Web Services API
Implementation Guide

ed2go Content Providers

May 2020

ed2go, a part of Cengage Learning | www.ed2go.com

Contents

Purpose

ed2go has created a Web Services API to facilitate system integration with content providers. Content providers can utilize this API to obtain student course registrations as well as to communicate registration status to *ed2go*. The design goal is to provide a simple, yet robust, method of integration which allows for some level of customization.

This document is a general guide to the API and intended to provide some essential elements that may not be readily apparent from directly accessing the Web service URL. Not every aspect of the API will be used by all content providers.

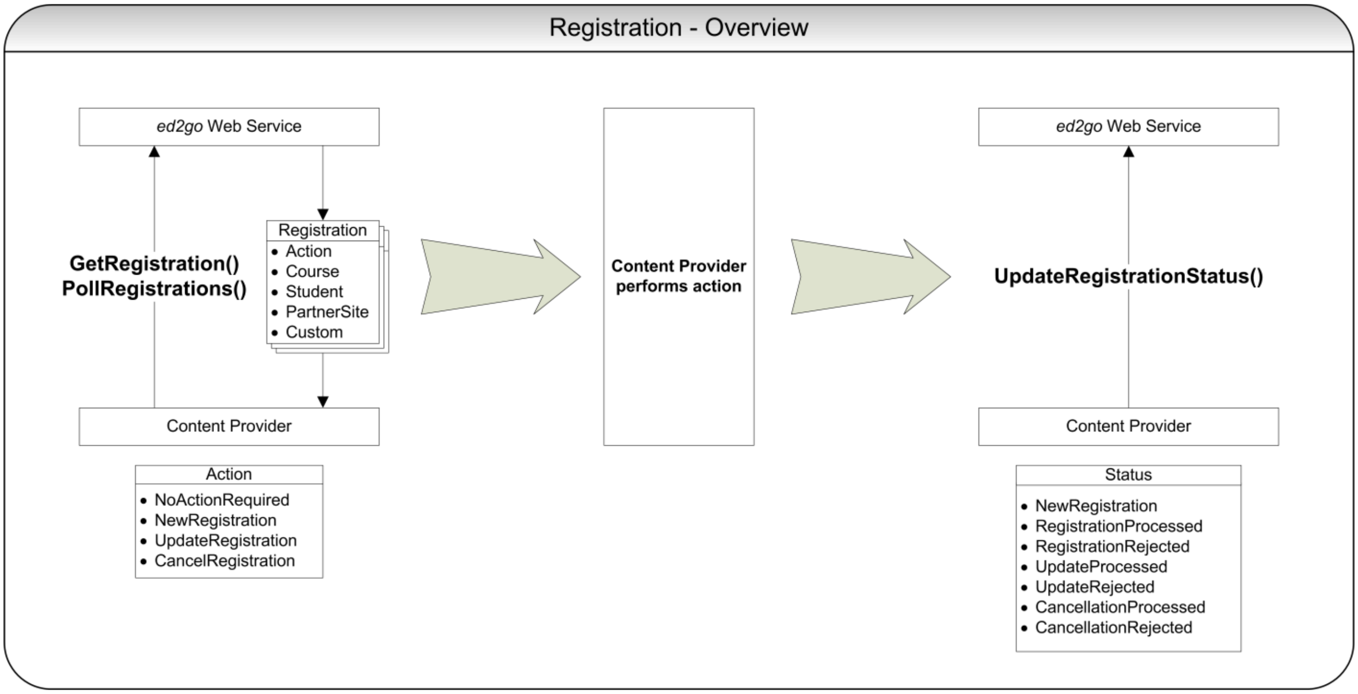
Overview

The API utilizes SOAP/XML Web services that can be used to:

- Retrieve new course registrations, including cancellation notices or course access extensions, if applicable.
- Update *ed2go* on the status of registration processing or student course access.

The API is currently grouped into a single service area—additional areas may be added in the future to extend functionality.

- Registration Service
  https://api.ed2go.com/vendor/1.2/RegistrationService.asmx
  ?WSDL
- Catalog Service
  https://api.ed2go.com/vendor/1.2/CatalogService.asmx



Web Services

General

The following are general descriptions of various data objects and elements found within the API. Further detailed descriptions of listed elements, and items not mentioned, can be found at the Web service URLs above.

Date Formats

All dates used within the API are GMT 24-hour formatted. For example: *2011-11-03T13:27:12Z*

Unique Identifier Keys

Students, partner sites and registrations will include a unique identifier or key. This key will be in the form of a Globally Unique Identifer (GUID). This key can be used to reliably identify, retrieve and update specific entities. An example GUID: *3F2504E0-4F89-11D3-9A0C-0305E82C3301*

Result Object

All service methods will include a result object which should be used to determine the success or failure of the method call.

