



Member Ping Integration Specifications

March 2016

Contents

PERSONIFY WEB SERVICES OVERVIEW	3
MEMBER PING	4
CONNECTING TO PERSONIFY WEB SERVICES.....	4
ACCESSING THE WEB SERVICES	4
AUTHENTICATION & AUTHORIZATION	5
GETTING MEMBERSHIP INFORMATION.....	5
<i>How to interpret.....</i>	<i>5</i>
SUGGESTIONS.....	7
SUGGESTED WORKFLOW	7

Personify Web Services Overview

Personify Data Services was developed using “WCF Data Services” to enable Open Data Protocol (OData) (<http://www.odata.org>) against existing Personify APIs. OData is a web protocol for querying and updating data that provides a way to unlock your data and free it from silos that exist in applications today. OData does this by applying and building upon web technologies such as HTTP, Atom Publishing Protocol (AtomPub), and JSON to provide access to information from a variety of applications, services, and stores. The protocol emerged from experiences implementing AtomPub clients and servers in a variety of products over the past several years. OData is being used to expose and access information from a variety of sources including, but not limited to, relational databases, file systems, content management systems, and traditional websites.

OData is consistent with the way the web works — it makes a deep commitment to URIs for resource identification and commits to an HTTP-based, uniform interface for interacting with those resources (just like the web). This commitment to core web principles allows OData to enable a new level of data integration and interoperability across a broad range of clients, servers, services, and tools. OData exposes data as a resource that is addressable by URIs. Data is accessed and changed by using standard HTTP verbs of GET, PUT, POST, and DELETE. OData uses the entity-relationship conventions of the Entity Data Model to expose resources as sets of entities that are related by associations.

Member Ping

US Lacrosse has a web service designed to provide third-party partners with customer and membership information. This web service replaces the Member Ping web service built in the legacy system, but provides the same level of visibility for US Lacrosse partners. This web service, called USLCustomerDetailView, allows the partner system to call US Lacrosse's AMS in order to identify/verify customers and provide membership details for those customers. Membership details include a customer's membership status (type of membership, membership expiration date) as well as their participation status (official, player, or coach). The partners can then use this information to allow registration of member only events, apply membership discounts, etc.

Connecting to Personify Web Services

Accessing the web services

US Lacrosse has both a test and a production environment. The test environment is periodically "refreshed" with production data, but can be used by partners when building/testing their integration or US Lacrosse enhancements. The connectivity details are generally comparable between the sites to avoid disruptions associated with database refreshes and to facilitate migration to production when enhancements are ready for launch.

The US Lacrosse environments are as follows:

a. Test

- Personify application: <https://usltst.ebiz.uapps.net/PersonifyEbusiness/default.aspx>
- Web Services: <https://usltst.ebiz.uapps.net/PersonifyDataServices/PersonifyDataUSL.svc>

b. Production

- Personify application: <https://usl.ebiz.uapps.net/PersonifyEbusiness/Default.aspx>
- Web Services: <https://usl.ebiz.uapps.net/PersonifyDataServices/PersonifyDataUSL.svc>

You can go to the metadata for the OData services to view all the entities/service operations (entire model) available for use, and to ensure you are viewing the correct schema/model.

c. TEST metadata

- [https://usltst.ebiz.uapps.net/PersonifyDataServices/PersonifyDataUSL.svc/\\$metadata](https://usltst.ebiz.uapps.net/PersonifyDataServices/PersonifyDataUSL.svc/$metadata)

d. PROD metadata

- [https://usl.ebiz.uapps.net/PersonifyDataServices/PersonifyDataUSL.svc/\\$metadata](https://usl.ebiz.uapps.net/PersonifyDataServices/PersonifyDataUSL.svc/$metadata)



Authentication & Authorization

For security reasons, any request to available entities/service operations must be accompanied with authentication information (through Authentication Headers of HTTP Request). The username/password for authentication of web service calls will be provided by US Lacrosse.

The credentials for authentication will be accepted through “Basic Authentication” (at service level). Any request that does not carry authentication information will be changed dynamically using HTTP status code 401 (written using HttpModule).

Getting Membership Information

Personify provides a web service entity called USLCustomerDetailView. This web service will return membership information for the filter criteria passed. If a customer holds more than one membership item, then we would return multiple line items for a single customer. In Personify, membership type for USL/PLAYER/COACH would have its own individual entities' values.

How to interpret

When you call this web service for a customer, you will get:

- 1) No records back—the customer cannot be found in Personify. I.e. no record exists in Personify that meets the criteria.
- 2) One or more records back—
 - a. If all records have **the same** MasterCustomerId, then you have gotten membership information for one customer.
 - i. If there is no information in the Level 1 attribute (or order number), then the customer is not a member.
 - ii. If there is a value of “National” in the Level 1 attribute, then the customer is a member. The corresponding member class will identify his membership type, and the cycle end date will identify when the membership is scheduled to expire.
 - iii. If there is a value of “Participatio” (no n) in the Level 1 attribute, then the member level will identify the participation type.
 - b. If records have **different** MasterCustomerIds, then you have retrieved multiple customers that meet the criteria you specified. You need to use the attributes or refine your query to further refine your search.

