect the fault message returned and take appropriate action. (See [Faults on page 235](#).)

**See Also**

- [Faults on page 235](#)

## HTTP 501 Not Implemented

This HTTP status code indicates that the server does not recognize the requested method in a URL.

**Cause**

This is most likely caused by an invalid REST request.

**Action**

Make sure you are using supported requests. Check the spelling of the method in the URL including case (method names are case-sensitive).

**See also**

- [Service endpoints](#) on page 4
- [RestInvocationException.missingMethod](#) on page 240
- [Faults](#) on page 235

## InvalidRequestException: Entity Operation is required

This fault indicates that the required Operation attribute is missing for some Entity type element in the request.

**Cause**

You did not specify an Operation for some Entity type element in the request.

**Action**

Specify a valid value for the operation attribute.

**See Also**

- [Entity](#) on page 194
- [Faults](#) on page 235

## InvalidRequestException: Invalid Operation for this request

This fault indicates that the value of the Operation attribute of an Entity type element is invalid for the request.

**Cause**

The value specified for the operation attribute of an Entity type element is invalid.

The value specified, even though one of the allowed options, is not valid in this context.

**Action**

Use a valid attribute. Valid values include NOOP, CREATE, UPDATE, DELETE.

Verify that the attribute value is spelled correctly. Values for the operation attribute are case-sensitive.

#### See Also

- [Entity](#) on page 194
- [Faults](#) on page 235

## LicenseException.notLicensed

This fault indicates that the Web Services instance does not have a license that allows the operation you've requested.

#### Cause

Some operations require a license (for example, operations in the Patron service). If the library has not installed a license for these, you will get a Not Licensed exception when you try to access licensed operations.

#### Action

Consult with the library system administrator to verify that these services are licensed and the license is properly installed.

#### See Also

- [Faults](#) on page 235
- [Licensed operations](#) on page 2

## RestInvocationException.missingMethod

This fault indicates that the specified REST operation is unrecognized. The error message includes the operation, "Method <operation-name> not implemented".

#### Cause

The operation you specified in the REST request is misspelled or otherwise invalid.

#### Action

Use a valid operation name in the URL. Verify that the operation name is spelled correctly. Operation names are case-sensitive.

#### See Also

- [HTTP 501 Not Implemented](#) on page 238
- [Service endpoints](#) on page 4
- [Faults](#) on page 235

## SecurityServiceException: Access Denied

This fault indicates a problem with user privileges.



**Cause**

The current session user (implied in the request `sessionToken`) does not have privileges to complete the requested action. For example, the user's privilege level is insufficient or the user does not have a required command in the user's command list.

The operation requires a session token and no `sessionToken` was specified.

**Action**

Create a new session for a user with the required permissions. Make sure the user has the necessary privilege level and commands (see [User access on page 11](#)).

Make sure you pass in a `sessionToken` for privileged operations (see [Privileged operations on page 19](#)).

**See Also**

- [Privileged operations on page 19](#)
- [loginUser on page 71](#)
- [Faults on page 235](#)

## SecurityServiceException: Session has timed out

This fault indicates that the session token specified in the request is invalid.

**Cause**

The session token specified in the request is not valid. For example, the session was already terminated with a `logoutUser` request, the session expired, or the value of `sessionToken` is invalid.

**Action**

Create a new session and repeat the request.

**See Also**

- [loginUser on page 71](#)
- [logoutUser on page 72](#)
- [Faults on page 235](#)

## SecurityServiceException: Unable to log in

This fault indicates a problem with the credentials used in a login request.

**Cause**

This fault may be returned by a login request for any of the following reasons.

- User is invalid or misspelled. User IDs and alternate IDs may be case-sensitive (for example, if the Web Services instances uses LDAP authentication).
- Password is invalid or misspelled. Passwords are case-sensitive.
- Required password missing.

**Action**

Make sure the user and password are valid and spelled correctly.

**See Also**

- [loginUser](#) on page 71
- [Faults](#) on page 235

# Appendix 7: Service Schemas

For convenient reference, this section lists the schemas for each service in Web Services.

- [Standard service](#) on page 243
- [Security service](#) on page 255
- [Patron service](#) on page 257
- [Admin service](#) on page 274
- [Request service](#) on page 287
- [Circulation service](#) on page 290
- [Catalog service](#) on page 295
- [Common service](#) on page 296
- [Selection service](#) on page 298

## Standard service

```
<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.sirsiidynix.com/symws/standard"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://schemas.sirsiidynix.com/symws/standard">

  <!-- messages -->
  <xs:element name="LookupTitleInfoRequest">
    <xs:complexType>
      <xs:sequence>
        <!-- Library ID or Search Library Group ID -->
        <xs:element name="libraryFilter" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:choice>
          <xs:element name="titleID" type="xs:long" minOccurs="1" maxOccurs="unbounded"/>
          <xs:element name="itemID" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
        </xs:choice>
        <!-- The next seven parameters are treated as false if not present. -->
        <xs:element name="includeAvailabilityInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        <xs:element name="includeItemInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        <xs:element name="includeCatalogingInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        <xs:element name="includeOrderInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        <xs:element name="includeOPACInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        <xs:element name="includeBoundTogether" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        <xs:element name="includeMarcHoldings" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        <!-- If no choice is made here, marcEntryFilter=NONE is assumed. -->
        <xs:choice minOccurs="0" maxOccurs="1">
```

```

<xs:element name="marcEntryFilter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ALL"/>
      <xs:enumeration value="BRIEF"/>
      <xs:enumeration value="FULL"/>
      <xs:enumeration value="NONE"/>
      <xs:enumeration value="TEMPLATE"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="marcEntryID" type="xs:string" maxOccurs="unbounded"/>
<!-- includeRelatedSearchInfo is treated as false if not present. If true,
      related search information is included in the display of bibliographic
      entries. Ignored if no marc entries are displayed. -->
</xs:choice>
<xs:element name="includeRelatedSearchInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
<xs:element name="includeCallNumberSummary" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="LookupTitleInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="TitleInfo" minOccurs="1" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="SearchCatalogRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="query" minOccurs="1" maxOccurs="8">
        <xs:complexType>
          <xs:sequence>
            <!-- The server supports a maximum term length of 1,024 characters. -->
            <xs:element name="term" minOccurs="1" maxOccurs="1">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="1024"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <!-- The search index will default to GENERAL if not specified. -->
            <xs:element name="searchType" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <!-- The search operator will default to AND if not specified. -->
            <xs:element name="operator" minOccurs="0" maxOccurs="1">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="AND"/>
                  <xs:enumeration value="OR"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        <xs:enumeration value="NOT"/>
        <xs:enumeration value="XOR"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- Will default to false if not specified. -->
<xs:element name="exactMatch" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
<!-- Will default to 20 if not specified. -->
<xs:element name="hitsToDisplay" type="xs:nonNegativeInteger" minOccurs="0"
    maxOccurs="1"/>
<xs:element name="filters" type="SearchCatalogFilters" minOccurs="0" maxOccurs="1"/>
<!-- includeAvailabilityInfo will be treated as false if not present. If true, causes
    TitleAvailabilityInfo to be included for each hitlist title. -->
<xs:element name="includeAvailabilityInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="SearchCatalogResponse" type="SearchCatalogResult"/>

<xs:element name="SearchCatalogPagingRequest" type="SearchCatalogPaging"/>

<xs:element name="SearchCatalogPagingResponse" type="SearchCatalogResult"/>

<xs:element name="LookupMostPopularRequest">
    <xs:complexType>
        <xs:sequence>
            <!-- Required type of Most Popular list -->
            <xs:element name="listType" minOccurs="1" maxOccurs="1">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="AUTHOR"/>
                        <xs:enumeration value="SUBJECT"/>
                        <xs:enumeration value="TITLE"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <!-- Optional Library ID or Library Group ID -->
            <xs:element name="libraryFilter" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <!-- Optional alternate file name for source file -->
            <xs:element name="sourceFile" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <!-- Optional maximum authors, subjects, or titles to display -->
            <xs:element name="linesToDisplay" type="xs:nonNegativeInteger" minOccurs="0"
                maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="LookupMostPopularResponse">
    <xs:complexType>

```

```

    <xs:sequence>
      <xs:element name="linesDisplayed" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="MostPopularInfo" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="SearchInfoDeskRequest">
  <xs:complexType>
    <xs:sequence>
      <!-- BLST policy name that the library uses to tag predefined searches -->
      <xs:choice minOccurs="1" maxOccurs="1">
        <xs:element name="infoDesk" type="InfoDesk"/>
        <xs:element name="customInfoDesk" type="xs:string"/>
      </xs:choice>
      <!-- Will default to 20 if not specified. -->
      <xs:element name="hitsToDisplay" type="xs:nonNegativeInteger" minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="filters" type="SearchCatalogFilters" minOccurs="0" maxOccurs="1"/>
      <!-- includeAvailabilityInfo will be treated as false if not present. If true, causes
        TitleAvailabilityInfo to be included for each hitlist title. -->
      <xs:element name="includeAvailabilityInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="SearchInfoDeskResponse" type="SearchCatalogResult"/>

<xs:element name="SearchInfoDeskPagingRequest" type="SearchCatalogPaging"/>

<xs:element name="SearchInfoDeskPagingResponse" type="SearchCatalogResult"/>

<xs:element name="LookupBestSellersDisplayListRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="bestSellerID" type="xs:string" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="LookupBestSellersDisplayListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="bestSellerDisplayInfo" type="BestSellerDisplayInfo" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="LookupRecommendedReadingDisplayListRequest">
  <xs:complexType>
    <xs:sequence>

```

```

        <xs:element name="recommendedReadingID" type="xs:string" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="LookupRecommendedReadingDisplayListResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="recommendedReadingDisplayInfo" type="RecommendedReadingDisplayInfo"
                minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:complexType name="BestSellerDisplayInfo">
    <xs:sequence>
        <xs:element name="bestSellerDisplayID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="description" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="copyright" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="RecommendedReadingDisplayInfo">
    <xs:sequence>
        <xs:element name="recommendedReadingDisplayID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="description" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="note" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="LookupBestSellerTitlesRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="bestSellerID" type="xs:string" minOccurs="1" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="LookupBestSellerTitlesResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="bestSellerTitleInfo" type="BestSellerTitleInfo" minOccurs="0"
                maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="LookupRecommendedReadingTitlesRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="recommendedReadingID" type="xs:string" minOccurs="1" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

</xs:element>

<xs:element name="LookupRecommendedReadingTitlesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="recommendedReadingTitleInfo" type="RecommendedReadingTitleInfo"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:complexType name="BestSellerTitleInfo">
  <xs:sequence>
    <xs:element name="ISBN" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="searchString" type="xs:string" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RecommendedReadingTitleInfo">
  <xs:sequence>
    <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="searchString" type="xs:string" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:element name="LookupHotSitesRequest" />

<xs:element name="LookupHotSitesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="hotSitesInfo" type="HotSitesInfo" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="VersionRequest" />
<xs:element name="VersionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="version" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="product" type="xs:string"/>
            <xs:element name="version" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```



```

<xs:element name="LicenseRequest"/>
<xs:element name="LicenseResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="attribute" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="key" type="xs:string"/>
            <xs:element name="value" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- objects -->
<xs:element name="TitleInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="titleID" type="xs:long" minOccurs="1" maxOccurs="1"/>
      <!-- TitleAvailabilityInfo is included only if includeAvailabilityInfo is true -->
      <xs:element ref="TitleInfo" minOccurs="0" maxOccurs="1"/>
      <!-- The next four elements will be present if includeCatalogingInfo is true -->
      <xs:element name="titleControlNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="catalogFormatID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="catalogFormatType" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="materialType" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <!-- The next fifteen elements will be present if includeOPACInfo is true -->
      <xs:element name="baseCallNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="ISBN" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="SICI" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="UPC" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="OCLCControlNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="sisacID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="publisherName" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="datePublished" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="yearOfPublication" type="xs:int" minOccurs="0" maxOccurs="1"/>
      <xs:element name="extent" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="netLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="numberOfCallNumbers" type="xs:int" minOccurs="0" maxOccurs="1"/>
      <xs:element name="numberOfTitleHolds" type="xs:int" minOccurs="0" maxOccurs="1"/>
      <!-- The next three elements are present if includeOrderInfo is true -->
      <xs:element name="copiesOnOrder" type="xs:int" minOccurs="0" maxOccurs="1"/>
      <xs:element name="outstandingCopiesOnOrder" type="xs:int" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="TitleOrderInfo" minOccurs="0" maxOccurs="unbounded"/>
      <!-- present if includeItemInfo is true -->
      <xs:element ref="CallInfo" minOccurs="0" maxOccurs="unbounded"/>
      <!-- present if marcEntryFilter parameter is not NONE or marcEntryID parameter
           is present -->

```

```

    <xs:element ref="BibliographicInfo" minOccurs="0" maxOccurs="1"/>
    <!-- present if includeMarcHoldings is true. -->
    <xs:element ref="MarcHoldingsInfo" minOccurs="0" maxOccurs="unbounded"/>
    <!-- Boundwith links count and list are included only if includeBoundTogether is true.
    -->
    <xs:element name="numberOfBoundwithLinks" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="BoundwithLinkInfo" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="callSummary" type="CallSummary" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="CallInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="libraryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="classificationID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="callNumber" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="numberOfCopies" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <!-- Bound parent information is returned only if includeBoundTogether is true. -->
      <xs:element name="boundParentAuthor" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="boundParentTitle" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="ItemInfo" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="ItemInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="itemID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="itemTypeID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="currentLocationID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="homeLocationID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <!-- Present only if the item is currently checked out. -->
      <xs:element name="dueDate" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
      <!-- Present only if the item is currently checked out and has been recalled. -->
      <xs:element name="recallDueDate" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
      <!-- reshelving location information is returned only for copies which are
      considered to be in the process of being reshelved. -->
      <xs:element name="reshelvingLocationID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <!-- transit information is returned only for copies which are in transit from one
      library to another. -->
      <xs:element name="transitSourceLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="transitDestinationLibraryID" type="xs:string" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="transitReason" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="transitDate" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
      <xs:element name="chargeable" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
      <xs:element name="numberOfHolds" type="xs:int" minOccurs="0" maxOccurs="1"/>
      <!-- Reserve information is sent only for items on academic reserve -->
      <xs:element name="reserveCollectionID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="reserveCirculationRule" type="xs:string" minOccurs="0" maxOccurs="1"/>

```

```

    <!-- Booking information is sent only for items available for booking -->
    <xs:element name="mediaDeskID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="fixedTimeBooking" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    <xs:element name="publicNote" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="staffNote" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="BoundwithLinkInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="linkedAsParent" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
      <xs:element name="callNumber" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="libraryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <!-- If the linked title has more than one copy, this is the itemID of the
           first copy of the linked callnum. -->
      <xs:element name="itemID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="linkedTitle" type="BriefTitleInfo" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:complexType name="CallSummary">
  <xs:sequence>
    <xs:element name="itemID" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="callNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="analyticZ" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="heldAtLibrariesDisplay" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<!-- This element contains the entries for the bibliographic record. -->
<xs:element name="BibliographicInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="MarcEntryInfo" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- This element contains the entries for the MARC holdings record. -->
<xs:element name="MarcHoldingsInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="holdingLibraryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="MarcEntryInfo" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="MarcEntryInfo">
  <xs:complexType>

```

```

<xs:sequence>
  <xs:element name="label" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <!-- alternateLabel is for entries linked to an 'alternate graphic representation' -->
  <xs:element name="alternateLabel" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="entryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
  <xs:element name="indicators" minOccurs="1" maxOccurs="1">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="text" type="xs:string" minOccurs="1" maxOccurs="1"/>
  <xs:element name="unformattedText" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="entryTypeCodes" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="url" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <!-- The elements below are for hypertext lookup. They are returned only if
       includeRelatedSearchInfo is true in the request. -->
  <xs:element name="relatedLookupText" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="relatedLookupIndexID" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="relatedLookupHeadingTypeID" type="xs:string" minOccurs="0"
    maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="TitleOrderInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="orderLibraryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="copiesOrdered" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="volumesOrdered" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="orderNote" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="orderDateReceived" type="xs:date" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:complexType name="SearchCatalogFilters">
  <xs:sequence>
    <xs:element name="libraryFilter" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="languageFilter" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="formatFilter" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <!-- server supports comparisons and ranges, so string is used instead of gYear -->
    <xs:element name="pubyearFilter" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="homelocationFilter" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemtypeFilter" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemcat1Filter" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemcat2Filter" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="SearchCatalogResult">

```

```

<xs:sequence>
  <xs:element name="queryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
  <xs:element name="totalHits" type="xs:long" minOccurs="1" maxOccurs="1"/>
  <xs:element name="firstHitDisplayed" type="xs:long" minOccurs="1" maxOccurs="1"/>
  <xs:element name="lastHitDisplayed" type="xs:long" minOccurs="1" maxOccurs="1"/>
  <xs:element ref="HitlistTitleInfo" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="SearchCatalogPaging">
  <xs:sequence>
    <xs:element name="SearchCatalogPaging" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="SearchCatalogPaging" type="xs:long" minOccurs="1" maxOccurs="1"/>
    <xs:element name="SearchCatalogPaging" type="xs:long" minOccurs="1" maxOccurs="1"/>
    <!-- includeAvailabilityInfo will be treated as false if not present. If true, causes
         TitleAvailabilityInfo to be included for each hitlist title. -->
    <xs:element name="SearchCatalogPagingHitlistTitleInfo" type="xs:boolean" minOccurs="0"
maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="HitlistTitleInfo">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="BriefTitleInfo">
        <xs:sequence>
          <xs:element name="documentNumber" type="xs:long" minOccurs="1" maxOccurs="1"/>
          <xs:element name="callNumber" type="xs:string" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

<xs:element name="MostPopularInfo">
  <xs:complexType>
    <xs:sequence>
      <!-- The author, subject, or title in a normalized form suitable for use as a
           search term -->
      <xs:element name="searchableHeading" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <!-- The author, subject, or title formatted for public display -->
      <xs:element name="displayableHeading" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="TitleAvailabilityInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="totalCopiesAvailable" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="libraryWithAvailableCopies" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="totalResvCopiesAvailable" type="xs:int" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

    <xs:element name="libraryWithAvailableResvCopies" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="locationOfFirstAvailableItem" type="xs:string" minOccurs="0"
      maxOccurs="1"/>
    <!-- True if there exists at least one copy on which patrons may place holds -->
    <xs:element name="holdable" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
    <!-- True if there exists at least one copy which is available for booking -->
    <xs:element name="bookable" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<!-- An enumeration of the set of predefined search tags delivered with the Symphony
      installation. -->
<xs:simpleType name="InfoDesk">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ANNOUNCE"/>
    <xs:enumeration value="HOURS"/>
    <xs:enumeration value="NEW"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="BriefTitleInfo">
  <xs:sequence>
    <!-- Should only be null if the information from the ILS is incomplete -->
    <xs:element name="titleID" type="xs:long" minOccurs="0" maxOccurs="1"/>
    <!-- TitleAvailabilityInfo is included only if includeAvailabilityInfo is true -->
    <xs:element ref="TitleAvailabilityInfo" minOccurs="0" maxOccurs="1"/>
    <xs:element name="materialType" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="title" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="line3" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="line4" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="yearOfPublication" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="edition" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="copiesOnOrder" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ISBN" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="SICI" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="UPC" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="OCLCControlNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <!-- Link to electronic content -->
    <xs:element name="url" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="message" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="BestSellerInfo">
  <xs:sequence>
    <xs:element name="ISBN" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="searchTerm" type="xs:string" minOccurs="0" maxOccurs="1" />
  </xs:sequence>

```

```

</xs:complexType>

<xs:complexType name="RecommendedReadInfo">
  <xs:sequence>
    <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="searchTerm" type="xs:string" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="HotSitesInfo">
  <xs:sequence>
    <xs:element name="title" type="xs:string" />
    <xs:element name="url" type="xs:string" />
  </xs:sequence>
</xs:complexType>

</xs:schema>

```

## Security service

```

<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.sirsidynix.com/symws/security"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://schemas.sirsidynix.com/symws/security">

  <!-- messages -->
  <xs:element name="AuthenticateUserRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="login" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="password" type="xs:string" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="AuthenticateUserResponse" type="xs:boolean"/>

  <xs:element name="LoginUserRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="login" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="password" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="workstationID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="LoginUserResponse">
    <xs:complexType>
      <xs:sequence>

```

```

        <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="sessionToken" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="LogoutUserRequest"/>

<xs:element name="LookupUserInfoRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <!-- only accountability operators will be able to request accountable information -->
            <!-- assumed to be false if not specified -->
            <xs:element name="includeAccountability" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
            <!-- assumed to be false if not specified -->
            <xs:element name="includeAllowedCommands" type="xs:boolean" minOccurs="0"
                maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="LookupUserInfoResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="displayName" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="privilegeLevel" type="PrivilegeLevel" minOccurs="1" maxOccurs="1"/>
            <xs:element name="sharedLogin" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
            <xs:element name="validUser" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
            <xs:element name="libraryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="accessID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="environmentID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="AccountabilityInfo" minOccurs="0" maxOccurs="1"/>
            <xs:element name="allowedCommand" type="xs:string" minOccurs="0"
                maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="LookupSessionInfoRequest"/>

