



Table of Contents

Preface	3
Audience and Prerequisites	3
Overview	3
Integration Overview	3
SOAP Integration	3
REST / Simple XML Integration.....	3
Exception Handling.....	3
UserManager (V1)	5
Description	5
WSDL	5
Login a User.....	5
Description	5
Notes	5
Services	5
Validate a Users Session Id.....	7
Description	7
Notes	7
Services	7

Preface

This document gives a technical overview of the Web Service integration capabilities for INFOplus. It covers the basic application concept and versioning overviews, security and fulfillment services APIs.

Audience and Prerequisites

The intent of this document is to provide a technical audience interested in implementing a SOAP 1.1, SOAP 1.2 or REST style integration with INFOplus with a basic understanding of fulfillment concepts and general web services concepts, SOAP or REST service offerings and Web Service Description Language (WSDL) files.

Overview

This document provides conceptual and technical information for application integration between Materialogic and outside party systems for Materialogic's User Manager authentication services, providing basic concepts in simple use-cases along with pseudo-code examples. The API console and the individual application WSDL URLs are available for in-depth service descriptions.

Integration Overview

The method offered by the UserManager service provide authentication to other Materialogic web services, i.e., the INFOplus Web service.

All APIs provide both SOAP and REST integration options and can be mixed and matched as needed by a client.

SOAP Integration

A SOAP integration path is available for systems that support SOAP 1.1 or SOAP 1.2.

The SOAP protocol is highly verbose and provides for a machine-to-machine communication, where the client has the ability to generate class stubs from WSDL files and then interact directly with the generated code, leaving the protocol and its transport to the supplied framework.

REST / Simple XML Integration

A REST or Simple XML integration path is available for systems that support REST clients, lack a SOAP client framework, or customers wanting to create a custom client based on HTTP GET/POSTs with XML return messages.

Exception Handling

When an error occurs making a request to a web service, information about the error is returned in the server response.

SOAP requests return errors in a SOAP fault element inside the body of the returned SOAP envelope. The fault string element inside the SOAP fault element contains a human readable error message of what problem occurred. Inside the detail element, there is an error element containing the exception name and the aforementioned error message.

XML request return only the error element.

SOAP Exception Example

```
<?xml version="1.0"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Client</faultcode>
      <faultstring>Could not find customer with id [xxx].</faultstring>
      <detail>
        <error>
          <exception>CustomerNotFoundException</exception>
          <message>Could not find customer with id [xxx].</message>
        </error>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

REST / XML Exception Example

```
<error>
  <exception>CustomerNotFoundException</exception>
  <message>Could not find customer with id [xxx].</message>
</error>
```

UserManager (V1)

Description

The UserManager application allows a client to create an active session for a known user account and to test to verify if that session is still valid on an as needed basis.

A valid session is necessary to access all other application services.

WSDL

<http://www.materialogic.com/~soapdev/sas.axis2/services/UserManagerV1?wsdl>

(development server)

Available Protocols

- SOAP 1.1
- SOAP 1.2
- REST

Login a User

Description

A valid username, password and inventory number is required to create a working session for a user. All other services require a working session for access.

A failed login attempt results in an exception being thrown with the appropriate error message.

The loginUser method within the UserManager API ties all calls from this session to the user's inventory.

Notes

- For security reasons, the application returns only a general error message for a bad username and password combination.
- Depending on the account configuration, an account may lock after consecutive failed login attempts.
- Contact the inventory Account Service Rep (ASR) for account configuration details.
- A session becomes invalid after a period of non-use. The default-session timeout value is four hours.
- Contact the inventory ASR for account configuration details.

Services

UserManagerV1.loginUser()

Throws

- RequiredParameterMissingException
- AccessDeniedException
- InvalidSessionException
- SessionException

SOAP Example

```
UserManagerV1 um1 = new UserManagerV1();

Try
{
    String sessionId = um1.loginUser("johndoe", "abcd-1234", 8001);
}
Catch(Error e)
{
    // Could not log in user - look at error message
}
```

REST / XML Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<loginUser xmlns="http://v1.usermgr.external.content.materialogic.com">
<username>demo</username>
<password>abcd-1234</password>
<inventoryNo>9999</inventoryNo>
</loginUser>
```

REST / XML Example Response

```
<ns:loginUserResponse
xmlns:ns="http://v1.usermgr.external.content.materialogic.com">
<return>be8cc5e2-1e03-4c22-90f7-7b321a33f392</return>
</ns:loginUserResponse>
```

Validate a Users Session Id

Description

A client can verify if a given sessionId, returned from a login user request, is still valid via the validateSession service. The service will return a Boolean result indicating if the sessionId is valid or not.

Notes

Each successful call to validateSession will also update the last used timestamp for that session.

For security reasons, consecutive calls to validateSession may be throttled by delaying the response times of the service.

Services

UserManagerV1.validateSession()

Throws

- RequiredParameterMissingException
- AccessDeniedException
- InvalidSessionException
- SessionException

SOAP Example

```
UserManagerV1 uml = new UserManagerV1 ();  
  
String sessionId = "f81d4fae-7dec-11d0-a765-00a0c91e6bf6";  
Boolean isSessionValid = uml.validateSession(sessionId);
```

REST / XML Example Request

```
<?xml version="1.0" encoding="utf-8"?>  
<validateSession  
  xmlns="http://v1.usermgr.external.content.materialogic.com">  
  <sessionId>780d96a5-59ca-4820-8110-4b4e52b62a8e</sessionId>  
</validateSession>
```

REST / XML Example Response

```
<ns:validateSessionResponse  
  xmlns:ns="http://v1.usermgr.external.content.materialogic.com">  
  <return>true</return>  
</ns:validateSessionResponse>
```