The different properties for these entities are:

```
<EntityType Name="USLCustomerDetailView">
- <Key>
  <PropertyRef Name="MasterCustomerId" />
  <PropertyRef Name="SubcustomerId" />
  <PropertyRef Name="OrderNumber" />
```



```

<PropertyRef Name="OrderLineNumber" />
</Key>
<Property Name="MasterCustomerId" Type="Edm.String" Nullable="false" />
<Property Name="SubcustomerId" Type="Edm.Int32" Nullable="false" />
<Property Name="Address1" Type="Edm.String" Nullable="true" />
<Property Name="Address2" Type="Edm.String" Nullable="true" />
<Property Name="AddressStatusCode" Type="Edm.String" Nullable="true" />
<Property Name="City" Type="Edm.String" Nullable="true" />
<Property Name="CountryCode" Type="Edm.String" Nullable="true" />
<Property Name="CycleBeginDate" Type="Edm.DateTime" Nullable="true" />
<Property Name="CycleEndDate" Type="Edm.DateTime" Nullable="true" />
<Property Name="FirstName" Type="Edm.String" Nullable="true" />
<Property Name="InitialBeginDate" Type="Edm.DateTime" Nullable="true" />
<Property Name="LabelName" Type="Edm.String" Nullable="true" />
<Property Name="LastName" Type="Edm.String" Nullable="true" />
<Property Name="Level1" Type="Edm.String" Nullable="true" />
<Property Name="MembershipClass" Type="Edm.String" Nullable="true" />
<Property Name="MembershipLevel" Type="Edm.String" Nullable="true" />
<Property Name="OrderLineNumber" Type="Edm.Decimal" Nullable="false" />
<Property Name="OrderNumber" Type="Edm.String" Nullable="false" />
<Property Name="PostalCode" Type="Edm.String" Nullable="true" />
<Property Name="PrimaryEmailAddress" Type="Edm.String" Nullable="true" />
<Property Name="PrimaryPhone" Type="Edm.String" Nullable="true" />
<Property Name="State" Type="Edm.String" Nullable="true" />
<Property Name="BirthDate" Type="Edm.DateTime" Nullable="true" />
<Property Name="InformalSalutation" Type="Edm.String" Nullable="true" />
<Property Name="MiddleName" Type="Edm.String" Nullable="true" />
<Property Name="NameSuffix" Type="Edm.String" Nullable="true" />
<Property Name="Nickname" Type="Edm.String" Nullable="true" />
<Property Name="UserDefinedSuffix2" Type="Edm.String" Nullable="true" />
<Property Name="VerifiedStatus" Type="Edm.Int32" Nullable="true"/>
</EntityType>

```

Example XML call using a browser is:

```

https://uslstst.ebiz.uapps.net/PersonifyDataServices/PersonifyDataUSL.svc/USLCustomerDetailView
s?$filter=(LastName eq 'Sullivan' and FirstName eq 'Leah' and PrimaryEmailAddress eq
'something@tmaresources.com')

```

Example JSON call using a browser is:

```

https://uslstst.ebiz.uapps.net/PersonifyDataServices/PersonifyDataUSL.svc/USLCustomerDetailView
s?$filter=(LastName eq 'Sullivan' and FirstName eq 'Leah' and PrimaryEmailAddress eq
'something@tmaresources.com')&$format=json

```

For further examples and documentation about OData and REST, please browse:

<http://www.odata.org>

For further examples of customer information for test cases which have specific characteristics, please contact:

helpdesk@uslacrosse.org



Suggestions

Suggested workflow

US Lacrosse suggests a workflow similar to the following:

- Ask the user for their First Name, Last Name, Date of Birth, Zip code, Email Address, and USL CustomerID
- Take this information from the form and query the Member Ping service by the CustomerID
 - If there is a result from the CustomerID, compare the returned results with the entered form data
 - If both sets match, check the CycleEndDate against the current date and your date requirements for account active date.
 - If the date is correct, allow the user to proceed in registration.
 - If the date is incorrect, warn the user that their USL account is expired. Display a link to the USL Join/Renew process to reactivate their account (<http://www.uslacrosse.org/Join>)
 - If the form data does not match, warn the user that there is a data inconsistency and keep the data inside the form for them to change as needed. Continue this until both sets match, and move onto checking the CycleEndDate against the current date and your date requirements for account active date. At this point, you may want to show the user contact information for checking their USL account (membership@uslacrosse.org or 410-235-6882)
 - If there is no result from the CustomerID
 - Warn the user that a USL CustomerID number is required. Display a link to the USL form which allows them to retrieve a USL CustomerID (<https://usl.ebiz.uapps.net/PersonifyEbusiness/Default.aspx?TabID=266>). You may also want to display a link which allows a user to create a new USL Account if they do not have one already (<http://www.uslacrosse.org/Join>)

Displaying the URLs for creating or renewing a USL or retrieving their USL CustomerID at the top of your form is suggested, so that a user can quickly access the USL parts that might slow down their registration on your site.