<xs:element name="LookupSessionInfoResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="userAccessID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="userAccessDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <xs:element name="userEnvironmentID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="userEnvironmentDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <xs:element name="userProfileID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="userProfileDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```



```

    <xs:element name="userLibraryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="userLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="serverIPAddress" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="serverPlatform" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="serverVersion" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<!-- objects -->
<!-- values correspond to the Symphony server "user hierarchy" enumeration -->
<xs:element name="AccountabilityInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="clearanceID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="patronType" minOccurs="0" maxOccurs="1">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="EXTERNAL"/>
            <xs:enumeration value="INTERNAL"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="clearanceGrantingFacility" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="citizenshipID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="ssn" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="birthDate" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="birthPlace" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="accessRestrictionID" type="xs:string" minOccurs="0"
        maxOccurs="unbounded"/>
      <xs:element name="needToKnowID" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:simpleType name="PrivilegeLevel">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EXTEND_PUB"/>
    <xs:enumeration value="PUBLIC"/>
    <xs:enumeration value="STAFF"/>
    <xs:enumeration value="SUPERVISOR"/>
    <xs:enumeration value="SYSADMIN"/>
  </xs:restriction>
</xs:simpleType>

</xs:schema>

```

## Patron service

```

<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.sirsiidynix.com/symws/patron"

```

```

xmlns:common="http://schemas.sirsiidynix.com/symws/common"
xmlns:circulation="http://schemas.sirsiidynix.com/symws/circulation"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.sirsiidynix.com/symws/patron">
<xs:import namespace="http://schemas.sirsiidynix.com/symws/common"
  schemaLocation="common.xsd"/>
<xs:import namespace="http://schemas.sirsiidynix.com/symws/circulation"
  schemaLocation="circulation.xsd"/>

<!-- messages -->
<xs:element name="RenewMyCheckoutRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="itemID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="RenewMyCheckoutResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="message" type="xs:string" minOccurs="1" maxOccurs="1" />
      <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="userName" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="itemID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="callNumber" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="title" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="dueDate" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="CreateMyHoldRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:choice minOccurs="1">
        <xs:element name="itemID" type="xs:string" minOccurs="0" maxOccurs="1" />
        <xs:element name="titleKey" type="xs:long" minOccurs="0" maxOccurs="1" />
      </xs:choice>
      <xs:element name="callNumber" type="xs:string" minOccurs="0" maxOccurs="1" />
      <!-- if not present, holdType defaults to TITLE on the server -->
      <xs:element name="holdType" type="circulation:HoldType" minOccurs="0" maxOccurs="1" />
      <!-- if not present, holdRange defaults to GROUP -->
      <xs:element name="holdRange" type="circulation:HoldRange" minOccurs="0" maxOccurs="1" />
      <xs:choice minOccurs="0" maxOccurs="1">
        <xs:element name="pickupLibraryID" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="mailServiceID" type="xs:string" minOccurs="1" maxOccurs="1" />
      </xs:choice>
      <!-- if not present, expiresDate will be calculated by the ILS -->
      <xs:element name="expiresDate" type="common:NillableDate" minOccurs="0" maxOccurs="1"/>
      <xs:element name="suspendStartDate" type="xs:date" minOccurs="0" maxOccurs="1" />
      <xs:element name="suspendEndDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

    <xs:element name="comment" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<!-- CreateMyHoldResponse will be the holdKey -->
<xs:element name="CreateMyHoldResponse" type="xs:long"/>

<xs:element name="CancelMyHoldRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="holdKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="CancelMyHoldResponse" type="xs:boolean"/>

<xs:element name="LookupMyAccountInfoRequest">
  <xs:complexType>
    <xs:sequence>
      <!-- All boolean "include" parameters are assumed false if not present. -->
      <xs:element name="includePatronInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      <xs:element name="includePatronCirculationInfo" type="xs:boolean" minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="includePatronAddressInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      <xs:element name="includePatronCheckoutInfo" type="circulation:CheckoutInfoFilter"
        minOccurs="0" maxOccurs="1"/>
      <xs:element name="includePatronCheckoutHistoryInfo" type="xs:boolean" minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="includePatronHoldInfo" type="circulation:HoldInfoFilter" minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="includeFeeInfo" type="circulation:FeeInfoFilter" minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="includePatronStatusInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      <xs:element name="includeUserGroupInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      <xs:element name="includeUserSuspensionInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      <!-- if present, stationLibraryID will override session station Library ID -->
      <xs:element name="stationLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="LookupMyAccountInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <!-- patronInfo is returned only if includePatronInfo is true -->
      <xs:element name="patronInfo" type="PatronInfo" minOccurs="0" maxOccurs="1"/>
      <!-- PatronCirculationInfo is returned only if includePatronCirculationInfo
        is true -->
      <xs:element name="patronCirculationInfo" type="PatronCirculationInfo" minOccurs="0"
        maxOccurs="1"/>
      <!-- Patron addresses are returned only if includePatronAddressInfo is true -->

```

```

<xs:element name="patronAddressInfo" type="PatronAddressInfo" minOccurs="0"
  maxOccurs="1"/>
<!-- patronCheckoutInfo is returned only if includePatronCheckoutInfo is set -->
<xs:element name="patronCheckoutInfo" type="PatronCheckoutInfo" minOccurs="0"
  maxOccurs="unbounded"/>
<!-- patronCheckoutHistoryInfo is returned only if includePatronCheckoutHistoryInfo is
  set -->
<xs:element name="includePatronCheckoutHistoryInfo" type="PatronCheckoutHistoryInfo"
  minOccurs="0" maxOccurs="unbounded"/>
<!-- patronHoldInfo and feeInfo are returned only if includePatronHoldInfo is set -->
<xs:element name="patronHoldInfo" type="PatronHoldInfo" minOccurs="0"
  maxOccurs="unbounded"/>
<!-- feeInfo is returned only if includePatronFeeInfo is set -->
<xs:element name="feeInfo" type="FeeInfo" minOccurs="0" maxOccurs="unbounded"/>
<!-- patronStatusInfo is returned only if includePatronStatusInfo is true -->
<xs:element name="patronStatusInfo" type="PatronStatusInfo" minOccurs="0" maxOccurs="1"/>
<xs:element name="groupInfo" type="GroupInfo" minOccurs="0" maxOccurs="1"/>
<xs:element name="userSuspensionInfo" type="UserSuspensionInfo" minOccurs="0"
  maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:complexType name="PatronInfo">
  <xs:sequence>
    <xs:element name="userKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
    <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="alternativeID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="webAuthID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="groupID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="groupDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="displayName" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="birthDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="patronLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="patronLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="department" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <!-- preferredLanguage is returned as a ISO 639-1 language code and ISO 3166-1 country code
      if specified. (ex: en, en_US) -->
    <!-- If these values have not been set in the Language policy, the Language Policy ID will be
      returned instead. -->
    <xs:element name="preferredLanguage" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GroupMemberFeeInfo">
  <xs:complexContent>
    <xs:extension base="FeeInfo">
      <xs:sequence>
        <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="userName" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>

```

```

</xs:complexType>

<xs:complexType name="FeeInfo">
  <xs:sequence>
    <xs:element name="billNumber" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="billReasonID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="billReasonDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="amount" type="common:Money" minOccurs="1" maxOccurs="1"/>
    <xs:element name="taxAmount" type="common:Money" minOccurs="0" maxOccurs="1"/>
    <!-- amountOutstanding is the portion of amount + taxAmount that is not yet paid -->
    <xs:element name="amountOutstanding" type="common:Money" minOccurs="1" maxOccurs="1"/>
    <!-- billLibraryID is returned only if config.multi -->
    <xs:element name="billLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="billLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="dateBilled" type="xs:date" minOccurs="1" maxOccurs="1"/>
    <xs:element name="dateCreated" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="numberOfPaymentsMade" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <!-- lastPaymentDate is returned only if numberOfPaymentsMade is greater than 0 -->
    <xs:element name="lastPaymentDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <!-- feePaymentInfo is returned only if includePatronFeeInfo includes PAYMENT -->
    <xs:element name="feePaymentInfo" type="FeePaymentInfo" minOccurs="0" maxOccurs="50"/>
    <!-- feeItemInfo is returned only if the fee is associated with an item (for example,
         an overdue fine) -->
    <xs:element name="feeItemInfo" type="FeeItemInfo" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FeePaymentInfo">
  <xs:sequence>
    <xs:element name="transactionID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="paymentLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="paymentLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="paymentDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="paymentAmount" type="common:Money" minOccurs="1" maxOccurs="1"/>
    <xs:element name="paymentTypeID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="paymentTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FeeItemInfo">
  <xs:sequence>
    <xs:element name="titleKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
    <!-- itemID and itemTypeDescription are returned only if library is configured for
         accession -->
    <xs:element name="itemID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="callNumber" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="copyNumber" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemLibraryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="itemLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="checkoutDate" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

```

    <xs:element name="dueDate" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
    <xs:element name="recallDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="renewedDate" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
    <xs:element name="circulationRuleID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="circulationRuleDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PatronAddressInfo">
  <xs:sequence>
    <xs:element name="primaryAddress" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Address1Info" type="common:AddressInfo" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Address2Info" type="common:AddressInfo" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Address3Info" type="common:AddressInfo" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PatronCirculationInfo">
  <xs:sequence>
    <xs:element name="numberOfCheckouts" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="numberOfClaimsReturned" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="numberOfBookings" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="numberOfRequests" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="numberOfUnansweredRequests" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="numberOfHolds" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="numberOfAvailableHolds" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="numberOfFees" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="estimatedFines" type="common:Money" minOccurs="1" maxOccurs="1"/>
    <xs:element name="estimatedOverdues" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="creditBalance" type="common:Money" minOccurs="0" maxOccurs="1"/>
    <!-- checkoutHistoryRule is returned if loan history is configured -->
    <xs:element name="checkoutHistoryRule" type="circulation:CheckoutHistoryRule" minOccurs="0"
      maxOccurs="1"/>
    <xs:choice minOccurs="1" maxOccurs="1">
      <xs:element name="numberOfCheckoutsAllowed" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <!-- setting numberOfCheckoutsAllowedUnlimited to false will throw an illegal argument
        exception -->
      <xs:element name="numberOfCheckoutsAllowedUnlimited" type="xs:boolean" minOccurs="1"
        maxOccurs="1" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GroupMemberCheckoutInfo">
  <xs:complexContent>
    <xs:extension base="PatronCheckoutInfo">
      <xs:sequence>
        <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="userName" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PatronCheckoutInfo">
  <xs:sequence>
    <xs:element name="titleKey" type="xs:long" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="callNumber" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="copyNumber" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="pieces" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="numberOfPiecesReturned" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <!-- checkoutLibrary is the library where the item was checked out.
         itemLibrary is the item's owning library. These are not returned
         in a single-library configuration. -->
    <xs:element name="checkoutLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="checkoutLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemTypeID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <!-- reserveCollection is returned only if the item is on academic reserve. -->
    <xs:element name="reserveCollectionID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="reserveCollectionDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="checkoutDate" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    <xs:element name="dueDate" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    <!-- recallDate is returned only if the checkout has been recalled. -->
    <xs:element name="recallDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <!-- date item is due after being recalled -->
    <xs:element name="recallDueDate" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
    <xs:element name="recallNoticesSent" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <!-- lastRenewedDate is returned only if the checkout has been renewed. -->
    <xs:element name="lastRenewedDate" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
    <xs:element name="renewals" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="unseenRenewals" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:choice minOccurs="0" maxOccurs="1">
      <xs:element name="renewalsRemaining" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <!-- setting renewalsRemainingUnlimited to false will throw an illegal argument
           exception -->
      <xs:element name="renewalsRemainingUnlimited" type="xs:boolean" minOccurs="1"
        maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:choice minOccurs="0" maxOccurs="1">
    <xs:element name="unseenRenewalsRemaining" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <!-- setting unseenRenewalsRemainingUnlimited to false will throw an illegal argument
         exception -->
    <xs:element name="unseenRenewalsRemainingUnlimited" type="xs:boolean" minOccurs="1"
      maxOccurs="1"/>
  </xs:choice>
  <xs:element name="overdue" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
  <xs:element name="overdueNoticesSent" type="xs:int" minOccurs="1" maxOccurs="1"/>

```

```

<!-- dateOfLastOverdueNotice is returned only if not NEVER. -->
<xs:element name="dateOfLastOverdueNotice" type="xs:date" minOccurs="0" maxOccurs="1"/>
<!-- fine of $0.00 is not returned. Note this amount includes both any
      outstanding bills on the checkout plus the estimated fine that would accrue
      if the item were returned at this moment. -->
<xs:element name="fine" type="common:Money" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="PatronCheckoutHistoryInfo">
  <xs:sequence>
    <xs:element name="itemID" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="callNumber" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="displayableCallNumber" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="copyNumber" type="xs:int" minOccurs="1" maxOccurs="1" />
    <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1" />
    <!--
      checkoutLibrary is the library where the item was checked out. itemLibrary is the item's
      owning library. These are not returned in a single-library configuration.
    -->
    <xs:element name="checkoutLibraryID" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="checkoutLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="itemLibraryID" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="itemLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="itemTypeID" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="itemTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <!-- reserveCollection is returned only if the item is on academic reserve. -->
    <xs:element name="reserveCollectionID" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="reserveCollectionDescription" type="xs:string" minOccurs="0"
      maxOccurs="1" />
    <xs:element name="checkoutDate" type="xs:dateTime" minOccurs="1" maxOccurs="1" />
    <xs:element name="dueDate" type="xs:dateTime" minOccurs="1" maxOccurs="1" />
    <xs:element name="circulationRuleID" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="circulationRuleDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <!-- recallDate is returned only if the checkout has been recalled. -->
    <xs:element name="recallDate" type="xs:date" minOccurs="0" maxOccurs="1" />
    <xs:element name="recallNoticesSent" type="xs:int" minOccurs="0" maxOccurs="1" />
    <!-- lastRenewedDate is returned only if the checkout has been renewed. -->
    <xs:element name="lastRenewedDate" type="xs:dateTime" minOccurs="0" maxOccurs="1" />
    <xs:element name="renewals" type="xs:int" minOccurs="0" maxOccurs="1" />
    <xs:element name="wasOverdue" type="xs:boolean" minOccurs="1" maxOccurs="1" />
    <xs:element name="overdueNoticesSent" type="xs:int" minOccurs="0" maxOccurs="1" />
    <!-- dateOfLastOverdueNotice is returned only if not NEVER. -->
    <xs:element name="dateOfLastOverdueNotice" type="xs:date" minOccurs="0" maxOccurs="1" />
    <!--
      fine of $0.00 is not returned. Note this amount includes both any outstanding bills on the
      checkout plus the estimated fine that would accrue if the item were returned at this moment.
    -->
    <xs:element name="fine" type="common:Money" minOccurs="0" maxOccurs="1" />
    <xs:element name="dateLoanHistoryDischarged" type="xs:dateTime" minOccurs="0" maxOccurs="1" />
    <xs:element name="loanHistoryDischargingLibraryID" type="xs:string" minOccurs="0"
      maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

```



```

    <xs:element name="loanHistoryDischargingLibraryDescription" type="xs:string" minOccurs="0"
      maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GroupMemberHoldInfo">
  <xs:complexContent>
    <xs:extension base="PatronHoldInfo">
      <xs:sequence>
        <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="userName" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PatronHoldInfo">
  <xs:sequence>
    <xs:element name="holdKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
    <xs:element name="titleKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
    <xs:element name="itemID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="callNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="displayableCallNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="blanketHoldID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="blanketHoldCopiesNeeded" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="blanketHoldCopiesReceived" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="expiresDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <!-- holdStatus: 1=Inactive 2=Active -->
    <xs:element name="holdStatus" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <!-- Each holdInactiveReasonID will be associated with a holdInactiveType -->
    <xs:element name="holdInactiveReasonID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="holdInactiveReasonDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="holdInactiveType" type="circulation:HoldInactiveType" minOccurs="0"
      maxOccurs="1" />
    <!-- itemID and itemType might not be present if hold was placed on an order. -->
    <xs:element name="itemTypeID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="title" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="pickupLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="pickupLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="placedDate" type="xs:date" minOccurs="1" maxOccurs="1"/>
    <xs:element name="mailServiceID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="mailServiceDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <!-- reserve indicates if the hold is for an academic reserve collection. -->
    <xs:element name="reserve" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
    <xs:element name="recallStatus" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <!-- available and intransit are mutually exclusive. Both may be false, but
      only one may be true. intransit means that the hold was filled at a
      library other than the pickup library, so the item is in transit to
      the pickup library and will become available when it gets there. -->

```

```

<xs:element name="available" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
<xs:element name="intransit" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
<!-- availableDate and availableExpiresDate are returned only if not NEVER. -->
<xs:element name="availableDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="availableExpiresDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="queuePosition" type="xs:int" minOccurs="1" maxOccurs="1"/>
<xs:element name="queueLength" type="xs:int" minOccurs="1" maxOccurs="1"/>
<xs:element name="suspendStartDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="suspendEndDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<!-- holdPlacedWithOverride will not be returned in a "traditional holds"
configuration. -->
<xs:element name="holdPlacedWithOverride" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
<!-- The order information elements are present only if the hold was placed on
"on order" materials. -->
<xs:element name="orderFiscalCycle" type="xs:int" minOccurs="0" maxOccurs="1"/>
<xs:element name="orderID" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="orderLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="orderLibraryDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="orderLine" type="xs:string" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>

<xs:element name="ModifyMyHoldRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="holdKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
      <xs:element name="expiresDate" type="common:NillableDate" minOccurs="0" maxOccurs="1"/>
      <xs:choice minOccurs="0" maxOccurs="1">
        <!-- Only if config.multi is true; is a multilibrary system -->
        <xs:element name="holdPickupLibraryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <!-- Only if books by mail is turned on; item must also be qualified for books
by mail -->
        <xs:element name="mailDeliveryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="ModifyMyHoldResponse" type="xs:boolean"/>

<xs:element name="SuspendMyHoldRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="holdKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
      <!-- Must be after the current date and no later than hold expiration date -->
      <xs:element name="suspendStartDate" type="xs:date" minOccurs="1" maxOccurs="1"/>
      <!-- Must not be later than the hold expiration date -->
      <xs:element name="suspendEndDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="SuspendMyHoldResponse" type="xs:boolean"/>

```

```

<xs:element name="UnsuspendMyHoldRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="holdKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="UnsuspendMyHoldResponse" type="xs:boolean"/>

<xs:complexType name="PatronStatusInfo">
  <xs:sequence>
    <xs:element name="datePrivilegeExpires" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="statusID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="statusDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="statusType" type="circulation:StatusType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="statusMessage" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="routingAllowed" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
    <xs:element name="userHasAlerts" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<!-- Online User Registration Feature must be configured on ILS -->
<xs:element name="CreateSelfRegisteredPatronRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstName" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="middleName" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="lastName" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="nameSuffix" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="preferredName" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="socialSecurityNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="birthDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
      <xs:element name="pin" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="street" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="apartment" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="city" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="state" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="postalCode" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="homePhone" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="emailAddress" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="patronLibraryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- CreateSelfRegisteredPatronResponse will be the userID of the Patron record created -->
<xs:element name="CreateSelfRegisteredPatronResponse" type="xs:string"/>

<xs:element name="ChangeMyPINRequest">
  <xs:complexType>
    <xs:sequence>

```

```

        <xs:element name="currentPIN" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="newPIN" type="xs:string" maxOccurs="1" minOccurs="1"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="ChangeMyPINResponse" type="xs:boolean"/>

<xs:element name="LookupMyListsRequest"/>
<xs:element name="LookupMyListsResponse">
    <xs:complexType>
        <xs:sequence>
            <!--The myList element only contains identifying information in this response-->
            <!-- The MyListEntry will be empty -->
            <!-- LookupMyListRequest should be used to get the list of Titles -->
            <xs:element name="myList" type="MyList" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="LookupMyListRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="myListID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="includeMyListTitleInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="LookupMyListResponse" type="MyListInfo"/>

<xs:element name="GetMyListRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="myListID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="GetMyListResponse" type="MyList"/>

<xs:element name="SaveMyListRequest" type="MyList" />
<xs:element name="SaveMyListResponse" type="MyList" />

<xs:element name="DeleteMyListRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="myListID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="DeleteMyListResponse" type="xs:boolean"/>

```

```

<xs:element name="ChangeMyAlternativeIDRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="newAlternativeID" type="xs:string" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="ChangeMyAlternativeIDResponse" type="xs:boolean" />

<xs:element name="enableMyCheckoutHistoryRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="enable" type="xs:boolean" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="enableMyCheckoutHistoryResponse" type="xs:boolean" />

<xs:element name="ActivateOnlineRegisteredUserRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="assignedUserID" type="xs:string" minOccurs="1" maxOccurs="1" />
      <xs:element name="newUserID" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="newAltID" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="newWebAuthID" type="xs:string" minOccurs="0" maxOccurs="1" />
      <!-- required if newWebAuthID is present -->
      <xs:element name="webAuthOverride" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="pin" type="xs:string" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="ActivateOnlineRegisteredUserResponse" type="xs:boolean" />