- *Success*
  A simple Boolean true or false value indicating success or failure.
- *Code*
  A result code that can be used for automated error handling.
- *Message*
  A more detailed description of the result, used to provide more information about a specific failure. This message is informational in nature and its format may change, so the message text should not be used for automated error handling.

Registration Service

Provides methods to retrieve and update status of course registrations.

Service Definition (WSDL)

https://api.ed2go.com/vendor/1.2/RegistrationService.asmx?WSDL

API Key

All service methods accept an API Key first parameter which is used to gain access to the Web service as well as to identify the content provider and associated privileges. This key will be provided by *ed2go*. You will need a different API Key for the production and sandbox environments.

Registration Object

The registration object contains information for a single course registration, including:

- *Registration*
  Registration-level information such as the unique registration key, registration date, action required, current status and access expiration date.
- *Student*
  Student information such as student name, email and unique student key.
- *Course*
  Course information such as the content provider's course identifier, *ed2go's* unique product code and course title.
- *Partner Site*
  Partner's *ed2go* site information such as name and unique partner key.
- *Extensions*
  A list of access extensions added to this registration. **This is for reference only**—the adjusted access expiration date is available in the *AccessExpirationDatetimeGMT* node. Manual adjustments will not be included.
- *Custom*
  If applicable, this element will contain any content provider-specific information as key-value pairs.

Methods

GetCompletionReport()

Returns registration completion report details for provided valid RegistrationKey.

Parameters:

- *APIKey*
- *RegistrationKey*

GetRegistration()

Returns a single registration object with provided valid registration key.

Parameters:

- *APIKey*
- *RegistrationKey*

GetRegistrationsByFilter()

Returns a list of registration objects based on input filter, such as student name, registration date, course, partner site, action or registration status.

Parameters:

- *APIKey*
- *Filter* – Filter object with properties set to values for exact match.

PollRegistrations()

Returns a list of registration objects that need action performed by content provider, such as *NewRegistration*, *CancelRegistration* or *NewExtension*. A registration with appear only once in a single poll and will be found in subsequent polls until its status has been updated.

Parameters:

- *APIKey*
- *MaxCount* – Allows throttling of return list. A value of zero or less will return all registrations requiring action.

UpdateCompletionReport()

Allows content provider to report progress and completion data to registration.

Parameters:

- *APIKey*
- *RegistrationKey*
- *PercentProgress* – Percentage of progress through the course (0-100).
- *LastAccessDatetimeGMT* – Date of student last access.
- *CoursePassed* – A simple Boolean true or false value indicating whether the student passed/did not pass the course.
- *PercentOverallScore* – Student's overall score as a percentage (0-100).
- *CompletionDatetimeGMT* – Date student completed course.
- *TimeSpent* – Total time recorded in course for student.
- *Custom* – If applicable, this element will contain any content provider-specific information as key-value pairs. These can be created at-will.

UpdateRegistrationStatus()

Allows content provider to update registration status. Once action is taken on a registration received and the status updated by the content provider, the registration will be removed from the results of *PollRegistrations()*.

Parameters:

- *APIKey*

- *RegistrationKey*
- *ReferenceID* – Increasing value to help ensure synchronization. Located in the *Registration* object.
- *RegistrationStatus* – Valid status code (See *Statuses* below for full list).
- *Note* – Informational message to include with status update *(Can be empty).*

The update will be unsuccessful if the:

- *RegistrationStatus* no longer corresponds to the current action state of the registration.
- *ReferenceID* is no longer in sync with the *ed2go* system.

In both cases, the content provider can obtain the latest action state or reference ID using *GetRegistration()* or handle it in the next call to *PollRegistrations()*.

Actions

A registration requiring action by the content provider will appear in the results of the *PollRegistrations()* method until the status is updated by the content provider.

NoActionRequired

Registration requires no further action by the content provider. A registration with this action will not appear in the results of *PollRegistrations().*

NewRegistration

Content provider should create a new course enrollment for the registered student. The *RegistrationKey* must be associated with your enrollment record.

UpdateRegistration

Content provider should update all information for registration with provided *RegistrationKey*. A key value to compare and update would be *AccessExpirationDatetimeGMT*.

CancelRegistration

Content provider should cancel the registration for *RegistrationKey* and cut off student access immediately.

Statuses

The status of a registration is maintained by the content provider via *UpdateRegistrationStatus()* method. The status given must correspond to the action requested or to a valid current status. Once the status has been updated to for an action, the registration will no longer appear in the results of *PollRegistrations().*

NewRegistration

This is the default initial status for a new registration. You cannot update a registration to this status.

RegistrationProcessed

The new registration has been processed (i.e. course is prepared for student access). Valid status update for action: *NewRegistration*.

RegistrationRejected

The new registration cannot or will not be processed for some reason. Valid status update for action: *NewRegistration*.

