Web Service Versioning Policy - USI Registry System

Background

The Unique Student Identifier (USI) Registry System provides a web service, containing multiple methods, allowing authorised organisations to initiate direct, system-to-system interactions with the USI Registry System.

Over time, the features of the web service and, as a result, the interface specifications are expected to change.

This brief document describes the policy and procedure for versioning of the USI Registry System Web Service.

The intended readers of this document are:

- the USI Registry System Development team
- the USI Registry System Operations team
- developers of systems (usually Student Management Systems) that consume the USI Registry System Web Service.

Web Service Versioning Policy

Web service versioning for the USI Registry System comprises 2 parts – minor versions and major versions.

The version identifier of the web service and its specification is denoted as MM.NN, where MM denotes the major version number and NN denotes the minor version within that major version. For example, the first Minor Version of Major Version 2 would be denoted as version 2.1. Note that leading zeroes (in both parts) are suppressed.

The version identifier is included in the WSDL as a documentation element, as shown in the example below:

Version identifier example

Minor Versions

As a general rule, all changes to the USI Registry System web service interface specification shall be 'non-breaking'. That is, the changes shall be implemented such that consumers of the service can continue to use existing features without changing their interface. These changes shall result in a Minor Version of the specification.

As a Minor Version does not impact the interface for existing consumers of the service, the endpoint for a Minor Version update shall not change.

Examples of Minor Version updates include, but are not limited to:

- defect fixes, where the interface is not functioning to specification
- new methods
- additional, optional parameters to an existing method.

Major Versions

In the event that a feature of an existing interface must be changed in a way that would 'break' calls to the service by existing consumers, a Major Version would be required.

As there are a large number of consumers of the web service, including hundreds of installations or more, the decision to implement a Major Version must be taken with serious consideration of the impact on the user community.

The release of a new Major Version will result in the creation of a new endpoint for the service, reflecting the Major Version. The previous and new endpoints will co-exist for a period of time – see Version Support below.

Examples of Major Version updates include, but are not limited to:

- change to a parameter format or structure that is incompatible with the existing interface
- a significant change to interface rules, such as making certain optional fields mandatory
- removal of a method from the web service.

Version support

As Minor Versions are non-breaking (backward compatible), only the latest Minor Version, within any Major Version, will be available and supported for consumption by users.

As Major Versions are breaking (not backward compatible), the previous Major Version shall be maintained for use by consumers for a period of 12 months.

In addition, a Major Version shall not be retired within 12 months of it being replaced in production by a new Major Version. That is, consumers will have a minimum of one year to migrate to a new Major Version, in the (exceptional) case that more than one Major Version is made available in a 12-month period.