<xs:element name="PlaceMyHoldRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:choice minOccurs="1">
        <xs:element name="itemID" type="xs:string" minOccurs="0" maxOccurs="1" />
        <xs:element name="titleKey" type="xs:long" minOccurs="0" maxOccurs="1" />
      </xs:choice>
      <xs:element name="callNumber" type="xs:string" minOccurs="0" maxOccurs="1" />
      <!-- if not present, holdType defaults to TITLE on the server -->
      <xs:element name="holdType" type="circulation:HoldType" minOccurs="0" maxOccurs="1" />
      <!-- if not present, holdRange defaults to GROUP -->
      <xs:element name="holdRange" type="circulation:HoldRange" minOccurs="0" maxOccurs="1" />
      <xs:choice minOccurs="0" maxOccurs="1">
        <xs:element name="pickupLibraryID" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="mailServiceID" type="xs:string" minOccurs="1" maxOccurs="1" />
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

    <!-- if not present, expiresDate will be calculated by the ILS -->
    <xs:element name="expiresDate" type="common:NillableDate" minOccurs="0" maxOccurs="1"/>
    <xs:element name="suspendStartDate" type="xs:date" minOccurs="0" maxOccurs="1" />
    <xs:element name="suspendEndDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="comment" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="PlaceMyHoldResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="holdKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
      <xs:element name="estimatedFee" type="common:Money" minOccurs="0" maxOccurs="1" />
      <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="itemAvailable" type="xs:boolean" minOccurs="1" maxOccurs="1" />
      <xs:element name="callNumber" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="itemID" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="dateHoldExpires" type="xs:date" minOccurs="0" maxOccurs="1" />
      <xs:element name="dateHoldExpiresOnShelf" type="xs:date" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- objects -->

<xs:complexType name="MyList">
  <xs:complexContent>
    <xs:extension base="common:Entity">
      <xs:sequence>
        <xs:element name="myListID" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="description" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="activeList" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        <xs:element name="myListEntry" type="MyListEntry" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MyListInfo">
  <xs:sequence>
    <xs:element name="myListID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="description" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="activeList" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
    <xs:element name="myListEntryInfo" type="MyListEntryInfo" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="MyListEntry">
  <xs:complexContent>
    <xs:extension base="common:Entity">

```

```

    <xs:sequence>
      <xs:element name="titleKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="MyListEntryInfo">
  <xs:sequence>
    <xs:element name="titleKey" type="xs:long" minOccurs="1" maxOccurs="1"/>
    <xs:element name="myListTitleInfo" type="MyListTitleInfo" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="MyListTitleInfo">
  <xs:sequence>
    <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="title" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="thirdBriefEntry" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="fourthBriefEntry" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="UserSuspensionInfo">
  <xs:sequence>
    <xs:element name="dateSuspended" type="xs:date" minOccurs="1" maxOccurs="1" />
    <xs:element name="dateSuspensionEnds" type="xs:date" minOccurs="1" maxOccurs="1" />
    <xs:element name="suspensionKey" type="xs:long" minOccurs="1" maxOccurs="1" />
    <xs:element name="linkedSuspensionKey" type="xs:long" minOccurs="1" maxOccurs="1" />
    <xs:element name="suspensionLibraryID" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="suspensionLibraryDescription" type="xs:string" minOccurs="1"
      maxOccurs="1" />
    <xs:element name="suspensionReasonID" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="suspensionReasonDescription" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="suspensionType" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="numberOfUnits" type="xs:int" minOccurs="1" maxOccurs="1" />
    <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="itemID" type="xs:string" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GroupInfo">
  <xs:sequence>
    <xs:element name="GroupInfo" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="GroupInfo" type="xs:int" minOccurs="1" maxOccurs="1" />
    <xs:element name="GroupInfo" type="xs:date" minOccurs="1" maxOccurs="1" />
    <xs:element name="groupStatusID" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="groupStatusDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="groupStatusMessage" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="userGroupResponsibilityPolicyID" type="xs:string" minOccurs="1"
      maxOccurs="1" />
    <xs:element name="userGroupResponsibilityPolicyDescription" type="xs:string" minOccurs="1"
      maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

```

```

<xs:element name="userGroupResponsibilityLevelID" type="xs:string" minOccurs="1"
  maxOccurs="1" />
<xs:element name="userGroupResponsibilityTypeID" type="xs:string" minOccurs="1"
  maxOccurs="1" />
<xs:element name="noticeMaster" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="allowedDisplayGroupCheckouts" type="xs:boolean" minOccurs="1"
  maxOccurs="1" />
<xs:element name="allowedDisplayGroupHolds" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="allowedCheckoutGroupHolds" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="allowedCancelGroupHolds" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="allowedDisplayGroupFees" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="allowedPayGroupFees" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="allowedGroupChoice" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<!-- groupCheckoutInfo is returned only if includePatronCheckoutInfo is set and the requesting
  user has permission -->
<xs:element name="groupMemberCheckoutInfo" type="GroupMemberCheckoutInfo" minOccurs="0"
  maxOccurs="unbounded" />
<!-- groupHoldInfo are returned only if includePatronHoldInfo is set and the requesting user
  has permission -->
<xs:element name="groupMemberHoldInfo" type="GroupMemberHoldInfo" minOccurs="0"
  maxOccurs="unbounded" />
<!-- groupFeeInfo is returned only if includeFeeInfo is set and the requesting user has
  permission -->
<xs:element name="groupMemberFeeInfo" type="GroupMemberFeeInfo" minOccurs="0"
  maxOccurs="unbounded" />
<xs:element name="groupMemberInfo" type="GroupMemberInfo" minOccurs="0"
  maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>

<xs:complexType name="GroupMemberInfo">
  <xs:sequence>
    <xs:element name="userName" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="userGroupResponsibilityPolicyID" type="xs:string" minOccurs="0"
      maxOccurs="1" />
    <xs:element name="userGroupResponsibilityPolicyDescription" type="xs:string" minOccurs="0"
      maxOccurs="1" />
    <xs:element name="noticeMaster" type="xs:boolean" minOccurs="1" maxOccurs="1" />
    <xs:element name="userStatusID" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="userStatusDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="userStatusMessage" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="userNumberOfBills" type="xs:int" minOccurs="0" maxOccurs="1" />
    <xs:element name="userEstimatedFines" type="common:Money" minOccurs="0" maxOccurs="1" />
    <xs:element name="userNumberOfOverdues" type="xs:int" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:element name="LookupPatronInfoRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:choice minOccurs="1" maxOccurs="1">
        <xs:element name="userID" type="xs:string" minOccurs="1" maxOccurs="1"/>

```



```

    <xs:element name="alternateID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="webAuthID" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:choice>
  <!-- All boolean "include" parameters are assumed false if not present . -->
  <xs:element name="includePatronInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
  <xs:element name="includePatronCirculationInfo" type="xs:boolean" minOccurs="0"
    maxOccurs="1"/>
  <xs:element name="includePatronAddressInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
  <xs:element name="includePatronCheckoutInfo" type="circulation:CheckoutInfoFilter"
    minOccurs="0" maxOccurs="1"/>
  <xs:element name="includePatronCheckoutHistoryInfo" type="xs:boolean" minOccurs="0"
    maxOccurs="1"/>
  <xs:element name="includePatronHoldInfo" type="circulation:HoldInfoFilter" minOccurs="0"
    maxOccurs="1"/>
  <xs:element name="includeFeeInfo" type="circulation:FeeInfoFilter" minOccurs="0"
    maxOccurs="1"/>
  <xs:element name="includePatronStatusInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
  <xs:element name="includeUserGroupInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
  <xs:element name="includeUserSuspensionInfo" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="LookupPatronInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <!-- patronInfo is returned only if includePatronInfo is true -->
      <xs:element name="patronInfo" type="PatronInfo" minOccurs="0" maxOccurs="1" />
      <!-- PatronCirculationInfo is returned only if includePatronCirculationInfo is true -->
      <xs:element name="patronCirculationInfo" type="PatronCirculationInfo" minOccurs="0"
        maxOccurs="1" />
      <!-- Patron addresses are returned only if includePatronAddressInfo is true -->
      <xs:element name="patronAddressInfo" type="PatronAddressInfo" minOccurs="0" maxOccurs="1" />
      <!-- patronCheckoutInfo is returned only if includePatronCheckoutInfo is set -->
      <xs:element name="patronCheckoutInfo" type="PatronCheckoutInfo" minOccurs="0"
        maxOccurs="unbounded" />
      <!-- patronCheckoutHistoryInfo is returned only if includePatronCheckoutHistoryInfo is
        set -->
      <xs:element name="patronCheckoutHistoryInfo" type="PatronCheckoutHistoryInfo" minOccurs="0"
        maxOccurs="unbounded" />
      <!-- patronHoldInfo and feeInfo are returned only if includePatronHoldInfo is set -->
      <xs:element name="patronHoldInfo" type="PatronHoldInfo" minOccurs="0"
        maxOccurs="unbounded" />
      <!-- feeInfo is returned only if includePatronFeeInfo is set -->
      <xs:element name="feeInfo" type="FeeInfo" minOccurs="0" maxOccurs="unbounded" />
      <!-- patronStatusInfo is returned only if includePatronStatusInfo is true -->
      <xs:element name="patronStatusInfo" type="PatronStatusInfo" minOccurs="0" maxOccurs="1" />
      <xs:element name="groupInfo" type="GroupInfo" minOccurs="0" maxOccurs="1" />
      <xs:element name="userSuspensionInfo" type="UserSuspensionInfo" minOccurs="0"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

## Admin service

```

<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.sirsidynix.com/symws/admin"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://schemas.sirsidynix.com/symws/admin">

  <xs:element name="LookupILSConfigurationRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:choice minOccurs="0" maxOccurs="1">
          <xs:element name="configGroup">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="ALL"/>
                <xs:enumeration value="MODULES"/>
                <xs:enumeration value="SYSTEM"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="configName" type="xs:string"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="LookupILSConfigurationResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="configInfo" type="ILSConfigurationInfo" minOccurs="1"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Generic policy operations -->
  <xs:element name="LookupPolicyListRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="policyType" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="LookupPolicyListResponse" type="PolicyList"/>

  <xs:element name="GetPolicyRequest">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="PolicyRequest">

```

```

        <xs:sequence>
          <xs:element name="policyType" type="xs:string" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

<xs:element name="GetPolicyResponse">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="PolicyInfo">
        <xs:sequence>
          <xs:element name="policyField" type="PolicyField" minOccurs="0"
            maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

<!-- Specific policy operations -->
<xs:element name="LookupAccessLevelPolicyListRequest"/>

<xs:element name="LookupAccessLevelPolicyListResponse" type="PolicyList"/>

<xs:element name="GetAccessLevelPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetAccessLevelPolicyResponse" type="AccessLevelPolicy"/>

<xs:element name="LookupAccessRestrictionPolicyListRequest"/>

<xs:element name="LookupAccessRestrictionPolicyListResponse" type="PolicyList"/>

<xs:element name="GetAccessRestrictionPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetAccessRestrictionPolicyResponse" type="PolicyInfo"/>

<xs:element name="LookupCitizenshipPolicyListRequest"/>

<xs:element name="LookupCitizenshipPolicyListResponse" type="PolicyList"/>

<xs:element name="GetCitizenshipPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetCitizenshipPolicyResponse" type="PolicyInfo"/>

<xs:element name="LookupClearancePolicyListRequest"/>

<xs:element name="LookupClearancePolicyListResponse" type="PolicyList"/>

<xs:element name="GetClearancePolicyRequest" type="PolicyRequest"/>

<xs:element name="GetClearancePolicyResponse" type="ClearancePolicy"/>

```

```

<xs:element name="LookupItemTypePolicyListRequest"/>

<xs:element name="LookupItemTypePolicyListResponse" type="PolicyList"/>

<xs:element name="GetItemTypePolicyRequest" type="PolicyRequest"/>

<xs:element name="GetItemTypePolicyResponse" type="ItemTypePolicy"/>

<xs:element name="LookupLanguagePolicyListRequest"/>

<xs:element name="LookupLanguagePolicyListResponse" type="PolicyList"/>

<xs:element name="GetLanguagePolicyRequest" type="PolicyRequest"/>

<xs:element name="GetLanguagePolicyResponse" type="LanguagePolicy"/>

<xs:element name="LookupLibraryPolicyListRequest"/>

<xs:element name="LookupLibraryPolicyListResponse" type="PolicyList"/>

<xs:element name="GetLibraryPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetLibraryPolicyResponse" type="LibraryPolicy"/>

<xs:element name="LookupLocationPolicyListRequest"/>

<xs:element name="LookupLocationPolicyListResponse" type="PolicyList"/>

<xs:element name="GetLocationPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetLocationPolicyResponse" type="LocationPolicy"/>

<xs:element name="LookupNeedToKnowPolicyListRequest"/>

<xs:element name="LookupNeedToKnowPolicyListResponse" type="PolicyList"/>

<xs:element name="GetNeedToKnowPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetNeedToKnowPolicyResponse" type="PolicyInfo"/>

<xs:element name="LookupUserLookupPolicyListRequest"/>

<xs:element name="LookupUserLookupPolicyListResponse" type="PolicyList"/>

<xs:element name="GetUserLookupPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetUserLookupPolicyResponse" type="UserLookupPolicy"/>

<xs:element name="LookupUserSortPolicyListRequest"/>

<xs:element name="LookupUserSortPolicyListResponse" type="PolicyList"/>

```

```

<xs:element name="GetUserSortPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetUserSortPolicyResponse" type="UserSortPolicy"/>

<xs:element name="LookupBillReasonPolicyListRequest"/>

<xs:element name="LookupBillReasonPolicyListResponse" type="PolicyList"/>

<xs:element name="GetBillReasonPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetBillReasonPolicyResponse" type="BillReasonPolicy"/>

<xs:element name="LookupCreditReasonPolicyListRequest"/>

<xs:element name="LookupCreditReasonPolicyListResponse" type="PolicyList"/>

<xs:element name="GetCreditReasonPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetCreditReasonPolicyResponse" type="CreditReasonPolicy"/>

<xs:element name="LookupPaymentTypePolicyListRequest"/>

<xs:element name="LookupPaymentTypePolicyListResponse" type="PolicyList"/>

<xs:element name="GetPaymentTypePolicyRequest" type="PolicyRequest"/>

<xs:element name="GetPaymentTypePolicyResponse" type="PaymentTypePolicy"/>

<xs:element name="LookupNativeCurrencyRequest"/>

<xs:element name="LookupNativeCurrencyResponse" type="xs:string"/>

<xs:element name="LookupMailServicePolicyListRequest"/>

<xs:element name="LookupMailServicePolicyListResponse" type="PolicyList"/>

<xs:element name="GetMailServicePolicyRequest" type="PolicyRequest"/>

<xs:element name="GetMailServicePolicyResponse" type="MailServicePolicy"/>

<xs:element name="IsPatronSelfRegistrationEnabledRequest"/>

<xs:element name="IsPatronSelfRegistrationEnabledResponse" type="xs:boolean"/>

<xs:element name="IsUserGroupsEnabledRequest" />

<xs:element name="IsUserGroupsEnabledResponse" type="xs:boolean" />

<!-- details of a request type policy -->
<xs:element name="GetRequestTypePolicyRequest" type="PolicyRequest"/>

<xs:element name="GetRequestTypePolicyResponse" type="RequestTypePolicy"/>

```

```

<!-- list of request type policies -->
<xs:element name="LookupRequestTypePolicyListRequest"/>

<xs:element name="LookupRequestTypePolicyListResponse" type="PolicyList"/>

<!-- details of a list of request type policies based on user level access -->
<xs:element name="GetRequestListPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetRequestListPolicyResponse" type="RequestListPolicy" />

<!-- list of request type policies -->
<xs:element name="LookupRequestListPolicyListRequest"/>

<xs:element name="LookupRequestListPolicyListResponse" type="PolicyList"/>

<!-- details of a request status policy -->
<xs:element name="GetRequestStatusPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetRequestStatusPolicyResponse" type="RequestStatusPolicy"/>

<!-- list of request status policies -->
<xs:element name="LookupRequestStatusPolicyListRequest"/>

<xs:element name="LookupRequestStatusPolicyListResponse" type="PolicyList"/>

<xs:element name="IsRestrictedAccessRequest" />

<xs:element name="IsRestrictedAccessResponse" type="xs:boolean" />

<xs:element name="GetLibraryGroupPolicyRequest" type="PolicyRequest" />

<xs:element name="GetLibraryGroupPolicyResponse" type="LibraryGroupPolicy" />

<xs:element name="LookupLibraryGroupListRequest" />

<xs:element name="LookupLibraryGroupListResponse" type="PolicyList" />

<xs:element name="GetSearchLibraryGroupPolicyRequest" type="PolicyRequest" />

<xs:element name="GetSearchLibraryGroupPolicyResponse" type="SearchLibraryGroupPolicy" />

<xs:element name="LookupSearchLibraryGroupListRequest" />

<xs:element name="LookupSearchLibraryGroupListResponse" type="PolicyList" />

<xs:element name="IsCheckoutHistoryEnabledRequest" />

<xs:element name="IsCheckoutHistoryEnabledResponse" type="xs:boolean" />

<xs:element name="GetGroupResponsibilityPolicyRequest" type="PolicyRequest" />

<xs:element name="GetGroupResponsibilityPolicyResponse" type="GroupResponsibilityPolicy" />

```

```

<xs:element name="LookupGroupResponsibilityPolicyListRequest" />

<xs:element name="LookupGroupResponsibilityPolicyListResponse" type="PolicyList" />

<xs:element name="LookupBestSellersPolicyListRequest" />

<xs:element name="LookupBestSellersPolicyListResponse" type="PolicyList" />

<xs:element name="GetBestSellerPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetBestSellerPolicyResponse" type="BestSellerPolicy"/>

<xs:element name="LookupRecommendedReadingPolicyListRequest" />

<xs:element name="LookupRecommendedReadingPolicyListResponse" type="PolicyList" />

<xs:element name="GetRecommendedReadingPolicyRequest" type="PolicyRequest"/>

<xs:element name="GetRecommendedReadingPolicyResponse" type="RecommendedReadingPolicy" />

<xs:element name="IsMailDeliveryEnabledRequest"/>

<xs:element name="IsMailDeliveryEnabledResponse" type="xs:boolean"/>

<xs:element name="IsAccessionEnabledRequest"/>

<xs:element name="IsAccessionEnabledResponse" type="xs:boolean"/>

<!-- Type definitions -->
<xs:complexType name="ILSConfigurationInfo">
  <xs:sequence>
    <xs:element name="name" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="value" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PolicyRequest">
  <xs:sequence>
    <xs:element name="policyID" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PolicyList">
  <xs:sequence>
    <xs:element name="policyInfo" type="PolicyInfo" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PolicyInfo">
  <xs:sequence>
    <xs:element name="policyNumber" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="policyID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="policyDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

```

    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PolicyField">
    <xs:sequence>
      <xs:element name="policyFieldCode" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="policyFieldValue" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AccessLevelPolicy">
    <xs:complexContent>
      <xs:extension base="PolicyInfo">
        <xs:sequence>
          <xs:element name="navigateAllowed" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
          <xs:element name="searchAllowed" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
          <xs:element name="printAllowed" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
          <xs:element name="downloadAllowed" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
          <xs:element name="createAllowed" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
          <xs:element name="modifyAllowed" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
          <xs:element name="deleteAllowed" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ClearancePolicy">
    <xs:complexContent>
      <xs:extension base="PolicyInfo">
        <xs:sequence>
          <xs:element name="clearanceLevel" minOccurs="1" maxOccurs="1">
            <xs:simpleType>
              <xs:restriction base="xs:int">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="99"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="purgePeriod" minOccurs="1" maxOccurs="1">
            <xs:simpleType>
              <xs:restriction base="xs:int">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="32000"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="chargeReceiptRequired" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
          <xs:element name="certificateOfDestructionRequired" type="xs:boolean" minOccurs="1"
            maxOccurs="1"/>
          <xs:element name="classified" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```



```

</xs:complexType>

<xs:complexType name="ItemTypePolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="floating" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
        <xs:element name="holdThreshold" type="xs:int" minOccurs="0" maxOccurs="1"/>
        <xs:element name="bookingProfileID" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="reservationProfileID" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="multiCirc" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="LanguagePolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="marcLanguageCode" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="defaultLanguage" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="writeDirection" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="directory" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="locale" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="article" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="charset" type="xs:string" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="LibraryPolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="userID" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="loanPeriod" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="accrueFine" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="closedDate" type="xs:date" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="closedEndDate" type="xs:date" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="holdLocationID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="closedSunday" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="closedMonday" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="closedTuesday" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="closedWednesday" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="closedThursday" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="closedFriday" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="closedSaturday" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="textIndexName" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="NUCSymbol" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="acquisitionsDisplayLibraryID" type="xs:string" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

<xs:element name="acquisitionsMaintenanceLibraryID" type="xs:string" minOccurs="0"
  maxOccurs="unbounded"/>
<xs:element name="holdPermissionLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="holdGroupLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="libraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="serialDisplayLibraryID" type="xs:string" minOccurs="0"
  maxOccurs="unbounded"/>
<xs:element name="serialMaintenanceLibraryID" type="xs:string" minOccurs="0"
  maxOccurs="unbounded"/>
<xs:element name="circDisplayLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="circMaintenanceLibraryID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="catDisplayLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="catMaintenanceLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="userDisplayLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="userMaintenanceLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="billDisplayLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="billMaintenanceLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="holdAvailableLibraryID" type="xs:string" minOccurs="0"
  maxOccurs="unbounded"/>
<xs:element name="SAN" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="closedDatesReport" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
<xs:element name="libraryNumber" type="xs:int" minOccurs="1" maxOccurs="1"/>
<xs:element name="callNumberMaintenanceLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="itemMaintenanceLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="marcHoldingsMaintenanceLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="holdExpireDays" type="xs:int" minOccurs="1" maxOccurs="1"/>
<xs:element name="holdAvailableExpireDays" type="xs:int" minOccurs="1" maxOccurs="1"/>
<xs:element name="OCLCCode" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="libraryHourMaintenanceLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="openUrlResolver" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="patronCreditMaintenanceLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="mailCenterLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="mailCenterLocationID" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="mailHoldLibraryGroupID" type="xs:string" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="floatingLibraryGroupID" type="xs:string" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:extension>