UpdateProcessed

The registration information was updated. Valid status update for action: *UpdateRegistration*.

UpdateRejected

The registration information cannot or will not be processed for some reason. Valid status update for action: *UpdateRegistration*.

CancellationProcessed

The registration was successfully canceled and course access removed. Valid status update for action: *CancelRegistration*.

CancellationRejected

The registration information cannot or will not be processed for some reason. Valid status update for action: *CancelRegistration*.

Catalog Service

Provides methods to add and update catalog-oriented course data.

Service Definition (WSDL)

https://api.ed2go.com/vendor/1.2/CatalogService.asmx?WSDL

API Key

All service methods accept an API Key first parameter which is used to gain access to the Web service as well as to identify the content provider and associated privileges. This key will be provided by *ed2go*.

Methods

AddCourse()

Create a new course with catalog descriptions and information. New courses will remain in a pending status until approved by *ed2go* staff.

Parameters:

- *APIKey*
- *CourseID*
- *CourseTitle*
- *LongDescriptionHtmlEncoded*
- *ShortDescriptionHtmlEncoded*

- *SpacesaverDescriptionHtmlEncoded*
- *PrerequisitesHtmlEncoded*
- *ObjectivesHtmlEncoded*
- *FAQsHtmlEncoded*
- *RequirementsHtmlEncoded*
- *Instructor*
- *Language*
- *Hourse*
- *DurationOfAccess*
- *Certificate*
- *Category*
- *CECredits*
- *CEType*
- *RetailPriceUSD*
- *CostUSD*
- *AvailabilityStartDatetimeGMT*
- *AvailabilityEndDatetimeGMT*

GetCourse()

Returns current course detail and status for course ID specified. Can return more than one course object if there is a pending and non-pending version of the course.

Parameters:

- *APIKey*
- *CourseID*

GetCourseAvailability()

Get the current availability dates for an existing course. Can return more than one summary course object if there is a pending and non-pending version of the course.

Parameters:

- *APIKey*
- *CourseID*

GetCoursesByFilter()

Returns a list of courses base on input filter parameters such as availability, course ID or status.

Parameters:

- *APIKey*
- *Filter* – Filter object with properties set to values for exact match.

UpdateCourse()

Update an existing course's catalog descriptions and information. Updates will remain in a pending status until approved by *ed2go* staff. However, changes to availability dates will take effect immediately.

Parameters:

- *APIKey*
- *CourseID*
- *CourseTitle*
- *LongDescriptionHtmlEncoded*
- *ShortDescriptionHtmlEncoded*
- *SpacesaverDescriptionHtmlEncoded*
- *PrerequisitesHtmlEncoded*
- *ObjectivesHtmlEncoded*
- *FAQsHtmlEncoded*
- *RequirementsHtmlEncoded*
- *Instructor*
- *Language*
- *Hourse*
- *DurationOfAccess*
- *Certificate*
- *Category*
- *CECredits*
- *CEType*
- *RetailPriceUSD*
- *CostUSD*
- *AvailabilityStartDatetimeGMT*
- *AvailabilityEndDatetimeGMT*

UpdateCourseAvailability()

Update availability dates for an existing course. Changes to availability dates will take effect immediately.

Parameters:

- *APIKey*
- *CourseID*
- *StartDatetimeGMT*
- *EndDatetimeGMT*

Single Sign On (SSO)

ed2go students should be able to seamlessly access their course at a content provider's website directly from an *ed2go* partner website.

Native SSO Model

For SSO, *ed2go* currently implements a design where *ed2go* passes authentication variables via the student browser to a secure URL on your system. You are responsible for creating a URL on your system that would accept the authentication variables, validate the request, then redirect the student to the correct location on your system based on the values provided.

Basic Flow

The basic SSO flow is as follows.

- Student logs in at *ed2go* partner site
- Student selects "Enter Classroom"
- *ed2go* directs student browser to post authentication variables to your SSO URL in new window. In the event of an authentication failure, your system should display a "Login failed" message and give the student the option to return to *ReturnURL*.

HTTP Post Variables

Form POST Variable	Description	Example Value
Checksum	SHA-1 message digest of posted values and a shared secret. This can be used to validate and authenticate each request.	7344fba7065d649f6ed0051f28a11a61981b15fe
RegistrationKey	Unique identifier for this course registration in the form of a GUID	9279eb93-3427-4705-8f53-4bc4c6efe322
RequestExpirationDatetimeGMT	Expiration date of this SSO request in GMT 24-hour format.	2012-02-15T23:59:59Z
ReturnURL	A URL that can be used to return the student to the *ed2go* system	https://www.ed2go.com/hlcc/classroom.html
Source	Identifies the source of the request	ed2go

Security

The request is secured by using an HTTP POST over SSL, a checksum and request expiration.