```

```

    </xs:complexContent>
</xs:complexType>

<xs:complexType name="LocationPolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="holdable" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="type" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="shadow" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
        <xs:element name="reshelvingPeriod" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="reshelvingLocation" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="available" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MailServicePolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="shipTime" type="xs:int" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="UserLookupPolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="searchID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="UserSortPolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="searchID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="BillReasonPolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="associatedItem" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <xs:element name="billReasonType" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="taxRate" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="taxBase" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="CreditReasonPolicy">
    <xs:complexContent>
        <xs:extension base="PolicyInfo">
            <xs:sequence>
                <xs:element name="type" type="xs:string" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="PaymentTypePolicy">
    <xs:complexContent>
        <xs:extension base="PolicyInfo">
            <xs:sequence>
                <xs:element name="kind" type="xs:string" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="RequestTypePolicy">
    <xs:complexContent>
        <xs:extension base="PolicyInfo">
            <xs:sequence>
                <xs:element name="requestFormatID" type="xs:string" minOccurs="1" maxOccurs="1"/>
                <xs:element name="responseFormatID" type="xs:string" minOccurs="1" maxOccurs="1"/>
                <xs:element name="userIDRequired" type="RequiredType" minOccurs="1" maxOccurs="1"/>
                <xs:element name="titleKeyRequired" type="RequiredType" minOccurs="1" maxOccurs="1"/>
                <xs:element name="requestTextRequired" type="RequiredType" minOccurs="1" maxOccurs="1"/>
                <xs:element name="defaultStatusID" type="xs:string" minOccurs="1" maxOccurs="1"/>
                <xs:element name="specialUse" type="SpecialUseType" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="RequestStatusPolicy">
    <xs:complexContent>
        <xs:extension base="PolicyInfo">
            <xs:sequence>
                <xs:element name="autoUpdateReplyDate" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

<xs:complexType name="lookupBestSellersPolicyList">
  <xs:sequence>
    <xs:element name="bestSellersPolicyInfo" type="BestSellersPolicyInfo" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="BestSellersPolicyInfo">
  <xs:sequence>
    <xs:element name="policyID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="policyDescription" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="copyright" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="lookupRecommendedReadingPolicyList">
  <xs:sequence>
    <xs:element name="recommendedReadingPolicyInfo" type="RecommendedReadingPolicyInfo"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RecommendedReadingPolicyInfo">
  <xs:sequence>
    <xs:element name="policyID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="policyDescription" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="note" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="RequiredType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="MANDATORY"/>
    <xs:enumeration value="OPTIONAL"/>
    <xs:enumeration value="NONE"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SpecialUseType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NONE"/>
    <xs:enumeration value="IMMEDIATE_HOLD"/>
    <xs:enumeration value="MESSAGE"/>
    <xs:enumeration value="ENHANCED_HOLD"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="RequestListPolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="requestTypeID" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <!-- request types allowed -->
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="LibraryGroupPolicy">
    <xs:complexContent>
        <xs:extension base="PolicyInfo">
            <xs:sequence>
                <xs:element name="libraryID" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="holdFill" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="SearchLibraryGroupPolicy">
    <xs:complexContent>
        <xs:extension base="PolicyInfo">
            <xs:sequence>
                <xs:element name="libraryID" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="GroupResponsibilityPolicy">
    <xs:complexContent>
        <xs:extension base="PolicyInfo">
            <xs:sequence>
                <xs:element name="responsibilityLevel" type="xs:string" minOccurs="1" maxOccurs="1" />
                <xs:element name="responsibilityType" type="xs:string" minOccurs="1" maxOccurs="1" />
                <xs:element name="allowedDisplayGroupCheckouts" type="xs:boolean" minOccurs="1"
                    maxOccurs="1" />
                <xs:element name="allowedDisplayGroupHolds" type="xs:boolean" minOccurs="1"
                    maxOccurs="1" />
                <xs:element name="allowedCheckoutGroupHolds" type="xs:boolean" minOccurs="1"
                    maxOccurs="1" />
                <xs:element name="allowedCancelGroupHolds" type="xs:boolean" minOccurs="1"
                    maxOccurs="1" />
                <xs:element name="allowedDisplayGroupFees" type="xs:boolean" minOccurs="1" maxOccurs="1"
/>
                <xs:element name="allowedPayGroupFees" type="xs:boolean" minOccurs="1" maxOccurs="1" />
                <xs:element name="allowedGroupChoice" type="xs:boolean" minOccurs="1" maxOccurs="1" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="BestSellerPolicy">
    <xs:complexContent>
        <xs:extension base="PolicyInfo">

```

```

    <xs:sequence>
      <xs:element name="bestSellerDisplay" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="RecommendedReadingPolicy">
  <xs:complexContent>
    <xs:extension base="PolicyInfo">
      <xs:sequence>
        <xs:element name="recommendedReadingDisplay" type="xs:string" minOccurs="0"
          maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

</xs:schema>

```

## Request service

```

<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.sirsidynix.com/symws/request"
  xmlns:admin="http://schemas.sirsidynix.com/symws/admin"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://schemas.sirsidynix.com/symws/request">

  <xs:import namespace="http://schemas.sirsidynix.com/symws/admin" schemaLocation="admin.xsd"/>

  <!-- messages -->

  <xs:element name="LookupRequestTemplateRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="requestTypeID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="LookupRequestTemplateResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="userIDRequired" type="admin:RequiredType" minOccurs="1" maxOccurs="1"/>
        <!-- catalogKeyRequired means either itemID, catalog key, or call number is required -->
        <xs:element name="catalogKeyRequired" type="admin:RequiredType" minOccurs="1" maxOccurs="1"/>
        <xs:element name="defaultStatusID" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="defaultStatusDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="specialUse" type="admin:SpecialUseType" minOccurs="1" maxOccurs="1"/>
        <xs:element name="requestEntryInfo" type="RequestEntryTemplateInfo" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="createRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="requestTypeID" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="userID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <!-- if accession is not configured on Symphony, then itemID cannot be used -->
      <!-- if catalogKeyRequired for the request type template is Mandatory, then catalogKey
           cannot be used -->
      <xs:choice>
        <xs:element name="userID" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="catalogKey" type="xs:long" minOccurs="0" maxOccurs="1"/>
        <xs:element name="callNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
      </xs:choice>
      <xs:element name="serviceLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="statusID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <!-- use the corresponding absolute entry number retrieved from
           LookupRequestTemplateResponse -->
      <xs:element name="requestEntry" type="RequestEntry" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="CreateRequestResponse" type="xs:string"/> <!-- Request key -->

<xs:element name="LookupRequestsRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element name="userID" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="userAltID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      </xs:choice>
      <xs:element name="requestStatusID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="serviceLibraryID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <!-- can only be used in conjunction with userID or userAltID -->
      <xs:element name="repliedToRequests" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="LookupRequestsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="RequestBriefInfo" type="RequestBriefInfo" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="LookupRequestRequest">

```



```

<xs:complexType>
  <xs:sequence>
    <xs:element name="requestKey" type="xs:string" minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="LookupRequestResponse" type="RequestInfo" />

<xs:element name="CancelRequestRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="requestKey" type="xs:string" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="CancelRequestResponse" type="xs:boolean" />

<!-- objects -->

<xs:complexType name="RequestEntryTemplateInfo">
  <xs:sequence>
    <xs:element name="absoluteEntryNumber" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="entryFormat" type="RequestEntryType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="entryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="entryDescription" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="internalEntryNumber" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="entryName" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="allowedEditCreate" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
    <xs:element name="required" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
    <xs:element name="dataType" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="minimumSize" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="maximumSize" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="defaultValue" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="entryClass" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="displayValue" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RequestInfo">
  <xs:complexContent>
    <xs:extension base="RequestBriefInfo">
      <xs:sequence>
        <xs:element name="requestEntry" type="RequestEntryInfo" minOccurs="0"
          maxOccurs="unbounded"/>
        <xs:element name="replyEntry" type="RequestEntryInfo" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

<xs:complexType name="RequestBriefInfo">
  <xs:sequence>
    <xs:element name="requestKey" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="requestTypeID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="requestTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="userID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="itemID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="statusID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="statusDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="RequestEntryType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="COMMENT"/> <!-- c -->
    <xs:enumeration value="EMAIL"/> <!-- m -->
    <xs:enumeration value="PHONE"/> <!-- h -->
    <xs:enumeration value="STAFFONLY"/> <!-- n -->
    <xs:enumeration value="USERREPLY"/> <!-- r -->
    <xs:enumeration value="Z3950"/> <!-- d -->
    <xs:enumeration value="ZIPCODE"/> <!-- z -->
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="RequestEntry">
  <xs:sequence>
    <xs:element name="absoluteEntryNumber" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="entryID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="entryData" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RequestEntryInfo">
  <xs:sequence>
    <xs:element name="entryDescription" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="entryData" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

</xs:schema>

```

## Circulation service

```

<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.sirsidynix.com/symws/circulation"
  xmlns:common="http://schemas.sirsidynix.com/symws/common"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://schemas.sirsidynix.com/symws/circulation">
  <xs:import namespace="http://schemas.sirsidynix.com/symws/common"
    schemaLocation="common.xsd"/>

```

```

<!-- messages -->

<!-- Even though it is possible to mix payments with applied credits, it is recommended that they
      be recorded in separate requests -->
<xs:element name="RecordPaymentsRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="billUserID" type="xs:string" minOccurs="1" maxOccurs="1" />
      <xs:element name="billKey" type="xs:int" minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="feePayment" type="FeePayment" minOccurs="1" maxOccurs="unbounded"/>
      <!-- selfPaid defaults to false if not provided -->
      <xs:element name="selfPaid" type="xs:boolean" minOccurs="0" maxOccurs="1" />
      <!-- If true, paid at library will be set to billing library -->
      <!-- Defaults to false if not specified -->
      <xs:element name="paidAtBillingLibrary" type="xs:boolean" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="RecordPaymentsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="feePaymentRecorded" type="FeePaymentRecorded" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element name="feePaymentProblem" type="FeePaymentProblem" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element name="netAmountRecorded" type="common:Money" minOccurs="1" maxOccurs="1" />
      <xs:element name="refundAmount" type="common:Money" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="CheckoutPatronInfoRequest">

</xs:element>

<xs:element name="CheckoutPatronInfoResponse">

</xs:element>

<xs:element name="CheckoutRequest">

</xs:element>

<xs:element name="CheckoutResponse">

</xs:element>

<xs:element name="LookupMailDeliveryEstimatesInfoRequest">
  <xs:complexType>
    <xs:sequence>
      <!-- if no user ID is sent, session user is used -->
      <xs:element name="userID" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        <xs:element name="deliveryServiceID" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="LookupMailDeliveryEstimatesInfoResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="maiDeliveryEstimateInfo" type="MailDeliveryEstimatesInfo" minOccurs="1"
                maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="HoldFeeRequest" />

<xs:element name="HoldFeeResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="holdFee" type="common:Money" minOccurs="1" maxOccurs="1" />
            <xs:element name="holdFeeTax" type="common:Money" minOccurs="1" maxOccurs="1" />
            <xs:element name="totalHoldFee" type="common:Money" minOccurs="1" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<!-- objects -->

<xs:complexType name="FeePayment">
    <xs:sequence>
        <xs:element name="amount" type="common:Money" minOccurs="1" maxOccurs="1" />
        <xs:element name="paymentTypeID" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="vendorTransactionNumber" type="xs:string" minOccurs="0" maxOccurs="1" />
        <!-- If payment type is CREDITACCT, the Credit Reason ID must be provided. -->
        <xs:element name="creditReasonID" type="xs:string" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="FeePaymentRecorded">
    <xs:sequence>
        <xs:element name="billKey" type="xs:int" minOccurs="1" maxOccurs="1" />
        <!-- paymentTypeID may be null if there was no payment recorded -->
        <xs:element name="paymentTypeID" type="xs:string" minOccurs="0" maxOccurs="1" />
        <xs:element name="paymentTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
        <xs:element name="amountPaid" type="common:Money" minOccurs="1" maxOccurs="1" />
        <xs:element name="amountRemaining" type="common:Money" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="FeePaymentProblem">
    <xs:sequence>
        <xs:element name="billKey" type="xs:int" minOccurs="1" maxOccurs="1" />
        <xs:element name="problem" type="xs:string" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
</xs:complexType>

```

```

    </xs:sequence>
</xs:complexType>

<xs:complexType name="MailDeliveryEstimatesInfo">
  <xs:sequence>
    <xs:element name="deliveryServiceID" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="deliveryServiceDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="estimatedShippingFee" type="common:Money" minOccurs="0" maxOccurs="1" />
    <xs:element name="estimatedShippingTime" type="xs:int" minOccurs="0" maxOccurs="1" />
    <!-- is false if no shipping cost mapping was found for delivery service associated to user's
         library and profile settings, or if delivery service ID is not found. -->
    <xs:element name="valid" type="xs:boolean" minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<!-- ***** Enumerations ***** -->
<!-- values correspond to the Symphony server "Hold Type" enumeration -->
<!-- Symphony Web Services only supports demand management holds which -->
<!-- accepts only COPY or TITLE. -->
<!-- CALL and SYSTEM are deprecated and only used for data load purposes -->
<xs:simpleType name="HoldType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="COPY"/>
    <xs:enumeration value="TITLE"/>
    <xs:enumeration value="CALL"/>
    <xs:enumeration value="SYSTEM" />
  </xs:restriction>
</xs:simpleType>

<!-- values correspond to the Symphony server "Hold Range" enumeration -->
<!-- Hold Range only applies when the Hold Type is "TITLE" -->
<xs:simpleType name="HoldRange">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LIBRARY"/>
    <xs:enumeration value="GROUP"/>
    <xs:enumeration value="SYSTEM"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CheckoutHistoryRule">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NOHISTORY" />
    <xs:enumeration value="CIRCRULE" />
    <xs:enumeration value="ALLCHARGES" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="FeeInfoFilter">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNPAID_FEES" />
    <xs:enumeration value="UNPAID_FEES_AND_PAYMENTS" />
    <xs:enumeration value="PAID_FEES" />
    <xs:enumeration value="PAID_FEES_AND_PAYMENTS" />
  </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="ALL_FEES" />
        <xs:enumeration value="ALL_FEES_AND_PAYMENTS" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CheckoutInfoFilter">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ALL" />
        <xs:enumeration value="ALLEXCEPTLOST" />
        <xs:enumeration value="ALLEXCEPTCLAIMEDRETURNED" />
        <xs:enumeration value="CLAIMEDRETURNED" />
        <xs:enumeration value="LOST" />
        <xs:enumeration value="OVERDUE" />
        <xs:enumeration value="RENEWED" />
        <xs:enumeration value="RECALLED" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="HoldInfoFilter">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ALL" />
        <xs:enumeration value="AVAILABLE" />
        <xs:enumeration value="COUNT_ONLY" />
        <xs:enumeration value="ACTIVE" />
        <xs:enumeration value="INACTIVE" />
        <xs:enumeration value="RECALLED" />
        <xs:enumeration value="RESERVE" />
        <xs:enumeration value="UNAVAILABLE" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="HoldInactiveType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="HOLD_INACTIVE_REASON_NONE" />
        <xs:enumeration value="HOLD_INACTIVE_REASON_FILLED" />
        <xs:enumeration value="HOLD_INACTIVE_REASON_CANCELLED" />
        <xs:enumeration value="HOLD_INACTIVE_REASON_EXPIRED" />
        <xs:enumeration value="HOLD_INACTIVE_REASON_CANCELLED_ON_RESERVE" />
        <xs:enumeration value="HOLD_INACTIVE_REASON_BLANKET_NONEED" />
        <xs:enumeration value="HOLD_INACTIVE_REASON_CANCELLED_ON_ORDER" />
        <xs:enumeration value="HOLD_INACTIVE_REASON_MODIFIED_ORDER" />
        <xs:enumeration value="HOLD_INACTIVE_REASON_EXP_ONSHLF" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="StatusType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="BARRED" />
        <xs:enumeration value="BLOCKED" />
        <xs:enumeration value="DELINQUENT" />
        <xs:enumeration value="OK" />
    </xs:restriction>
</xs:simpleType>

```

```
</xs:schema>
```

## Catalog service

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:common="http://schemas.sirsidynix.com/symws/common"
  targetNamespace="http://schemas.sirsidynix.com/symws/catalog"
  xmlns="http://schemas.sirsidynix.com/symws/catalog">

  <xs:import namespace="http://schemas.sirsidynix.com/symws/common" schemaLocation="common.xsd"/>

  <!-- messages -->

  <xs:element name="CreateBriefTitleRequest" type="BriefTitle"/>
  <xs:element name="CreateBriefTitleResponse" type="BriefTitle"/>

  <xs:complexType name="CoreTitle">
    <xs:complexContent>
      <xs:extension base="common:Entity">
        <xs:sequence>
          <xs:element name="titleKey" type="xs:long" minOccurs="0" maxOccurs="1"/>
          <xs:element name="titleControlNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="catalogFormatID" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="yearOfPublication" type="xs:int" minOccurs="0" maxOccurs="1"/>
          <xs:element name="shadow" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
          <xs:element name="reviewStatus" type="xs:string" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="BriefTitle">
    <xs:complexContent>
      <xs:extension base="CoreTitle">
        <xs:sequence>
          <xs:element name="author" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="publisherName" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="publicationPlace" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="edition" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="ISBN" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="ISSN" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="UPC" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="OCLCControlNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="url" type="xs:string" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```
</xs:schema>
```

## Common service

```
<xs:schema xmlns="http://schemas.sirsidynix.com/symws/common"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://schemas.sirsidynix.com/symws/common"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- Base transactional entity type -->
  <xs:complexType name="Entity">
    <xs:attribute name="operation" type="EntityOperation"/>
  </xs:complexType>

  <xs:simpleType name="EntityOperation">
    <xs:restriction base="xs:string">
      <xs:enumeration value="NOOP"/>
      <xs:enumeration value="CREATE"/>
      <xs:enumeration value="UPDATE"/>
      <xs:enumeration value="DELETE"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="Money">
    <xs:simpleContent>
      <xs:extension base="xs:decimal">
        <xs:attribute name="currency" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="NillableDate">
    <xs:choice>
      <xs:element name="value" type="xs:date"/>
      <xs:element name="nil" type="xs:boolean"/> <!-- should only be true if specified -->
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="NillableDateTime">
    <xs:choice>
      <xs:element name="value" type="xs:dateTime"/>
      <xs:element name="nil" type="xs:boolean"/> <!-- should only be true if specified -->
    </xs:choice>
  </xs:complexType>

  <!-- restful faults -->
  <xs:element name="Fault">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="code" type="xs:string"/>
        <xs:element name="string" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



```

</xs:complexType>
</xs:element>

<xs:complexType name="SearchTerm">
  <xs:sequence>
    <!-- The server supports a maximum term length of 1,024 characters. -->
    <xs:element name="searchValue" minOccurs="1" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="1024" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="searchIndexID" type="xs:string" minOccurs="0" maxOccurs="1" />
    <!-- The search operator will default to AND if not specified. -->
    <!-- The search operator is not valid in the final searchTerm and will be ignored. -->
    <xs:element name="operator" minOccurs="0" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="AND" />
          <xs:enumeration value="OR" />
          <xs:enumeration value="NOT" />
          <xs:enumeration value="XOR" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="SearchResult">
  <xs:sequence>
    <xs:element name="queryID" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="totalRecords" type="xs:long" minOccurs="1" maxOccurs="1" />
    <xs:element name="firstRecordNumber" type="xs:long" minOccurs="1" maxOccurs="1" />
    <xs:element name="lastRecordNumber" type="xs:long" minOccurs="1" maxOccurs="1" />
    <xs:element name="responseMessage" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="SearchPagingRequest">
  <xs:sequence>
    <xs:element name="queryID" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="firstRecordToRetrieve" type="xs:long" minOccurs="1" maxOccurs="1" />
    <!-- numberOfRecordsToRetrieve will default to 20 if not specified. -->
    <xs:element name="numberOfRecordsToRetrieve" type="xs:long" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AddressInfo">
  <xs:sequence>
    <xs:element name="addressPolicyID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="addressPolicyDescription" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="addressValue" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

```

    </xs:sequence>
  </xs:complexType>

</xs:schema>

```

## Selection service

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:common="http://schemas.sirsidynix.com/symws/common"
  targetNamespace="http://schemas.sirsidynix.com/symws/selection"
  xmlns="http://schemas.sirsidynix.com/symws/selection">

  <xs:import namespace="http://schemas.sirsidynix.com/symws/common" schemaLocation="common.xsd"/>

  <!-- messages -->
  <xs:element name="SaveSelectionListRequest" type="SelectionList"/>
  <xs:element name="SaveSelectionListResponse" type="SelectionList"/>

  <xs:element name="SaveSelectionLineRequest" type="SelectionLine"/>
  <xs:element name="SaveSelectionLineResponse" type="SelectionLine"/>

  <!-- objects -->
  <xs:complexType name="SelectionList">
    <xs:complexContent>
      <xs:extension base="common:Entity">
        <xs:sequence>
          <xs:element name="selectionListID" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="library" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="fiscalCycle" type="xs:int" minOccurs="0" maxOccurs="1"/>
          <xs:element name="selectionListName" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="category1" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="category2" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="owner" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="status" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="cutoffDate" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="extendedInfo" type="ExtendedInfo" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="selectionListLine" type="SelectionListLine" minOccurs="0"
            maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SelectionListLine">
    <xs:complexContent>
      <xs:extension base="common:Entity">
        <xs:sequence>
          <xs:element name="selectionListID" type="xs:long" minOccurs="0" maxOccurs="1"/>
          <xs:element name="lineNumber" type="xs:int" minOccurs="0" maxOccurs="1"/>

```

```

<xs:element name="fiscalCycle" type="xs:int" minOccurs="0" maxOccurs="1"/>
<xs:element name="vendorID" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="userID" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="callNumber" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="itemID" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="volume" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="binding" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="quantity" type="xs:int" minOccurs="0" maxOccurs="1"/>
<xs:element name="vendorUnitListPrice" type="common:Money" minOccurs="0" maxOccurs="1"/>
<xs:element name="vendorCurrency" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="conversionFactor" type="xs:string" minOccurs="0" maxOccurs="1"/>
<!-- conversionFactor defaults to TABLE when vendorCurrency is specified -->
<xs:element name="discount" type="common:Money" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="ExtendedInfo">
  <xs:sequence>
    <xs:element name="EntryType" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="EntryData" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

</xs:schema>

```



# Appendix 8: Service WSDL Definitions

For convenient reference, this section lists the WSDL definitions for each service in Web Services.

- [Standard service](#)
- [Security service](#)
- [Patron service](#)
- [Admin service](#)
- [Request service](#) on page 343
- [Circulation service](#) on page 346
- [Catalog service](#) on page 347

## Standard service

```
<?xml version="1.0"?>
<definitions name='StandardService'
  targetNamespace='http://www.sirsidynix.com/symws/standard'
  xmlns='http://schemas.xmlsoap.org/wsdl/'
  xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:header='http://www.sirsidynix.com/xmlns/common/header'
  xmlns:stnd='http://schemas.sirsidynix.com/symws/standard'
  xmlns:tns="http://www.sirsidynix.com/symws/standard"
  xmlns:xs='http://www.w3.org/2001/XMLSchema'>

  <types>
    <xs:schema>
      <xs:import schemaLocation="standard.xsd" namespace="http://schemas.sirsidynix.com/symws/standard"/>
      <xs:import schemaLocation="header.xsd" namespace="http://www.sirsidynix.com/xmlns/common/header"/>
    </xs:schema>
  </types>

  <message name='LookupTitleInfoRequest'>
    <part name='body' element='stnd:LookupTitleInfoRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='LookupTitleInfoResponse'>
    <part name='return' element='stnd:LookupTitleInfoResponse' />
  </message>

  <message name='SearchCatalogRequest'>
    <part name='body' element='stnd:SearchCatalogRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
```

```

</message>
<message name='SearchCatalogResponse'>
  <part name='return' element='std:SearchCatalogResponse' />
</message>

<message name='SearchCatalogPagingRequest'>
  <part name='body' element='std:SearchCatalogPagingRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='SearchCatalogPagingResponse'>
  <part name='return' element='std:SearchCatalogPagingResponse' />
</message>

<message name='LookupMostPopularRequest'>
  <part name='body' element='std:LookupMostPopularRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupMostPopularResponse'>
  <part name='return' element='std:LookupMostPopularResponse' />
</message>

<message name='SearchInfoDeskRequest'>
  <part name='body' element='std:SearchInfoDeskRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='SearchInfoDeskResponse'>
  <part name='return' element='std:SearchInfoDeskResponse' />
</message>

<message name='SearchInfoDeskPagingRequest'>
  <part name='body' element='std:SearchInfoDeskPagingRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='SearchInfoDeskPagingResponse'>
  <part name='return' element='std:SearchInfoDeskPagingResponse' />
</message>

<message name='VersionRequest'>
  <part name='body' element='std:VersionRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='VersionResponse'>
  <part name='return' element='std:VersionResponse' />
</message>

<message name='LicenseRequest'>
  <part name='body' element='std:LicenseRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LicenseResponse'>
  <part name='return' element='std:LicenseResponse' />
</message>

```

```

<message name='LookupBestSellersDisplayListRequest'>
  <part name='body' element='stnd:LookupBestSellersDisplayListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupBestSellersDisplayListResponse'>
  <part name='return' element='stnd:LookupBestSellersDisplayListResponse' />
</message>

<message name='LookupRecommendedReadingDisplayListRequest'>
  <part name='body' element='stnd:LookupRecommendedReadingDisplayListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupRecommendedReadingDisplayListResponse'>
  <part name='return' element='stnd:LookupRecommendedReadingDisplayListResponse' />
</message>

<message name='LookupBestSellerTitlesRequest'>
  <part name='body' element='stnd:LookupBestSellerTitlesRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupBestSellerTitlesResponse'>
  <part name='return' element='stnd:LookupBestSellerTitlesResponse' />
</message>

<message name='LookupRecommendedReadingTitlesRequest'>
  <part name='body' element='stnd:LookupRecommendedReadingTitlesRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupRecommendedReadingTitlesResponse'>
  <part name='return' element='stnd:LookupRecommendedReadingTitlesResponse' />
</message>

<message name='LookupHotSitesRequest'>
  <part name='body' element='stnd:LookupHotSitesRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupHotSitesResponse'>
  <part name='return' element='stnd:LookupHotSitesResponse' />
</message>

<portType name='StandardEndpoint'>
  <operation name='lookupTitleInfo'>
    <input message='tns:LookupTitleInfoRequest' />
    <output message='tns:LookupTitleInfoResponse' />
  </operation>
  <operation name='searchCatalog'>
    <input message='tns:SearchCatalogRequest' />
    <output message='tns:SearchCatalogResponse' />
  </operation>
  <operation name='searchCatalogPaging'>
    <input message='tns:SearchCatalogPagingRequest' />
    <output message='tns:SearchCatalogPagingResponse' />
  </operation>

```

```

<operation name='lookupMostPopular'>
  <input message='tns:LookupMostPopularRequest' />
  <output message='tns:LookupMostPopularResponse' />
</operation>
<operation name='searchInfoDesk'>
  <input message='tns:SearchInfoDeskRequest' />
  <output message='tns:SearchInfoDeskResponse' />
</operation>
<operation name='searchInfoDeskPaging'>
  <input message='tns:SearchInfoDeskPagingRequest' />
  <output message='tns:SearchInfoDeskPagingResponse' />
</operation>
<operation name='version'>
  <input message='tns:VersionRequest' />
  <output message='tns:VersionResponse' />
</operation>
<operation name='license'>
  <input message='tns:LicenseRequest' />
  <output message='tns:LicenseResponse' />
</operation>
<operation name='lookupBestSellersDisplayList'>
  <input message='tns:LookupBestSellersDisplayListRequest' />
  <output message='tns:LookupBestSellersDisplayListResponse' />
</operation>
<operation name='LookupRecommendedReadingDisplayList'>
  <input message='tns:LookupRecommendedReadingDisplayListRequest' />
  <output message='tns:LookupRecommendedReadingDisplayListResponse' />
</operation>
<operation name='lookupBestSellerTitles'>
  <input message='tns:LookupBestSellerTitlesRequest' />
  <output message='tns:LookupBestSellerTitlesResponse' />
</operation>
<operation name='lookupRecommendedReadingTitles'>
  <input message='tns:LookupRecommendedReadingTitlesRequest' />
  <output message='tns:LookupRecommendedReadingTitlesResponse' />
</operation>
<operation name='lookupHotSites'>
  <input message='tns:LookupHotSitesRequest' />
  <output message='tns:LookupHotSitesResponse' />
</operation>
</portType>

<binding name='StandardServiceBinding' type='tns:StandardEndpoint'>
<soap:binding style='document' transport='http://schemas.xmlsoap.org/soap/http' />
  <operation name='lookupTitleInfo'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupTitleInfoRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>

```



```

</operation>
<operation name='searchCatalog'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:SearchCatalogRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='searchCatalogPaging'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:SearchCatalogPagingRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='lookupMostPopular'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupMostPopularRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='searchInfoDesk'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:SearchInfoDeskRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='searchInfoDeskPaging'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:SearchInfoDeskPagingRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='version'>
  <soap:operation soapAction=''/>

```

```

    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:VersionRequest" part="header" use="literal" />
    </input>
  </output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='license'>
  <soap:operation soapAction='' />
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LicenseRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='lookupBestSellersDisplayList'>
  <soap:operation soapAction='' />
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupBestSellersDisplayListRequest" part="header" use="literal"
/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='LookupRecommendedReadingDisplayList'>
  <soap:operation soapAction='' />
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupRecommendedReadingDisplayListRequest" part="header"
      use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='lookupBestSellerTitles'>
  <soap:operation soapAction='' />
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupBestSellerTitlesRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='lookupRecommendedReadingTitles'>
  <soap:operation soapAction='' />
  <input>

```

```

        <soap:body parts="body" use="literal"/>
        <soap:header message="tns:LookupRecommendedReadingTitlesRequest" part="header" use="literal"
/>
    </input>
    <output>
        <soap:body use='literal'/>
    </output>
</operation>
<operation name='lookupHotSites'>
    <soap:operation soapAction=''/>
    <input>
        <soap:body parts="body" use="literal"/>
        <soap:header message="tns:LookupHotSitesRequest" part="header" use="literal" />
    </input>
    <output>
        <soap:body use='literal'/>
    </output>
</operation>
</binding>

<!-- custom note here to show it uses our wsdl -->
<service name='StandardService'>
    <port binding='tns:StandardServiceBinding' name='StandardPort'>
        <soap:address location='http://localhost:8080/symws/soap/standard'/>
    </port>
</service>
</definitions>

```

## Security service

```

<?xml version="1.0"?>
<definitions name='SecurityService'
    targetNamespace='http://www.sirsidynix.com/symws/security'
    xmlns='http://schemas.xmlsoap.org/wsdl/'
    xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
    xmlns:header='http://www.sirsidynix.com/xmlns/common/header'
    xmlns:sec='http://schemas.sirsidynix.com/symws/security'
    xmlns:tns="http://www.sirsidynix.com/symws/security"
    xmlns:xs='http://www.w3.org/2001/XMLSchema'>

    <types>
        <xs:schema>
            <xs:import schemaLocation="security.xsd" namespace="http://schemas.sirsidynix.com/symws/
security"/>
            <xs:import schemaLocation="header.xsd" namespace="http://www.sirsidynix.com/xmlns/
common/header"/>
        </xs:schema>
    </types>

    <message name='AuthenticateUserRequest'>
        <part name='body' element='sec:AuthenticateUserRequest' />
        <part name='header' element='header:SdHeader' />
    </message>

```

```

<message name='AuthenticateUserResponse'>
  <part name='return' element='sec:AuthenticateUserResponse' />
</message>

<message name='LoginUserRequest'>
  <part name='body' element='sec:LoginUserRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LoginUserResponse'>
  <part name='return' element='sec:LoginUserResponse' />
</message>

<message name='LogoutUserRequest'>
  <part name='body' element='sec:LogoutUserRequest' />
  <part name='header' element='header:SdHeader' />
</message>

<message name='LookupUserInfoRequest'>
  <part name='body' element='sec:LookupUserInfoRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupUserInfoResponse'>
  <part name='return' element='sec:LookupUserInfoResponse' />
</message>

<message name='LookupSessionInfoRequest'>
  <part name='body' element='sec:LookupSessionInfoRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupSessionInfoResponse'>
  <part name='return' element='sec:LookupSessionInfoResponse' />
</message>

<message name='GetPassthroughTokenRequest'>
  <part name='body' element='sec:GetPassthroughTokenRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetPassthroughTokenResponse'>
  <part name='return' element='sec:GetPassthroughTokenResponse' />
</message>

<portType name='SecurityEndpoint'>
  <operation name='authenticateUser'>
    <input message='tns:AuthenticateUserRequest' />
    <output message='tns:AuthenticateUserResponse' />
  </operation>
  <operation name='loginUser'>
    <input message='tns:LoginUserRequest' />
    <output message='tns:LoginUserResponse' />
  </operation>
  <operation name='logoutUser'>
    <input message='tns:LogoutUserRequest' />
  </operation>

```

```

<operation name='lookupUserInfo'>
  <input message='tns:LookupUserInfoRequest' />
  <output message='tns:LookupUserInfoResponse' />
</operation>
<operation name='lookupSessionInfo'>
  <input message='tns:LookupSessionInfoRequest' />
  <output message='tns:LookupSessionInfoResponse' />
</operation>
<operation name='getPassthroughToken'>
  <input message='tns:GetPassthroughTokenRequest' />
  <output message='tns:GetPassthroughTokenResponse' />
</operation>
</portType>

<binding name='SecurityServiceBinding' type='tns:SecurityEndpoint'>
  <soap:binding style='document' transport='http://schemas.xmlsoap.org/soap/http' />
  <operation name='authenticateUser'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:AuthenticateUserRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='loginUser'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LoginUserRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='logoutUser'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LogoutUserRequest" part="header" use="literal" />
    </input>
  </operation>
  <operation name='lookupUserInfo'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupUserInfoRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>

```

```

<operation name='lookupSessionInfo'>
  <soap:operation soapAction='' />
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupSessionInfoRequest" part="header" use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='getPassthroughToken'>
  <soap:operation soapAction='' />
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:GetPassthroughTokenRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
</binding>

<!-- custom note here to show it uses our wsdl -->
<service name='SecurityService'>
  <port binding='tns:SecurityServiceBinding' name='SecurityPort'>
    <soap:address location='http://localhost:8080/symws/soap/security'/>
  </port>
</service>
</definitions>

```

## Patron service

```

<?xml version="1.0"?>
<definitions name='PatronService'
  targetNamespace='http://www.sirsidynix.com/symws/patron'
  xmlns='http://schemas.xmlsoap.org/wsdl/'
  xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:header='http://www.sirsidynix.com/xmlns/common/header'
  xmlns:pat='http://schemas.sirsidynix.com/symws/patron'
  xmlns:tns='http://www.sirsidynix.com/symws/patron'
  xmlns:xs='http://www.w3.org/2001/XMLSchema'>

  <types>
    <xs:schema>
      <xs:import schemaLocation="patron.xsd" namespace="http://schemas.sirsidynix.com/symws/patron"/>
      <xs:import schemaLocation="header.xsd" namespace="http://www.sirsidynix.com/xmlns/common/header"/>
    </xs:schema>
  </types>

  <message name='LookupPatronInfoRequest'>
    <part name='body' element='pat:LookupPatronInfoRequest' />
  </message>

```

```

    <part name='header' element='header:SdHeader' />
  </message>
  <message name='LookupPatronInfoResponse'>
    <part name='return' element='pat:LookupPatronInfoResponse' />
  </message>

  <message name='LookupPatronInfoRequest'> <part name='body' element='pat:LookupPatronInfoRequest' />
<part name='header' element='header:SdHeader' /> </message> <message name='LookupPatronInfoResponse'>
<part name='return' element='pat:LookupPatronInfoResponse' /> </message>

  <message name='LookupMyAccountInfoRequest'>
    <part name='body' element='pat:LookupMyAccountInfoRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='LookupMyAccountInfoResponse'>
    <part name='return' element='pat:LookupMyAccountInfoResponse' />
  </message>

  <message name='RenewMyCheckoutRequest'>
    <part name='body' element='pat:RenewMyCheckoutRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='RenewMyCheckoutResponse'>
    <part name='return' element='pat:RenewMyCheckoutResponse' />
  </message>

  <message name='CreateMyHoldRequest'>
    <part name='body' element='pat:CreateMyHoldRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='CreateMyHoldResponse'>
    <part name='return' element='pat:CreateMyHoldResponse' />
  </message>

  <message name='CancelMyHoldRequest'>
    <part name='body' element='pat:CancelMyHoldRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='CancelMyHoldResponse'>
    <part name='return' element='pat:CancelMyHoldResponse' />
  </message>

  <message name='CreateSelfRegisteredPatronRequest'>
    <part name='body' element='pat:CreateSelfRegisteredPatronRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='CreateSelfRegisteredPatronResponse'>
    <part name='return' element='pat:CreateSelfRegisteredPatronResponse' />
  </message>

  <message name='LookupMyListsRequest'>
    <part name='body' element='pat:LookupMyListsRequest' />

```

```

    <part name='header' element='header:SdHeader' />
</message>
<message name='LookupMyListsResponse'>
    <part name='return' element='pat:LookupMyListsResponse' />
</message>

<message name='LookupMyListRequest'>
    <part name='body' element='pat:LookupMyListRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='LookupMyListResponse'>
    <part name='return' element='pat:LookupMyListResponse' />
</message>

<message name='GetMyListRequest'>
    <part name='body' element='pat:GetMyListRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='GetMyListResponse'>
    <part name='return' element='pat:GetMyListResponse' />
</message>

<message name='SaveMyListRequest'>
    <part name='body' element='pat:SaveMyListRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='SaveMyListResponse'>
    <part name='return' element='pat:SaveMyListResponse' />
</message>

<message name='DeleteMyListRequest'>
    <part name='body' element='pat:DeleteMyListRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='DeleteMyListResponse'>
    <part name='return' element='pat:DeleteMyListResponse' />
</message>

<message name='ChangeMyPINRequest'>
    <part name='body' element='pat:ChangeMyPINRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='ChangeMyPINResponse'>
    <part name='return' element='pat:ChangeMyPINResponse' />
</message>

<message name='ModifyMyHoldRequest'>
    <part name='body' element='pat:ModifyMyHoldRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='ModifyMyHoldResponse'>
    <part name='return' element='pat:ModifyMyHoldResponse' />
</message>

```



```

<message name='SuspendMyHoldRequest'>
  <part name='body' element='pat:SuspendMyHoldRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='SuspendMyHoldResponse'>
  <part name='return' element='pat:SuspendMyHoldResponse' />
</message>

<message name='UnsuspendMyHoldRequest'>
  <part name='body' element='pat:UnsuspendMyHoldRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='UnsuspendMyHoldResponse'>
  <part name='return' element='pat:UnsuspendMyHoldResponse' />
</message>

<message name='ChangeMyAlternativeIDRequest'>
  <part name='body' element='pat:ChangeMyAlternativeIDRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='ChangeMyAlternativeIDResponse'>
  <part name='return' element='pat:ChangeMyAlternativeIDResponse' />
</message>

<message name='EnableMyCheckoutHistoryRequest'>
  <part name='body' element='pat:enableMyCheckoutHistoryRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='EnableMyCheckoutHistoryResponse'>
  <part name='return' element='pat:enableMyCheckoutHistoryResponse' />
</message>

<message name='ActivateOnlineRegisteredUserRequest'>
  <part name='body' element='pat:ActivateOnlineRegisteredUserRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='ActivateOnlineRegisteredUserResponse'>
  <part name='return' element='pat:ActivateOnlineRegisteredUserResponse' />
</message>

<message name='PlaceMyHoldRequest'>
  <part name='body' element='pat:PlaceMyHoldRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='PlaceMyHoldResponse'>
  <part name='return' element='pat:PlaceMyHoldResponse' />
</message>

<portType name='PatronEndpoint'>
  <operation name='lookupPatronInfo'>
    <input message='tns:LookupPatronInfoRequest' />
    <output message='tns:LookupPatronInfoResponse' />
  </operation>

```

```

</operation>
<operation name='lookupMyAccountInfo'>
  <input message='tns:LookupMyAccountInfoRequest' />
  <output message='tns:LookupMyAccountInfoResponse' />
</operation>

<operation name='renewMyCheckout'>
  <input message='tns:RenewMyCheckoutRequest' />
  <output message='tns:RenewMyCheckoutResponse' />
</operation>

<operation name='createMyHold'>
  <input message='tns:CreateMyHoldRequest' />
  <output message='tns:CreateMyHoldResponse' />
</operation>
<operation name='cancelMyHold'>
  <input message='tns:CancelMyHoldRequest' />
  <output message='tns:CancelMyHoldResponse' />
</operation>
<operation name='createSelfRegisteredPatron'>
  <input message='tns:CreateSelfRegisteredPatronRequest' />
  <output message='tns:CreateSelfRegisteredPatronResponse' />
</operation>
<operation name='lookupMyLists'>
  <input message='tns:LookupMyListsRequest' />
  <output message='tns:LookupMyListsResponse' />
</operation>
<operation name='lookupMyList'>
  <input message='tns:LookupMyListRequest' />
  <output message='tns:LookupMyListResponse' />
</operation>
<operation name='getMyList'>
  <input message='tns:GetMyListRequest' />
  <output message='tns:GetMyListResponse' />
</operation>
<operation name='saveMyList'>
  <input message='tns:SaveMyListRequest' />
  <output message='tns:SaveMyListResponse' />
</operation>
<operation name='deleteMyList'>
  <input message='tns:DeleteMyListRequest' />
  <output message='tns:DeleteMyListResponse' />
</operation>
<operation name='changeMyPIN'>
  <input message='tns:ChangeMyPINRequest' />
  <output message='tns:ChangeMyPINResponse' />
</operation>
<operation name='modifyMyHold'>
  <input message='tns:ModifyMyHoldRequest' />
  <output message='tns:ModifyMyHoldResponse' />
</operation>
<operation name='suspendMyHold'>
  <input message='tns:SuspendMyHoldRequest' />