Checksum

A checksum will be included with each post to ensure that all posted values were supplied by *ed2go* and have not been manipulated. The checksum is created by a SHA-1 message digest of the following concatenated URL-decoded values with no line-end characters:

- Pre-determined shared secret (determined by the value of *Source*). This will be your API Key.
- All posted field values, ordered by field name, excluding:
    - *Checksum*
    - *Source*

Request Expiration

A posted SSO request should not be honored if it is after the included *RequestExpirationDatetimeGMT*.

Example SSO POST

The following is a mock example of a raw HTTP form post with example SSO data.

```
POST http://www.contentprovider.com/your/sso-page.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: www.contentprovider.com
Content-Length: 231
Expect: 100-continue
Connection: Keep-Alive

Checksum=7344fba7065d649f6ed0051f28a11a61981b15fe&RegistrationKey=9279eb93-3427-4705-8f53-
4bc4c6efe322&RequestExpirationDatetimeGMT=2012-02-
15T23%3a59%3a59Z&ReturnURL=https%3a%2f%2fwww.ed2go.com%2fhlcc%2fclassroom.html&Source=ed2go
```

The above *Checksum* value was calculated using a shared secret value of “abcdefg”. Here is the raw string that was encoded using SHA-1.

```
abcdefg9279eb93-3427-4705-8f53-4bc4c6efe3222012-02-15T23:59:59Zhttps://www.ed2go.com/hlcc/classroom.html
```

Action Request Notification (preferred)

Rather than polling the *ed2go* system regularly, some may prefer to be notified only when an action is needed. The following notification mechanism can be implemented to achieve this goal. The design is similar to the SSO model described earlier using HTTP POST requests.

Basic Flow

The basic action request notification flow is as follows.

- A *NewRegistration*, *UpdateRegistration* or *CancelRegistration* event has occurred.
- *ed2go* posts a notification to the vendor system via a simple HTTPS form post.

- Vendor system calls *GetRegistration* method and processes request.

HTTP Post Variables

Form POST Variable	Description	Example Value
Checksum	SHA-1 message digest of posted values and a shared secret. This can be used to validate and authenticate each request.	f11242bcf0b0d1b84ad916ecd4a40b9a9d8012e0
Action	Name of method to call. Will only be GetRegistration at this time.	GetRegistration
RegistrationKey	Unique identifier for course registration affected in the form of a GUID.	9279eb93-3427-4705-8f53-4bc4c6efe322
RequestExpirationDatetimeGMT	Expiration date of this SSO request in GMT 24-hour format.	2012-02-15T23:59:59Z
Source	Identifies the source of the request	ed2go

If multiple registrations require action, a notification for each registration will be sent to the vendor system. Notifications will be sent regularly until the the registration status is updated using *UpdateRegistrationStatus* method.

Security

The request is secured by using an HTTP POST over SSL, a checksum and request expiration.

Checksum

A checksum will be included with each post to ensure that all posted values were supplied by *ed2go* and have not been manipulated. The checksum is created by a SHA-1 message digest of the following concatenated URL-decoded values with no line-end characters:

- Pre-determined shared secret (determined by the value of *Source*). This will be your API Key.
- All posted field values, ordered by field name, excluding:
  - *Checksum*
  - *Source*

Request Expiration

A posted SSO request should not be honored if it is after the included *RequestExpirationDatetimeGMT*.

Example Action Request POST

The following is a mock example of a raw HTTP form post with example action request data.

```
POST http://www.contentprovider.com/your/action-page.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: www.contentprovider.com
Content-Length: 193
Expect: 100-continue
Connection: Keep-Alive

Action=GetRegistration&Checksum= f11242bcf0b0d1b84ad916ecd4a40b9a9d8012e0&RegistrationKey=9279eb93-3427-
4705-8f53-4bc4c6efe322&RequestExpirationDatetimeGMT=2012-02-15T23%3a59%3a59Z&Source=ed2go
```

The above *Checksum* value was calculated using a shared secret value of “abcdefg”. Here is the raw string that was encoded using SHA-1.

```
abcdefgGetRegistration9279eb93-3427-4705-8f53-4bc4c6efe3222012-02-15T23:59:59Z
```

Sandbox Environment

You may utilize a sandbox environment to develop and test your API implementation. Data in the sandbox environment may be periodically refreshed without notice. To use the sandbox, replace the production URLs outlined earlier with the following:

- Registration Service
  https://x-api.ed2go.com/vendor/1.2/RegistrationService.asmx
- Catalog Service
  http://x-api.ed2go.com/vendor/1.2/CatalogService.asmx

You will need to obtain a separate sandbox API Key from *ed2go*.

Versions

All versions of the API will remain backward compatible. If compatibility cannot be maintained when adding or changing functionality, a new version will be deployed under a separate URL. Content providers will be notified if an API version will become discontinued.