```

```

        <output message='tns:SuspendMyHoldResponse' />
    </operation>
    <operation name='unsuspendMyHold'>
        <input message='tns:UnsuspendMyHoldRequest' />
        <output message='tns:UnsuspendMyHoldResponse' />
    </operation>
    <operation name='changeMyAlternativeID'>
        <input message='tns:ChangeMyAlternativeIDRequest' />
        <output message='tns:ChangeMyAlternativeIDResponse' />
    </operation>
    <operation name='enableMyCheckoutHistory'>
        <input message='tns:enableMyCheckoutHistoryRequest' />
        <output message='tns:enableMyCheckoutHistoryResponse' />
    </operation>
    <operation name='activateOnlineRegisteredUser'>
        <input message='tns:ActivateOnlineRegisteredUserRequest' />
        <output message='tns:ActivateOnlineRegisteredUserResponse' />
    </operation>
    <operation name='placeMyHold'>
        <input message='tns:PlaceMyHoldRequest' />
        <output message='tns:PlaceMyHoldResponse' />
    </operation>
</portType>

<binding name='PatronServiceBinding' type='tns:PatronEndpoint'>
    <soap:binding style='document' transport='http://schemas.xmlsoap.org/soap/http' />
    <operation name='lookupPatronInfo'>
        <soap:operation soapAction='' />
        <input>
            <soap:body parts="body" use="literal" />
            <soap:header message="tns:LookupPatronInfoRequest" part="header" use="literal" />
        </input>
        <output>
            <soap:body use='literal' />
        </output>
    </operation>
    <operation name='lookupMyAccountInfo'>
        <soap:operation soapAction='' />
        <input>
            <soap:body parts="body" use="literal" />
            <soap:header message="tns:LookupMyAccountInfoRequest" part="header" use="literal" />
        </input>
        <output>
            <soap:body use='literal' />
        </output>
    </operation>

    <operation name='renewMyCheckout'>
        <soap:operation soapAction='' />
        <input>
            <soap:body parts="body" use="literal" />
            <soap:header message="tns:RenewMyCheckoutRequest" part="header" use="literal" />
        </input>

```

```

    <output>
      <soap:body use='literal' />
    </output>
  </operation>

  <operation name='createMyHold'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:CreateMyHoldRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>

  <operation name='cancelMyHold'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:CancelMyHoldRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>

  <operation name='createSelfRegisteredPatron'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:CreateSelfRegisteredPatronRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>

  <operation name='lookupMyLists'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupMyListsRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>

  <operation name='lookupMyList'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupMyListRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>

```

```

    </output>
</operation>
<operation name='getMyList'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:GetMyListRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='saveMyList'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:SaveMyListRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='deleteMyList'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:DeleteMyListRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='changeMyPIN'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:ChangeMyPINRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='modifyMyHold'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:ModifyMyHoldRequest" part="header" use="literal" />
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='suspendMyHold'>

```

```

    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:SuspendMyHoldRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='unsuspendMyHold'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:UnsuspendMyHoldRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='changeMyAlternativeID'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:ChangeMyAlternativeIDRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='enableMyCheckoutHistory'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:enableMyCheckoutHistoryRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='activateOnlineRegisteredUser'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:ActivateOnlineRegisteredUserRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='placeMyHold'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />

```

```

        <soap:header message="tns:PlaceMyHoldRequest" part="header" use="literal" />
    </input>
    <output>
        <soap:body use='literal' />
    </output>
</operation>
</binding>

<!-- custom note here to show it uses our wsdl -->
<service name='PatronService'>
    <port binding='tns:PatronServiceBinding' name='PatronPort'>
        <soap:address location='http://localhost:8080/symws/soap/patron' />
    </port>
</service>
</definitions>

```

## Admin service

```

<?xml version="1.0"?>
<definitions name='AdminService'
    targetNamespace='http://www.sirsidynix.com/symws/admin'
    xmlns='http://schemas.xmlsoap.org/wsdl/'
    xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
    xmlns:header='http://www.sirsidynix.com/xmlns/common/header'
    xmlns:adm='http://schemas.sirsidynix.com/symws/admin'
    xmlns:tns="http://www.sirsidynix.com/symws/admin"
    xmlns:xs='http://www.w3.org/2001/XMLSchema'>

    <types>
        <xs:schema>
            <xs:import schemaLocation="admin.xsd" namespace="http://schemas.sirsidynix.com/symws/admin"/>
            <xs:import schemaLocation="header.xsd" namespace="http://www.sirsidynix.com/xmlns/common/header"/>
        </xs:schema>
    </types>

    <!-- ILS configuration operations -->
    <message name='LookupILSConfigurationRequest'>
        <part name='body' element='adm:LookupILSConfigurationRequest' />
        <part name='header' element='header:SdHeader' />
    </message>
    <message name='LookupILSConfigurationResponse'>
        <part name='return' element='adm:LookupILSConfigurationResponse' />
    </message>

    <message name='LookupPolicyListRequest'>
        <part name='body' element='adm:LookupPolicyListRequest' />
        <part name='header' element='header:SdHeader' />
    </message>
    <message name='LookupPolicyListResponse'>
        <part name='return' element='adm:LookupPolicyListResponse' />
    </message>

```

```

<message name='GetPolicyRequest'>
  <part name='body' element='adm:GetPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetPolicyResponse'>
  <part name='return' element='adm:GetPolicyResponse' />
</message>

<message name='LookupAccessLevelPolicyListRequest'>
  <part name='body' element='adm:LookupAccessLevelPolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupAccessLevelPolicyListResponse'>
  <part name='return' element='adm:LookupAccessLevelPolicyListResponse' />
</message>

<message name='GetAccessLevelPolicyRequest'>
  <part name='body' element='adm:GetAccessLevelPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetAccessLevelPolicyResponse'>
  <part name='return' element='adm:GetAccessLevelPolicyResponse' />
</message>

<message name='LookupAccessRestrictionPolicyListRequest'>
  <part name='body' element='adm:LookupAccessRestrictionPolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>

<message name='LookupAccessRestrictionPolicyListResponse'>
  <part name='return' element='adm:LookupAccessRestrictionPolicyListResponse' />
</message>

<message name='GetAccessRestrictionPolicyRequest'>
  <part name='body' element='adm:GetAccessRestrictionPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetAccessRestrictionPolicyResponse'>
  <part name='return' element='adm:GetAccessRestrictionPolicyResponse' />
</message>

<message name='LookupCitizenshipPolicyListRequest'>
  <part name='body' element='adm:LookupCitizenshipPolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupCitizenshipPolicyListResponse'>
  <part name='return' element='adm:LookupCitizenshipPolicyListResponse' />
</message>

<message name='GetCitizenshipPolicyRequest'>
  <part name='body' element='adm:GetCitizenshipPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>

```



```

<message name='GetCitizenshipPolicyResponse'>
  <part name='return' element='adm:GetCitizenshipPolicyResponse' />
</message>

<message name='LookupClearancePolicyListRequest'>
  <part name='body' element='adm:LookupClearancePolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupClearancePolicyListResponse'>
  <part name='return' element='adm:LookupClearancePolicyListResponse' />
</message>
<message name='GetClearancePolicyRequest'>
  <part name='body' element='adm:GetClearancePolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetClearancePolicyResponse'>
  <part name='return' element='adm:GetClearancePolicyResponse' />
</message>

<message name='LookupItemTypePolicyListRequest'>
  <part name='body' element='adm:LookupItemTypePolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupItemTypePolicyListResponse'>
  <part name='return' element='adm:LookupItemTypePolicyListResponse' />
</message>

<message name='GetItemTypePolicyRequest'>
  <part name='body' element='adm:GetItemTypePolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetItemTypePolicyResponse'>
  <part name='return' element='adm:GetItemTypePolicyResponse' />
</message>

<message name='LookupLanguagePolicyListRequest'>
  <part name='body' element='adm:LookupLanguagePolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupLanguagePolicyListResponse'>
  <part name='return' element='adm:LookupLanguagePolicyListResponse' />
</message>

<message name='GetLanguagePolicyRequest'>
  <part name='body' element='adm:GetLanguagePolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetLanguagePolicyResponse'>
  <part name='return' element='adm:GetLanguagePolicyResponse' />
</message>

<message name='LookupLibraryPolicyListRequest'>
  <part name='body' element='adm:LookupLibraryPolicyListRequest' />

```

```

    <part name='header' element='header:SdHeader' />
</message>
<message name='LookupLibraryPolicyListResponse'>
    <part name='return' element='adm:LookupLibraryPolicyListResponse' />
</message>

<message name='GetLibraryPolicyRequest'>
    <part name='body' element='adm:GetLibraryPolicyRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='GetLibraryPolicyResponse'>
    <part name='return' element='adm:GetLibraryPolicyResponse' />
</message>

<message name='LookupLocationPolicyListRequest'>
    <part name='body' element='adm:LookupLocationPolicyListRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='LookupLocationPolicyListResponse'>
    <part name='return' element='adm:LookupLocationPolicyListResponse' />
</message>

<message name='GetLocationPolicyRequest'>
    <part name='body' element='adm:GetLocationPolicyRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='GetLocationPolicyResponse'>
    <part name='return' element='adm:GetLocationPolicyResponse' />
</message>

<message name='LookupMailServicePolicyListRequest'>
    <part name='body' element='adm:LookupMailServicePolicyListRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='LookupMailServicePolicyListResponse'>
    <part name='return' element='adm:LookupMailServicePolicyListResponse' />
</message>

<message name='GetMailServicePolicyRequest'>
    <part name='body' element='adm:GetMailServicePolicyRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='GetMailServicePolicyResponse'>
    <part name='return' element='adm:GetMailServicePolicyResponse' />
</message>

<message name='IsPatronSelfRegistrationEnabledRequest'>
    <part name='body' element='adm:IsPatronSelfRegistrationEnabledRequest' />
    <part name='header' element='header:SdHeader' />
</message>
<message name='IsPatronSelfRegistrationEnabledResponse'>
    <part name='return' element='adm:IsPatronSelfRegistrationEnabledResponse' />
</message>

```

```

<message name='LookupNeedToKnowPolicyListRequest'>
  <part name='body' element='adm:LookupNeedToKnowPolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupNeedToKnowPolicyListResponse'>
  <part name='return' element='adm:LookupNeedToKnowPolicyListResponse' />
</message>

<message name='GetNeedToKnowPolicyRequest'>
  <part name='body' element='adm:GetNeedToKnowPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetNeedToKnowPolicyResponse'>
  <part name='return' element='adm:GetNeedToKnowPolicyResponse' />
</message>

<message name='LookupUserLookupPolicyListRequest'>
  <part name='body' element='adm:LookupUserLookupPolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupUserLookupPolicyListResponse'>
  <part name='return' element='adm:LookupUserLookupPolicyListResponse' />
</message>

<message name='GetUserLookupPolicyRequest'>
  <part name='body' element='adm:GetUserLookupPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetUserLookupPolicyResponse'>
  <part name='return' element='adm:GetUserLookupPolicyResponse' />
</message>

<message name='LookupUserSortPolicyListRequest'>
  <part name='body' element='adm:LookupUserSortPolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupUserSortPolicyListResponse'>
  <part name='return' element='adm:LookupUserSortPolicyListResponse' />
</message>

<message name='GetUserSortPolicyRequest'>
  <part name='body' element='adm:GetUserSortPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetUserSortPolicyResponse'>
  <part name='return' element='adm:GetUserSortPolicyResponse' />
</message>

<message name='LookupBillReasonPolicyListRequest'>
  <part name='body' element='adm:LookupBillReasonPolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>

```

```

<message name='LookupBillReasonPolicyListResponse'>
  <part name='return' element='adm:LookupBillReasonPolicyListResponse' />
</message>

<message name='GetBillReasonPolicyRequest'>
  <part name='body' element='adm:GetBillReasonPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetBillReasonPolicyResponse'>
  <part name='return' element='adm:GetBillReasonPolicyResponse' />
</message>

<message name='LookupPaymentTypePolicyListRequest'>
  <part name='body' element='adm:LookupPaymentTypePolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupPaymentTypePolicyListResponse'>
  <part name='return' element='adm:LookupPaymentTypePolicyListResponse' />
</message>

<message name='GetPaymentTypePolicyRequest'>
  <part name='body' element='adm:See "GetBillPaymentTypePolicyRequest"' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetBillPaymentTypePolicyResponse'>
  <part name='return' element='adm:See "GetBillPaymentTypePolicyResponse"' />
</message>

<message name='GetCreditReasonPolicyRequest'>
  <part name='body' element='adm:GetCreditReasonPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetCreditReasonPolicyResponse'>
  <part name='return' element='adm:GetCreditReasonPolicyResponse' />
</message>

<message name='LookupNativeCurrencyRequest'>
  <part name='body' element='adm:LookupNativeCurrencyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupNativeCurrencyResponse'>
  <part name='return' element='adm:LookupNativeCurrencyResponse' />
</message>

<message name='LookupRequestTypePolicyListRequest'>
  <part name='body' element='adm:LookupRequestTypePolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupRequestTypePolicyListResponse'>
  <part name='return' element='adm:LookupRequestTypePolicyListResponse' />
</message>

<message name='GetRequestTypePolicyRequest'>

```

```

    <part name='body' element='adm:GetRequestTypePolicyRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='GetRequestTypePolicyResponse'>
    <part name='return' element='adm:GetRequestTypePolicyResponse' />
  </message>

  <message name='GetRequestListPolicyRequest'>
    <part name='body' element='adm:GetRequestListPolicyRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='GetRequestListPolicyResponse'>
    <part name='return' element='adm:GetRequestListPolicyResponse' />
  </message>

  <message name='LookupRequestListPolicyListRequest'>
    <part name='body' element='adm:LookupRequestListPolicyListRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='LookupRequestListPolicyListResponse'>
    <part name='return' element='adm:LookupRequestListPolicyListResponse' />
  </message>

  <message name='GetRequestStatusPolicyRequest'>
    <part name='body' element='adm:GetRequestStatusPolicyRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='GetRequestStatusPolicyResponse'>
    <part name='return' element='adm:GetRequestStatusPolicyResponse' />
  </message>

  <message name='LookupRequestStatusPolicyListRequest'>
    <part name='body' element='adm:LookupRequestStatusPolicyListRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='LookupRequestStatusPolicyListResponse'>
    <part name='return' element='adm:LookupRequestStatusPolicyListResponse' />
  </message>

  <message name='IsRestrictedAccessRequest'>
    <part name='body' element='adm:IsRestrictedAccessRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='IsRestrictedAccessResponse'>
    <part name='return' element='adm:IsRestrictedAccessResponse' />
  </message>

  <message name='GetLibraryGroupPolicyRequest'>
    <part name='body' element='adm:GetLibraryGroupPolicyRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='GetLibraryGroupPolicyResponse'>
    <part name='return' element='adm:GetLibraryGroupPolicyResponse' />
  </message>

```

```

</message>

<message name='LookupLibraryGroupListRequest'>
  <part name='body' element='adm:LookupLibraryGroupListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupLibraryGroupListResponse'>
  <part name='return' element='adm:LookupLibraryGroupListResponse' />
</message>

<message name='GetSearchLibraryGroupPolicyRequest'>
  <part name='body' element='adm:GetSearchLibraryGroupPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetSearchLibraryGroupPolicyResponse'>
  <part name='return' element='adm:GetSearchLibraryGroupPolicyResponse' />
</message>

<message name='LookupSearchLibraryGroupListRequest'>
  <part name='body' element='adm:LookupSearchLibraryGroupListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupSearchLibraryGroupListResponse'>
  <part name='return' element='adm:LookupSearchLibraryGroupListResponse' />
</message>

<message name='IsCheckoutHistoryEnabledRequest'>
  <part name='body' element='adm:IsCheckoutHistoryEnabledRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='IsCheckoutHistoryEnabledResponse'>
  <part name='return' element='adm:IsCheckoutHistoryEnabledResponse' />
</message>

<message name='GetGroupResponsibilityPolicyRequest'>
  <part name='body' element='adm:GetGroupResponsibilityPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetGroupResponsibilityPolicyResponse'>
  <part name='return' element='adm:GetGroupResponsibilityPolicyResponse' />
</message>

<message name='LookupGroupResponsibilityPolicyListRequest'>
  <part name='body' element='adm:LookupGroupResponsibilityPolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupGroupResponsibilityPolicyListResponse'>
  <part name='return' element='adm:LookupGroupResponsibilityPolicyListResponse' />
</message>

<message name='LookupBestSellersPolicyListRequest'>
  <part name='body' element='adm:LookupBestSellersPolicyListRequest' />
  <part name='header' element='header:SdHeader' />

```

```

</message>
<message name='LookupBestSellersPolicyListResponse'>
  <part name='return' element='adm:LookupBestSellersPolicyListResponse' />
</message>

<message name='GetBestSellerPolicyRequest'>
  <part name='body' element='adm:GetBestSellerPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetBestSellerPolicyResponse'>
  <part name='return' element='adm:GetBestSellerPolicyResponse' />
</message>

<message name='LookupRecommendedReadingPolicyListRequest'>
  <part name='body' element='adm:LookupRecommendedReadingPolicyListRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupRecommendedReadingPolicyListResponse'>
  <part name='return' element='adm:LookupRecommendedReadingPolicyListResponse' />
</message>

<message name='GetRecommendedReadingPolicyRequest'>
  <part name='body' element='adm:GetRecommendedReadingPolicyRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='GetRecommendedReadingPolicyResponse'>
  <part name='return' element='adm:GetRecommendedReadingPolicyResponse' />
</message>

<message name='IsMailDeliveryEnabledRequest'>
  <part name='body' element='adm:IsMailDeliveryEnabledRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='IsMailDeliveryEnabledResponse'>
  <part name='return' element='adm:IsMailDeliveryEnabledResponse' />
</message>

<message name='IsAccessionEnabledRequest'>
  <part name='body' element='adm:IsAccessionEnabledRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='IsAccessionEnabledResponse'>
  <part name='return' element='adm:IsAccessionEnabledResponse' />
</message>

<portType name='AdminEndpoint'>
  <operation name='lookupILSConfiguration'>
    <input message='tns:LookupILSConfigurationRequest' />
    <output message='tns:LookupILSConfigurationResponse' />
  </operation>
  <operation name='lookupPolicyList'>
    <input message='tns:LookupPolicyListRequest' />
    <output message='tns:LookupPolicyListResponse' />
  </operation>

```

```

</operation>
<operation name='getPolicy'>
  <input message='tns:GetPolicyRequest' />
  <output message='tns:GetPolicyResponse' />
</operation>
<operation name='lookupAccessLevelPolicyList'>
  <input message='tns:LookupAccessLevelPolicyListRequest' />
  <output message='tns:LookupAccessLevelPolicyListResponse' />
</operation>
<operation name='getAccessLevelPolicy'>
  <input message='tns:GetAccessLevelPolicyRequest' />
  <output message='tns:GetAccessLevelPolicyResponse' />
</operation>
<operation name='lookupAccessRestrictionPolicyList'>
  <input message='tns:LookupAccessRestrictionPolicyListRequest' />
  <output message='tns:LookupAccessRestrictionPolicyListResponse' />
</operation>
<operation name='getAccessRestrictionPolicy'>
  <input message='tns:GetAccessRestrictionPolicyRequest' />
  <output message='tns:GetAccessRestrictionPolicyResponse' />
</operation>
<operation name='lookupCitizenshipPolicyList'>
  <input message='tns:LookupCitizenshipPolicyListRequest' />
  <output message='tns:LookupCitizenshipPolicyListResponse' />
</operation>
<operation name='getCitizenshipPolicy'>
  <input message='tns:GetCitizenshipPolicyRequest' />
  <output message='tns:GetCitizenshipPolicyResponse' />
</operation>
<operation name='lookupClearancePolicyList'>
  <input message='tns:LookupClearancePolicyListRequest' />
  <output message='tns:LookupClearancePolicyListResponse' />
</operation>
<operation name='getClearancePolicy'>
  <input message='tns:GetClearancePolicyRequest' />
  <output message='tns:GetClearancePolicyResponse' />
</operation>
<operation name='lookupItemTypePolicyList'>
  <input message='tns:LookupItemTypePolicyListRequest' />
  <output message='tns:LookupItemTypePolicyListResponse' />
</operation>
<operation name='getItemTypePolicy'>
  <input message='tns:GetItemTypePolicyRequest' />
  <output message='tns:GetItemTypePolicyResponse' />
</operation>
<operation name='lookupLanguagePolicyList'>
  <input message='tns:LookupLanguagePolicyListRequest' />
  <output message='tns:LookupLanguagePolicyListResponse' />
</operation>
<operation name='getLanguagePolicy'>
  <input message='tns:GetLanguagePolicyRequest' />
  <output message='tns:GetLanguagePolicyResponse' />
</operation>

```



```

<operation name='lookupLibraryPolicyList'>
  <input message='tns:LookupLibraryPolicyListRequest' />
  <output message='tns:LookupLibraryPolicyListResponse' />
</operation>
<operation name='getLibraryPolicy'>
  <input message='tns:GetLibraryPolicyRequest' />
  <output message='tns:GetLibraryPolicyResponse' />
</operation>
<operation name='lookupLocationPolicyList'>
  <input message='tns:LookupLocationPolicyListRequest' />
  <output message='tns:LookupLocationPolicyListResponse' />
</operation>
<operation name='getLocationPolicy'>
  <input message='tns:GetLocationPolicyRequest' />
  <output message='tns:GetLocationPolicyResponse' />
</operation>
<operation name='lookupMailServicePolicyList'>
  <input message='tns:LookupMailServicePolicyListRequest' />
  <output message='tns:LookupMailServicePolicyListResponse' />
</operation>
<operation name='getMailServicePolicy'>
  <input message='tns:GetMailServicePolicyRequest' />
  <output message='tns:GetMailServicePolicyResponse' />
</operation>
<operation name='isPatronSelfRegistrationEnabled'>
  <input message='tns:IsPatronSelfRegistrationEnabledRequest' />
  <output message='tns:IsPatronSelfRegistrationEnabledResponse' />
</operation>
<operation name='lookupNeedToKnowPolicyList'>
  <input message='tns:LookupNeedToKnowPolicyListRequest' />
  <output message='tns:LookupNeedToKnowPolicyListResponse' />
</operation>
<operation name='getNeedToKnowPolicy'>
  <input message='tns:GetNeedToKnowPolicyRequest' />
  <output message='tns:GetNeedToKnowPolicyResponse' />
</operation>
<operation name='lookupUserLookupPolicyList'>
  <input message='tns:LookupUserLookupPolicyListRequest' />
  <output message='tns:LookupUserLookupPolicyListResponse' />
</operation>
<operation name='getUserLookupPolicy'>
  <input message='tns:GetUserLookupPolicyRequest' />
  <output message='tns:GetUserLookupPolicyResponse' />
</operation>
<operation name='lookupUserSortPolicyList'>
  <input message='tns:LookupUserSortPolicyListRequest' />
  <output message='tns:LookupUserSortPolicyListResponse' />
</operation>
<operation name='getUserSortPolicy'>
  <input message='tns:GetUserSortPolicyRequest' />
  <output message='tns:GetUserSortPolicyResponse' />
</operation>
<operation name='lookupBillReasonPolicyList'>

```

```

    <input message='tns:LookupBillReasonPolicyListRequest' />
    <output message='tns:LookupBillReasonPolicyListResponse' />
  </operation>
  <operation name='getBillReasonPolicy'>
    <input message='tns:GetBillReasonPolicyRequest' />
    <output message='tns:GetBillReasonPolicyResponse' />
  </operation>
  <operation name='lookupPaymentTypePolicyList'>
    <input message='tns:LookupPaymentTypePolicyListRequest' />
    <output message='tns:LookupPaymentTypePolicyListResponse' />
  </operation>
  <operation name='getPaymentTypePolicy'>
    <input message='tns:GetPaymentTypePolicyRequest' />
    <output message='tns:GetPaymentTypePolicyResponse' />
  </operation>
  <operation name='lookupCreditReasonPolicyList'>
    <input message='tns:LookupCreditReasonPolicyListRequest' />
    <output message='tns:LookupCreditReasonPolicyListResponse' />
  </operation>
  <operation name='getCreditReasonPolicy'>
    <input message='tns:GetCreditReasonPolicyRequest' />
    <output message='tns:GetCreditReasonPolicyResponse' />
  </operation>
  <operation name='lookupNativeCurrency'>
    <input message='tns:LookupNativeCurrencyRequest' />
    <output message='tns:LookupNativeCurrencyResponse' />
  </operation>
  <operation name='isUserGroupsEnabled'>
    <input message='tns:IsUserGroupsEnabledRequest' />
    <output message='tns:IsUserGroupsEnabledResponse' />
  </operation>
  <operation name='lookupRequestTypePolicyList'>
    <input message='tns:LookupRequestTypePolicyListRequest' />
    <output message='tns:LookupRequestTypePolicyListResponse' />
  </operation>
  <operation name='getRequestTypePolicy'>
    <input message='tns:GetRequestTypePolicyRequest' />
    <output message='tns:GetRequestTypePolicyResponse' />
  </operation>
  <operation name='getRequestListPolicy'>
    <input message='tns:GetRequestListPolicyRequest' />
    <output message='tns:GetRequestListPolicyResponse' />
  </operation>
  <operation name='lookupRequestListPolicyList'>
    <input message='tns:LookupRequestListPolicyListRequest' />
    <output message='tns:LookupRequestListPolicyListResponse' />
  </operation>
  <operation name='getRequestStatusPolicy'>
    <input message='tns:GetRequestStatusPolicyRequest' />
    <output message='tns:GetRequestStatusPolicyResponse' />
  </operation>
  <operation name='lookupRequestStatusPolicyList'>
    <input message='tns:LookupRequestStatusPolicyListRequest' />

```

```

    <output message='tns:LookupRequestStatusPolicyListResponse' />
  </operation>
  <operation name='isRestrictedAccess'>
    <input message='tns:IsRestrictedAccessRequest' />
    <output message='tns:IsRestrictedAccessResponse' />
  </operation>
  <operation name='getLibraryGroupPolicy'>
    <input message='tns:GetLibraryGroupPolicyRequest' />
    <output message='tns:GetLibraryGroupPolicyResponse' />
  </operation>
  <operation name='lookupLibraryGroupList'>
    <input message='tns:LookupLibraryGroupListRequest' />
    <output message='tns:LookupLibraryGroupListResponse' />
  </operation>
  <operation name='getSearchLibraryGroupPolicy'>
    <input message='tns:GetSearchLibraryGroupPolicyRequest' />
    <output message='tns:GetSearchLibraryGroupPolicyResponse' />
  </operation>
  <operation name='lookupSearchLibraryGroupList'>
    <input message='tns:LookupSearchLibraryGroupListRequest' />
    <output message='tns:LookupSearchLibraryGroupListResponse' />
  </operation>
  <operation name='isCheckoutHistoryEnabled'>
    <input message='tns:IsCheckoutHistoryEnabledRequest' />
    <output message='tns:IsCheckoutHistoryEnabledResponse' />
  </operation>
  <operation name='getGroupResponsibilityPolicy'>
    <input message='tns:GetGroupResponsibilityPolicyRequest' />
    <output message='tns:GetGroupResponsibilityPolicyResponse' />
  </operation>
  <operation name='lookupGroupResponsibilityPolicyList'>
    <input message='tns:LookupGroupResponsibilityPolicyListRequest' />
    <output message='tns:LookupGroupResponsibilityPolicyListResponse' />
  </operation>
  <operation name='lookupBestSellersPolicyList'>
    <input message='tns:LookupBestSellersPolicyListRequest' />
    <output message='tns:LookupBestSellersPolicyListResponse' />
  </operation>
  <operation name='getBestSellerPolicy'>
    <input message='tns:GetBestSellerPolicyRequest' />
    <output message='tns:GetBestSellerPolicyResponse' />
  </operation>
  <operation name='lookupRecommendedReadingPolicyList'>
    <input message='tns:LookupRecommendedReadingPolicyListRequest' />
    <output message='tns:LookupRecommendedReadingPolicyListResponse' />
  </operation>
  <operation name='getRecommendedReadingPolicy'>
    <input message='tns:GetRecommendedReadingPolicyRequest' />
    <output message='tns:GetRecommendedReadingPolicyResponse' />
  </operation>
  <operation name='isMailDeliveryEnabled'>
    <input message='tns:IsMailDeliveryEnabledRequest' />
    <output message='tns:IsMailDeliveryEnabledResponse' />
  </operation>

```

```

</operation>
<operation name='isAccessionEnabled'>
  <input message='tns:IsAccessionEnabledRequest' />
  <output message='tns:IsAccessionEnabledResponse' />
</operation>
</portType>

<binding name='AdminServiceBinding' type='tns:AdminEndpoint'>
  <soap:binding style='document' transport='http://schemas.xmlsoap.org/soap/http' />
  <operation name='lookupILSConfiguration'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupILSConfigurationRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='lookupPolicyList'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupPolicyListRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='getPolicy'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:GetPolicyRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='lookupAccessRestrictionPolicyList'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupAccessRestrictionPolicyListRequest" part="header"
        use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='getAccessRestrictionPolicy'>
    <soap:operation soapAction='' />
    <input>

```

```

        <soap:body parts="body" use="literal"/>
        <soap:header message="tns:GetAccessRestrictionPolicyRequest" part="header" use="literal"/>
    </input>
    <output>
        <soap:body use='literal'/>
    </output>
</operation>
<operation name='lookupAccessLevelPolicyList'>
    <soap:operation soapAction=''/>
    <input>
        <soap:body parts="body" use="literal"/>
        <soap:header message="tns:LookupAccessLevelPolicyListRequest" part="header" use="literal"/>
    </input>
    <output>
        <soap:body use='literal'/>
    </output>
</operation>
<operation name='getAccessLevelPolicy'>
    <soap:operation soapAction=''/>
    <input>
        <soap:body parts="body" use="literal"/>
        <soap:header message="tns:GetAccessLevelPolicyRequest" part="header" use="literal"/>
    </input>
    <output>
        <soap:body use='literal'/>
    </output>
</operation>
<operation name='lookupCitizenshipPolicyList'>
    <soap:operation soapAction=''/>
    <input>
        <soap:body parts="body" use="literal"/>
        <soap:header message="tns:LookupCitizenshipPolicyListRequest" part="header" use="literal"/>
    </input>
    <output>
        <soap:body use='literal'/>
    </output>
</operation>
<operation name='getCitizenshipPolicy'>
    <soap:operation soapAction=''/>
    <input>
        <soap:body parts="body" use="literal"/>
        <soap:header message="tns:GetCitizenshipPolicyRequest" part="header" use="literal"/>
    </input>
    <output>
        <soap:body use='literal'/>
    </output>
</operation>
<operation name='lookupClearancePolicyList'>
    <soap:operation soapAction=''/>
    <input>
        <soap:body parts="body" use="literal"/>
        <soap:header message="tns:LookupClearancePolicyListRequest" part="header"
            use="literal"/>
    </input>
    <output>
        <soap:body use='literal'/>
    </output>
</operation>

```

```

    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='getClearancePolicy'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:GetClearancePolicyRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='lookupItemTypePolicyList'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupItemTypePolicyListRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='getItemTypePolicy'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:GetItemTypePolicyRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='lookupLanguagePolicyList'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupLanguagePolicyListRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='getLanguagePolicy'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:GetLanguagePolicyRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>

```

```

</output>
</operation>
<operation name='lookupLibraryPolicyList'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:LookupLibraryPolicyListRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='getLibraryPolicy'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:GetLibraryPolicyRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='lookupLocationPolicyList'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:LookupLocationPolicyListRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='getLocationPolicy'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:GetLocationPolicyRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='lookupMailServicePolicyList'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:LookupMailServicePolicyListRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='getMailServicePolicy'>

```

```

    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:GetMailServicePolicyRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='isPatronSelfRegistrationEnabled'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:IsPatronSelfRegistrationEnabledRequest" part="header"
use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='lookupNeedToKnowPolicyList'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupNeedToKnowPolicyListRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='getNeedToKnowPolicy'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:GetNeedToKnowPolicyRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='lookupUserLookupPolicyList'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal" />
      <soap:header message="tns:LookupUserLookupPolicyListRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal' />
    </output>
  </operation>
  <operation name='getUserLookupPolicy'>
    <soap:operation soapAction='' />
    <input>

```



```

    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:GetUserLookupPolicyRequest" part="header" use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='lookupUserSortPolicyList'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupUserSortPolicyListRequest" part="header" use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='getUserSortPolicy'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:GetUserSortPolicyRequest" part="header" use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='lookupBillReasonPolicyList'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupBillReasonPolicyListRequest" part="header" use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='getBillReasonPolicy'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:GetBillReasonPolicyRequest" part="header" use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='lookupPaymentTypePolicyList'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupPaymentTypePolicyListRequest" part="header" use="literal"/>
  </input>

```

```

<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='getPaymentTypePolicy'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:GetPaymentTypePolicyRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='lookupCreditReasonPolicyList'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:LookupCreditReasonPolicyListRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='getCreditReasonPolicy'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:GetCreditReasonPolicyRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='lookupNativeCurrency'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:LookupNativeCurrencyRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='isUserGroupsEnabled'>
<soap:operation soapAction=''/>
<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:IsUserGroupsEnabledRequest" part="header" use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>

```

```

    </operation>
    <operation name='lookupRequestTypePolicyList'>
    <soap:operation soapAction=''/>
    <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupRequestTypePolicyListRequest" part="header" use="literal"/>
    </input>
    <output>
    <soap:body use='literal'/>
    </output>
    </operation>
    <operation name='getRequestTypePolicy'>
    <soap:operation soapAction=''/>
    <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:GetRequestTypePolicyRequest" part="header" use="literal"/>
    </input>
    <output>
    <soap:body use='literal'/>
    </output>
    </operation>
    <operation name='getRequestListPolicy'>
    <soap:operation soapAction=''/>
    <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:GetRequestListPolicyRequest" part="header" use="literal"/>
    </input>
    <output>
    <soap:body use='literal'/>
    </output>
    </operation>
    <operation name='lookupRequestListPolicyList'>
    <soap:operation soapAction=''/>
    <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupRequestListPolicyListRequest" part="header"
        use="literal"/>
    </input>
    <output>
    <soap:body use='literal'/>
    </output>
    </operation>
    <operation name='getRequestStatusPolicy'>
    <soap:operation soapAction=''/>
    <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:GetRequestStatusPolicyRequest" part="header" use="literal"/>
    </input>
    <output>
    <soap:body use='literal'/>
    </output>
    </operation>
    <operation name='lookupRequestStatusPolicyList'>

```

```

    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:LookupRequestStatusPolicyListRequest" part="header"
        use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='isRestrictedAccess'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:IsRestrictedAccessRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='getLibraryGroupPolicy'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:GetLibraryGroupPolicyRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='lookupLibraryGroupList'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:LookupLibraryGroupListRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='getSearchLibraryGroupPolicy'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:GetSearchLibraryGroupPolicyRequest" part="header"
        use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='lookupSearchLibraryGroupList'>
    <soap:operation soapAction=''/>

```

```

<input>
  <soap:body parts="body" use="literal"/>
  <soap:header message="tns:LookupSearchLibraryGroupListRequest" part="header"
    use="literal"/>
</input>
<output>
  <soap:body use='literal'/>
</output>
</operation>
<operation name='isCheckoutHistoryEnabled'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:IsCheckoutHistoryEnabledRequest" part="header" use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='getGroupResponsibilityPolicy'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:GetGroupResponsibilityPolicyRequest" part="header"
      use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='lookupGroupResponsibilityPolicyList'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupGroupResponsibilityPolicyListRequest" part="header"
      use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='lookupBestSellersPolicyList'>
  <soap:operation soapAction=''/>
  <input>
    <soap:body parts="body" use="literal"/>
    <soap:header message="tns:LookupBestSellersPolicyListRequest" part="header"
      use="literal"/>
  </input>
  <output>
    <soap:body use='literal'/>
  </output>
</operation>
<operation name='getBestSellerPolicy'>

```

```

    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:GetBestSellerPolicyRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='lookupRecommendedReadingPolicyList'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:LookupRecommendedReadingPolicyListRequest" part="header"
        use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='getRecommendedReadingPolicy'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:GetRecommendedReadingPolicyRequest" part="header"
        use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='isMailDeliveryEnabled'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:IsMailDeliveryEnabledRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='isAccessionEnabled'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:IsAccessionEnabledRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
</binding>

```

```

<!-- custom note here to show it uses our wsdl -->
<service name='AdminService'>
  <port binding='tns:AdminServiceBinding' name='AdminPort'>
    <soap:address location='http://localhost:8080/symws/soap/admin' />
  </port>
</service>
</definitions>

```

## Request service

```

<?xml version="1.0"?>
<definitions name='RequestService'
  targetNamespace='http://www.sirsidynix.com/symws/request'
  xmlns='http://schemas.xmlsoap.org/wsdl/' xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:header='http://www.sirsidynix.com/xmlns/common/header'
  xmlns:req='http://schemas.sirsidynix.com/symws/request'
  xmlns:tns="http://www.sirsidynix.com/symws/request"
  xmlns:xs='http://www.w3.org/2001/XMLSchema'>

  <types>
    <xs:schema>
      <xs:import schemaLocation="request.xsd"
        namespace="http://schemas.sirsidynix.com/symws/request"/>
      <xs:import schemaLocation="header.xsd"
        namespace="http://www.sirsidynix.com/xmlns/common/header"/>
    </xs:schema>
  </types>

  <!-- ILS configuration operations -->
  <message name='LookupRequestTemplateRequest'>
    <part name='body' element='req:LookupRequestTemplateRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='LookupRequestTemplateResponse'>
    <part name='return' element='req:LookupRequestTemplateResponse' />
  </message>

  <message name='CreateRequestRequest'>
    <part name='body' element='req:CreateRequestRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='CreateRequestResponse'>
    <part name='return' element='req:CreateRequestResponse' />
  </message>

  <message name='LookupRequestsRequest'>
    <part name='body' element='req:LookupRequestsRequest' />
    <part name='header' element='header:SdHeader' />
  </message>

```

```

<message name='LookupRequestsResponse'>
  <part name='return' element='req:LookupRequestsResponse' />
</message>

<message name='LookupRequestRequest'>
  <part name='body' element='req:LookupRequestRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='LookupRequestResponse'>
  <part name='return' element='req:LookupRequestResponse' />
</message>

<message name='CancelRequestRequest'>
  <part name='body' element='req:CancelRequestRequest' />
  <part name='header' element='header:SdHeader' />
</message>
<message name='CancelRequestResponse'>
  <part name='return' element='req:CancelRequestResponse' />
</message>

<portType name='RequestEndpoint'>
  <operation name='lookupRequestTemplate'>
    <input message='tns:LookupRequestTemplateRequest' />
    <output message='tns:LookupRequestTemplateResponse' />
  </operation>
  <operation name='createRequest'>
    <input message='tns:CreateRequestRequest' />
    <output message='tns:CreateRequestResponse' />
  </operation>
  <operation name='lookupRequests'>
    <input message='tns:LookupRequestsRequest' />
    <output message='tns:LookupRequestsResponse' />
  </operation>
  <operation name='lookupRequest'>
    <input message='tns:LookupRequestRequest' />
    <output message='tns:LookupRequestResponse' />
  </operation>
  <operation name='cancelRequest'>
    <input message='tns:CancelRequestRequest' />
    <output message='tns:CancelRequestResponse' />
  </operation>
</portType>

<binding name='RequestServiceBinding' type='tns:RequestEndpoint'>
  <soap:binding style='document' transport='http://schemas.xmlsoap.org/soap/http' />
  <operation name='lookupRequestTemplate'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:LookupRequestTemplateRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>

```



```

    </output>
  </operation>
  <operation name='createRequest'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:CreateRequestRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='lookupRequests'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:LookupRequestsRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='lookupRequest'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:LookupRequestRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='cancelRequest'>
    <soap:operation soapAction=''/>
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:CancelRequestRequest" part="header" use="literal"/>
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
</binding>

<!-- custom note here to show it uses our wsdl -->
<service name='RequestService'>
  <port binding='tns:RequestServiceBinding' name='RequestPort'>
    <soap:address location='http://localhost:8080/symws/soap/request'/>
  </port>
</service>
</definitions>

```

## Circulation service

```

<?xml version="1.0"?>
<definitions name='CirculationService'
  targetNamespace='http://www.sirsidynix.com/symws/circulation'
  xmlns='http://schemas.xmlsoap.org/wsdl/'
  xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:header='http://www.sirsidynix.com/xmlns/common/header'
  xmlns:stnd='http://schemas.sirsidynix.com/symws/circulation'
  xmlns:tns="http://www.sirsidynix.com/symws/circulation"
  xmlns:xs='http://www.w3.org/2001/XMLSchema'>

  <types>
    <xs:schema>
      <xs:import schemaLocation="circulation.xsd"
        namespace="http://schemas.sirsidynix.com/symws/circulation"/>
      <xs:import schemaLocation="header.xsd"
        namespace="http://www.sirsidynix.com/xmlns/common/header"/>
    </xs:schema>
  </types>

  <message name='RecordPaymentsRequest'>
    <part name='body' element='stnd:RecordPaymentsRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='RecordPaymentsResponse'>
    <part name='return' element='stnd:RecordPaymentsResponse' />
  </message>

  <message name='LookupMailDeliveryEstimatesInfoRequest'>
    <part name='body' element='stnd:LookupMailDeliveryEstimatesInfoRequest' />
    <part name='header' element='header:SdHeader' />
  </message>
  <message name='LookupMailDeliveryEstimatesInfoResponse'>
    <part name='return' element='stnd:LookupMailDeliveryEstimatesInfoResponse' />
  </message>

  <portType name='CirculationEndpoint'>
    <operation name='recordPayments'>
      <input message='tns:RecordPaymentsRequest' />
      <output message='tns:RecordPaymentsResponse' />
    </operation>
    <operation name='lookupMailDeliveryEstimatesInfo'>
      <input message='tns:LookupMailDeliveryEstimatesInfoRequest' />
      <output message='tns:LookupMailDeliveryEstimatesInfoResponse' />
    </operation>
  </portType>

  <binding name='CirculationServiceBinding' type='tns:CirculationEndpoint'>
    <soap:binding style='document' transport='http://schemas.xmlsoap.org/soap/http' />
    <operation name='recordPayments'>
      <soap:operation soapAction='' />

```

```

    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:RecordPaymentsRequest" part="header" use="literal" />
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
  <operation name='lookupMailDeliveryEstimatesInfo'>
    <soap:operation soapAction='' />
    <input>
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:LookupMailDeliveryEstimatesInfoRequest" part="header"
        use="literal" />
    </input>
    <output>
      <soap:body use='literal'/>
    </output>
  </operation>
</binding>

<!-- custom note here to show it uses our wsdl -->
<service name='CirculationService'>
  <port binding='tns:CirculationServiceBinding' name='CirculationPort'>
    <soap:address location='http://localhost:8080/symws/soap/circulation' />
  </port>
</service>

</definitions>

```

## Catalog service

```

<?xml version="1.0"?>
<definitions name='CatalogService'
  targetNamespace='http://www.sirsidynix.com/symws/catalog'
  xmlns='http://schemas.xmlsoap.org/wsdl/'
  xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:header='http://www.sirsidynix.com/xmlns/common/header'
  xmlns:cat='http://schemas.sirsidynix.com/symws/catalog'
  xmlns:tns="http://www.sirsidynix.com/symws/catalog"
  xmlns:xs='http://www.w3.org/2001/XMLSchema'>

  <types>
    <xs:schema>
      <xs:import schemaLocation="catalog.xsd"
        namespace="http://schemas.sirsidynix.com/symws/catalog"/>
      <xs:import schemaLocation="header.xsd"
        namespace="http://www.sirsidynix.com/xmlns/common/header"/>
    </xs:schema>
  </types>

  <message name='CreateBriefTitleRequest'>
    <part name='body' element='cat:CreateBriefTitleRequest' />
  </message>

```

```

    <part name='header' element='header:SdHeader' />
  </message>
  <message name='CreateBriefTitleResponse'>
    <part name='return' element='cat:CreateBriefTitleResponse' />
  </message>

  <portType name='CatalogEndpoint'>
    <operation name='createBriefTitle'>
      <input message='tns:CreateBriefTitleRequest' />
      <output message='tns:CreateBriefTitleResponse' />
    </operation>
  </portType>

  <binding name='CatalogServiceBinding' type='tns:CatalogEndpoint'>
    <soap:binding style='document' transport='http://schemas.xmlsoap.org/soap/http' />
    <operation name='createBriefTitle'>
      <soap:operation soapAction='' />
      <input>
        <soap:body parts="body" use="literal" />
        <soap:header message="tns:CreateBriefTitleRequest" part="header" use="literal" />
      </input>
      <output>
        <soap:body use='literal' />
      </output>
    </operation>
  </binding>

  <!-- custom note here to show it uses our wsdl -->
  <service name='CatalogService'>
    <port binding='tns:CatalogServiceBinding' name='CatalogPort'>
      <soap:address location='http://localhost:8080/symws/soap/catalog' />
    </port>
  </service>

</definitions>

```

# Index

## 4

404 238

## 5

500 238

501 238

## A

access type 11

AccessLevelPolicy 167

accountability 5, 12, 77

address 208

AddressInfo 194

Admin

service 116, 166, 274, 319

alternate ID 10

Always Require Authentication option 18

authenticateUser operation 70

AuthenticateUserRequest element 70, 79

AuthenticateUserResponse element 70, 80

authentication 5, 143

require 18

availability 65

## B

base URL 3-4

BestSellerDisplayInfo 228

BestSellerPolicy 168

BestSellerTitleInfo 229

BibliographicInfo 59

BillReasonPolicy 168

Boolean 163

BoundwithLinkInfo 60

BriefTitleInfo 229

## C

CallInfo 60

CallSummary 230

cancelMyHold 81

cancelRequest 107

catalog  
search 52, 55

Catalog  
service 295, 347

changeMyAlternativeID 80

changeMyPIN 82

checkout 202, 212

filter 188

history 87, 187, 208

renew 103

CheckoutHistoryRule 187

CheckoutInfoFilter 188

circulation 215

Circulation

service 113, 187, 290, 346

ClearancePolicy 169

client ID 237

clientID

overview 2

clientID element 25

command list 12

Common

service 193, 296

configuration

ILS 150

Connect to host 235

connection to server 236

createMyHold 82

createRequest 108

createSelfRegisteredPatron 85

CreditReasonPolicy 170

currency 15, 155

## D

data

patron 37

data type 163, 165

data types 163

date 164

debugging 34

decimal 164

deleteMyList 86

## E

elements	
AuthenticateUserRequest	70, 79
AuthenticateUserResponse	70, 80
clientID	25
login	70, 72, 80
LoginUserRequest	71
LogoutUserRequest	73
LookupTitleInfoRequest	50
LookupTitleInfoResponse	52
password	70, 72, 80
referenced	59, 76
SdHeader	25
sessionToken	25, 72
userID	72
enableMyCheckoutHistory	87
Entity	194
EntityOperation	195
error	
Bad Client ID	237
Cannot connect to host	235
Entity Operation is required	239
HTTP 404 File not found	238
HTTP 500 Internal server error	238
HTTP 501 Not Implemented	238
Invalid Operation for this request	239
LicenseException.notLicensed	240
onnection to server has been closed	236
Read of response timed out	236
RestInvocationException.missingMethod	240
SecurityServiceException Access Denied	240
Session has timed out	241
Symphony Soap Header is required	237
Unable to acquire session with server	237
Unable to log in	241

## F

fault	
checking	33-35
handling	33
faults	235
FeeInfo	197
FeeInfoFilter	188
FeeItemInfo	198
FeePayment	189
FeePaymentInfo	199
FeePaymentProblem	189
FeePaymentRecorded	190
filter	
checkout	188
fee	188
filters	
SearchCatalogFilters	233

## G

getAccessLevelPolicy	119
getAccessRestrictionPolicy	120
getBestSellerPolicy	121
getBillReasonPolicy	121
getCitizenshipPolicy	122
getClearancePolicy	123-124
getGroupResponsibilityPolicy	125
getItemTypePolicy	126
getLanguagePolicy	126
getLibraryGroupPolicy	127
getLibraryPolicy	128
getLocationPolicy	129
getMailServicePolicy	130
getMyList	88
getNeedToKnowPolicy	131
getPassthroughToken	71
getPaymentTypePolicy	131
getPolicy	132
getRecommendedReadingPolicy	133
getRequestListPolicy	134
getRequestStatusPolicy	135
getRequestTypePolicy	136
getSearchLibraryGroupPolicy	137
getUserGroupResponsibilityPolicy	138
getUserLookupPolicy	139
getUserSortPolicy	140
GroupInfo	200
GroupMemberCheckoutInfo	202
GroupMemberFeeInfo	203
GroupMemberHoldInfo	203
GroupMemberInfo	204
GroupResponsibilityPolicy	171
groups	
library	152
user	144

## H

HatIOException	235-236
HatSessionException	237
header	237
HeaderException	237
headers	27
custom	27
help	xiii, xiv, xiv, xiv, xv, xv, xv
history	
checkout	87, 141
HitlistTitleInfo	61
HoldInactiveType	190
HoldInfoFilter	191
HoldRange	191
holds	81-82, 99-100, 105-106, 191-192, 217
volume	12

HoldType	192
hot sites	46, 231
HotSitesInfo	231

## I

ID	
alt	80
ILS	
access type	11
alternate ID	10
command list	12
configuration	150
language configuration	15
overview	9
pins	11
station library	9
supported versions	1
user access	11
user ID	10
user level	11
InfoDesk	56-57, 231
int	165
invalidRequestException	239
InvalidRequestException	239
isCheckoutHistoryEnabled	141
isMailDeliveryEnabled	142
isPatronSelfRegistrationEnabled	142
isRestrictedAccess	143
isUserGroupsEnabled	144
ItemInfo	61
ItemTypePolicy	170

## J

JSON	29
------	----

## L

language	15
language'	151
LanguagePolicy	172
LibraryGroupPolicy	173
LibraryPolicy	173
license	43, 240
licensed operations	2
LicenseException	240
lists	92-93, 104, 205-207
Lists	86, 88
LocationPolicy	176
log in	241
login element	70, 72, 80
LoginUserRequest element	71
logoutUser operation	72
LogoutUserRequest element	73

long	165
lookupAccessLevelPolicyList	144
lookupAccessRestrictionPolicyList	145
lookupBestSellersDisplayList	44
lookupBestSellersPolicyList	146
lookupBestSellerTitles	45
lookupBillReasonPolicyList	146
lookupCitizenshipPolicyList	147
lookupClearancePolicyList	148
lookupCreditReasonPolicyList	148
lookupGroupResponsibilityPolicyList	149
lookupHotSites	46
lookupILSConfiguration	150
lookupItemTypePolicyList	151
lookupLanguagePolicyList	151
lookupLibraryGroupList	152
lookupLibraryPolicyList	153
lookupLocationPolicyList	153
lookupMailDeliveryEstimatesInfo	113
lookupMailServicePolicyList	154
lookupMostPopular	46
lookupMyAccountInfo	89
lookupMyList	92
lookupMyLists	93
lookupNativeCurrency	155
lookupNeedToKnowPolicyList	155
lookupPatronInfo	94
lookupPaymentTypePolicyList	156
lookupPolicyList	157
LookupRecommendedReadingDisplayList	48
lookupRecommendedReadingPolicyList	158
lookupRecommendedReadingTitles	49
lookupRequest	109
lookupRequestListPolicyList	158
lookupRequests	110
lookupRequestStatusPolicyList	159
lookupRequestTemplate	111
lookupRequestTypePolicyList	160
lookupSearchLibraryGroupList	160
lookupSessionInfo	73
lookupTitleInfo operation	50
LookupTitleInfoRequest element	50
LookupTitleInfoResponse element	52
lookupUserInfo	75
lookupUserLookupPolicyList	161
lookupUserSortPolicyList	162

## M

mail	113, 142, 177, 192
MailDeliveryEstimatesInfo	192
MailServicePolicy	177
manage sessions	31
MarcEntryInfo	63
MarcHoldingsInfo	64
modifyMyHold	99

money	15, 155
Money	195
MostPopularInfo	65
MyList	205
MyListEntry	206
MyListEntryInfo	206
MyListInfo	207
MyListTitleInfo	207

## N

namespace	4
nonNegativeInteger	165

## O

online user registration	16, 85, 142
operation	
Entity	195
operations	
authenticateUser	70
cancelMyHold	81
cancelRequest	107
changeMyAlternativeID	80
changeMyPIN	82
createMyHold	82
createRequest	108
createSelfRegisteredPatron	85
deleteMyList	86
enableMyCheckoutHistory	87
getAccessLevelPolicy	119
getAccessRestrictionPolicy	120
getBestSellerPolicy	121
getBillReasonPolicy	121
getCitizenshipPolicy	122
getClearancePolicy	123-124
getGroupResponsibilityPolicy	125
getItemTypePolicy	126
getLanguagePolicy	126
getLibraryGroupPolicy	127
getLibraryPolicy	128
getLocationPolicy	129
getMailServicePolicy	130
getMyList	88
getNeedToKnowPolicy	131
getPaymentTypePolicy	131
getPolicy	132
getRecommendedReadingPolicy	133
getRequestListPolicy	134
getRequestStatusPolicy	135
getRequestTypePolicy	136
getSearchLibraryGroupPolicy	137
getUserGroupResponsibilityPolicy	138
getUserLookupPolicy	139
getUserSortPolicy	140

isCheckoutHistoryEnabled	141
isMailDeliveryEnabled	142
isPatronSelfRegistrationEnabled	142
isRestrictedAccess	143
isUserGroupsEnabled	144
license	43
logoutUser	72
lookupAccessLevelPolicyList	144
lookupAccessRestrictionPolicyList	145
lookupBestSellersDisplayList	44
lookupBestSellersPolicyList	146
lookupBestSellerTitles	45
lookupBillReasonPolicyList	146
lookupCitizenshipPolicyList	147
lookupClearancePolicyList	148
lookupCreditReasonPolicyList	148
lookupGroupResponsibilityPolicyList	149
lookupHotSites	46
lookupILSConfiguration	150
lookupItemTypePolicyList	151
lookupLanguagePolicyList	151
lookupLibraryGroupList	152
lookupLibraryPolicyList	153
lookupLocationPolicyList	153
lookupMailDeliveryEstimatesInfo	113
lookupMailServicePolicyList	154
lookupMostPopular	46
lookupMyAccountInfo	89
lookupMyList	92
lookupMyLists	93
lookupNativeCurrency	155
lookupNeedToKnowPolicyList	155
lookupPatronInfo	94
lookupPaymentTypePolicyList	156
lookupPolicyList	157
LookupRecommendedReadingDisplayList	48
lookupRecommendedReadingPolicyList	158
lookupRecommendedReadingTitles	49
lookupRequest	109
lookupRequestListPolicyList	158
lookupRequests	110
lookupRequestStatusPolicyList	159
lookupRequestTemplate	111
lookupRequestTypePolicyList	160
lookupSearchLibraryGroupList	160
lookupSessionInfo	73
lookupTitleInfo	50
lookupUserInfo	75
lookupUserLookupPolicyList	161
lookupUserSortPolicyList	162
modifyMyHold	99
overview	41
placeMyHold	100
privileged	18-19
recordPayments	114
renewMyCheckout	103



saveMyList	104
searchCatalog	52
searchCatalogPaging	55
searchInfoDesk	56
searchInfoDeskPaging	57
suspendMyHold	105
unsuspendMyHold	106
version	26, 29, 58
OUR	16, 85, 142
overview	xiii, xiii, 1, 3, 5-6, 9, 13, 17, 20
holds	12

## P

password element	70, 72, 80
Patron	
service	77, 196, 257, 310
PatronAddressInfo	208
PatronCheckoutHistoryInfo	208
PatronCheckoutInfo	212
PatronCirculationInfo	215
PatronHoldInfo	217
PatronInfo	220
PatronStatusInfo	221
PaymentTypePolicy	178
PDF	xiii
Perl	34-35, 37
PERL	35
pins	11, 82
placeMyHold	100
policy	
AccessLevelPolicy	167
BestSellerPolicy	168
BillReasonPolicy	168
ClearancePolicy	169
CreditReasonPolicy	170
getAccessLevelPolicy	119
getAccessRestrictionPolicy	120
getBestSellerPolicy	121
getBillReasonPolicy	121
getCitizenshipPolicy	122
getClearancePolicy	123-124
getGroupResponsibilityPolicy	125
getItemTypePolicy	126
getLanguagePolicy	126
getLibraryGroupPolicy	127
getLibraryPolicy	128
getLocationPolicy	129
getMailServicePolicy	130
getNeedToKnowPolicy	131
getPaymentTypePolicy	131
getPolicy	132
getRecommendedReadingPolicy	133
getRequestListPolicy	134
getRequestStatusPolicy	135
getRequestTypePolicy	136

getSearchLibraryGroupPolicy	137
getUserGroupResponsibilityPolicy	138
getUserLookupPolicy	139
getUserSortPolicy	140
GroupResponsibilityPolicy	171
ItemTypePolicy	170
LanguagePolicy	172
LibraryGroupPolicy	173
LibraryPolicy	173
LocationPolicy	176
MailServicePolicy	177
PaymentTypePolicy	178
PolicyField	178
PolicyInfo	179
PolicyList	180
PolicyRequest	181
RecommendedReadingPolicy	182
RequestListPolicy	183
RequestStatusPolicy	183
RequestTypePolicy	184
SearchLibraryGroupPolicy	185
UserLookupPolicy	186
UserSortPolicy	186
PolicyField	178
PolicyInfo	179
PolicyList	180
PolicyRequest	181
privilege level	11, 143, 227
privileged operations	5, 18-19, 227
PrivilegeLevel	227

## R

RecommendedReadingDisplayInfo	232
RecommendedReadingPolicy	182
RecommendedReadingTitleInfo	232
recordPayments	114
registration	
user	16, 85, 142
renewMyCheckout	103
Request	
service	223, 287, 343
RequestBriefInfo	224
RequestEntry	224
RequestEntryInfo	225
RequestEntryTemplateInfo	225
RequestEntryType	226
RequestInfo	227
RequestListPolicy	183
requests	13, 24, 27, 106-111, 158-160, 183-184, 224-227
JSON	29
testing	22-23
version	26, 29
RequestStatusPolicy	183

RequestTypePolicy	184
RequiredType	184
response	236
REST	27, 29, 35, 37, 240
RestInvocationException	240
restricted	143
rule	
checkout	187

## S

samples	18, 22, 26, 29, 35, 37
saveMyList	104
schema	20, 163, 243, 255, 257, 274, 287, 290, 295-296, 298
SdHeader element	25
search	185, 233-234
searchCatalog	52
SearchCatalogFilters	233
searchCatalogPaging	55
SearchCatalogPaging	233
SearchCatalogResult	234
searchInfoDesk	56
searchInfoDeskPaging	57
SearchLibraryGroupPolicy	185
Security	
service	69, 76, 227, 255, 307
SecurityServiceException	240-241
Selection	
service	298
services	3-4, 41, 243
Admin	116, 166, 274, 319
Catalog	295, 347
Circulation	113, 187, 290, 346
Common	193, 296
endpoints	4
Patron	77, 196, 257, 310
Request	223, 287, 343
Requests	106
Security	69, 76, 227, 255, 307
Selection	298
Standard	42, 59, 228, 243, 301
session	237, 241
sessions	
starting	71
stopping	72
working with	5, 18, 27, 31, 73
sessionToken	5, 18, 27, 31
sessionToken element	25, 72
SOAP	23-24, 26, 34
SpecialUseType	185
Standard	
service	42, 59, 228, 243, 301
station library	9
StatusType	193

string	165
suspendMyHold	105
Symphony ILS, supported versions	1
Symphony Web Services	
about	1, 6, 17, 20
base URL	3
configuring languages	15

## T

test	
requests	22-23
timed out	236
TitleAvailabilityInfo	65
TitleInfo	66
TitleOrderInfo	68
troubleshooting	235
type	
AccessLevelPolicy	167
AccountabilityInfo	77
AddressInfo	194
BestSellerDisplayInfo	228
BestSellerPolicy	168
BestSellerTitleInfo	229
BibliographiInfo	59
BillReasonPolicy	168
Boolean	163
BoundwithLinkInfo	60
BriefTitleInfo	229
CallInfo	60
CallSummary	230
CheckoutHistoryRule	187
CheckoutInfoFilter	188
ClearancePolicy	169
CreditReasonPolicy	170
date	164
dateTime	164
decimal	164
Entity	194
EntityOperation	195
FeeInfo	197
FeeInfoFilter	188
FeeItemInfo	198
FeePayment	189
FeePaymentInfo	199
FeePaymentProblem	189
FeePaymentRecorded	190
GroupInfo	200
GroupMemberCheckoutInfo	202
GroupMemberFeeInfo	203
GroupMemberHoldInfo	203
GroupMemberInfo	204
GroupResponsibilityPolicy	171
HitlistTitleInfo	61
HoldInactiveType	190
HoldInfoFilter	191

HoldRange	191	string	165
HoldType	192	TitleAvailabilityInfo	65
HotSitesInfo	231	TitleInfo	66
InfoDesk	231	TitleOrderInfo	68
int	165	UserLookupPolicy	186
ItemInfo	61	UserSortPolicy	186
ItemTypePolicy	170	UserSuspensionInfo	222
LanguagePolicy	172		
LibraryGroupPolicy	173	<b>U</b>	
LibraryPolicy	173	unsuspendMyHold	106
LocationPolicy	176	URL	46
long	165	URL (Web Services)	3
MailDeliveryEstimatesInfo	192	user access	11
MailServicePolicy	177	user ID	10
MarcEntryInfo	63	user level	11
MarcHoldingsInfo	64	user registration	
Money	195	online	16, 85
MostPopularInfo	65	userID element	72
MyList	205	UserLookupPolicy	186
MyListEntry	206	UserSortPolicy	186
MyListEntryInfo	206	UserSuspensionInfo	222
MyListInfo	207		
MyListTitleInfo	207	<b>V</b>	
nonNegativeInteger	165	version	58
PatronAddressInfo	208	volume holds	12
PatronCheckoutHistoryInfo	208		
PatronCheckoutInfo	212	<b>W</b>	
PatronCirculationInfo	215	WSDL	20, 301, 307, 310, 319, 343, 346-347
PatronHoldInfo	217		
PatronInfo	220	<b>X</b>	
PatronStatusInfo	221	xs	
PaymentTypePolicy	178	boolean	163
PolicyField	178	date	164
PolicyInfo	179	dateTime	164
PolicyList	180	decimal	164
PolicyRequest	181	int	165
PrivilegeLevel	227	long	165
RecommendedReadingDisplayInfo	232	nonNegativeInteger	165
RecommendedReadingPolicy	182	string	165
RecommendedReadingTitleInfo	232	XSD	20, 243, 255, 257, 274, 287, 290, 295-296, 298
RequestBriefInfo	224	XSL	163
RequestEntry	224		
RequestEntryInfo	225		
RequestEntryTemplateInfo	225		
RequestEntryType	226		
RequestInfo	227		
RequestListPolicy	183		
RequestStatusPolicy	183		
RequestTypePolicy	184		
RequiredType	184		
SearchCatalogFilters	233		
SearchCatalogPaging	233		
SearchCatalogResult	234		
SearchLibraryGroupPolicy	185		
SpecialUseType	185		
StatusType	193		

