

API Integration Specification



Kryterion owns the copyright to this document. All rights are reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language or computer language, in any form by any means including electronic, mechanical, magnetic, optical, chemical, manual or otherwise without the prior written permission of Kryterion at 7776 S. Pointe Parkway West, Suite 200, Phoenix, AZ 85044. Copyright 2020 Kryterion, Inc. Kryterion and Webassessor are trademarks of Drake International. All other brand and product names are trademarks or registered trademarks of their respective companies.

Table of Contents

Requirements and Specifications	3
Security Token	3
Ping	3
Add User Web Service	4
Edit User Web Service	7
Edit User Single Field Web Service	10
Get User Web Service	14
Get User by ID Web Service	16
Get Users by Date Range Web Service	18
Delete User Web Service	19
Get Custom Fields Web Service	20
Get Transcripts by User Web Service	21
Get Transcripts by Date Range Web Service	24
Add Transcript (single topic) Web Service	26
Add Transcript (multitopic) Web Service	28
Get Registrations Web Service	29
Get Registrations by Product Code Web Service	31
Get Registrations by Date Range Web Service	33
Cancel Registration Web Service	35
Get Products Web Service	37
Get Cert Status by User Web Service	38
Get Cert Info by Date Range Web Service	40
Get Client Regions Web Service	41
Add Registration Web Service	42
Add Registration with Validation Web Service	54
Add Item Web Service	59
Edit Item Web Service	63
Get Unassigned Vouchers by Batch Web Service	66
Get Coupons Web Service	67
Add Voucher Batch Web Service	68
Assign Voucher Batch to Candidate Users Web Service	71
Activate Deactivate Voucher Batch Web Service	73
Get Registration by Hash Key Web Service	77
Update Lab Progress Web Service	80
Valid Country and Province Codes	82
Error Codes and Validation Rules	82

Integration Web Service API

Requirements and Specifications

Kryterion Web Services makes use of Java features not available until Java version 6, which is the minimum recommended version. Kryterion Web Services are accessed by passing and accepting JSON formatted strings as specified in this document.

Security Token

A security token is required to use Kryterion Web Services. To obtain your security token, contact your Program Manager.

Web Service Detail

Ping

The ping web service can be used to ensure that web services are available. It can be accessed via a URL:

URI Template

<https://www.webassessor.com/WebAssessorWebServices/jaxrs/wa webservices/ping>

End-Point URL:

<https://www.webassessor.com/WebAssessorWebServices/jaxrs/wa webservices/processRequest>

Request Parameters

URL Parameters None

Response Parameters

Hitting the ping web service will result in text being sent back to the browser: "Ping Successful!"

Possible Error Messages

None from within the code. Possible error messages would be HTTP error codes (i.e. 404 - Page Not Found, etc.)

Add User Web Service

The Add User web service allows you to add a user with the role of a Test Taker to Webassessor™.

Request Parameters

These are the allowable field names (* indicates optional fields):

Add User Request Parameters

requestType	addressStreet1
securityToken	addressStreet2*
login (user's login name, must be unique)	addressCity
firstName	addressState
lastName	addressPostalCod
email	addressCountry
password	userDefined1
sendEmail	notes
returnFormat	homePhone
customFields(Object)	

(Please look below for request fields)

Calling the Add User Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "ADD_USER",
  "securityToken": "*****" "login": "testadd",
  "firstName": "testi",
  "lastName": "xxxxx",
  "email": "prashsdf@gmail.com",
  "returnFormat": "JSON",
  "addressCity": "Columbia",
  "addressCountry": "United States",
  "addressPostalCode": "21044",
  "addressState": "ARIZONA",
  "addressStreet1": "11000 Broken Land Parkway",
  "addressStreet2": "",
  "homePhone": "",
  "customFields": [
    {"id": "831", "stringValue": "No"},
    {"id": "2089231", "numberValue": "1235"},
    {"id": "2089229", "intValue": "1"},
    {"id": "2089230", "dateValue": "2018-12-23"},
    {"id": "2092028", "optionValue": "Chevy"},
    {"id": "2099419", "stringValue": "hi"}
  ],
  "sendEmail": false
}
```

Example Add User Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of an Add User web service call in JSON. {

```
"addressCity": "Columbia",
"addressCountry": "United States",
"addressPostalCode": "21044",
"addressState": "ARIZONA",
"addressStreet1": "11000 Broken Land Parkway",
"addressStreet2": [],
"clientRegion": {
  "id": "1",
  "name": "Kryterion Default Region"
},
"customFieldValue": [
  {
    "customFieldId": "831",
    "customFieldText": "Approved to test",
    "intValue": "0",
    "stringValue": "No"
  },
  {
    "customFieldId": "2089230",
    "customFieldText": "Date Custom Field",
    "dateValue": "2018-12-23T00:00:00-07:00",
    "intValue": "0"
  },
  {
    "customFieldId": "2089229",
    "customFieldText": "Integer Custom Field",
    "intValue": "1"
  },
  {
    "customFieldId": "2099419",
    "customFieldText": "test",
    "intValue": "0"
  },
  {
    "customFieldId": "2089231",
    "customFieldText": "Numeric Custom Field",
    "intValue": "0",
    "numberValue": "1235.0"
  },
  {
    "customFieldId": "2092028",
    "customFieldText": "Test View",
    "intValue": "0",
    "stringValue": "Chevy"
  }
],
"email": "prashsdf@gmail.com",
```

```
"firstName": "testi",  
"homePhone": [],
```

```
"id": "5766445",  
"lastName": "xxxxx",  
"login": "testadd",  
"password": "@$ckBa6Mh9"
```

```
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00002 - NULL LOGIN
WAWSE-00004 - NULL/INVALID FIRST NAME
WAWSE-00006 - NULL PASSWORD
WAWSE-00007 - NULL PASSWORD SECRET
WAWSE-00013 - NULL/INVALID LAST NAME
WAWSE-00014 - NULL ADDRESS STREET
WAWSE-00015 - NULL ADDRESS CITY
WAWSE-00016 - NULL ADDRESS POSTAL CODE
WAWSE-00017 - NULL ADDRESS COUNTRY
WAWSE-00018 - INVALID COUNTRY
WAWSE-00019 - INVALID STATE/PROVINCE
WAWSE-00021 - INVALID FAX
WAWSE-00022 - INVALID PHONE
WAWSE-00024 - INVALID LOGIN FORMAT
WAWSE-00026 - INVALID POSTAL CODE
WAWSE-00027 - INVALID NAME
WAWSE-00029 - INVALID STREET ADDRESS
WAWSE-00034 - INVALID PROVINCE FOR COUNTRY
WAWSE-00035 - INVALID ADDRESS CITY
WAWSE-00039 - INVALID FAX LENGTH
WAWSE-00040 - INVALID PHONE LENGTH
WAWSE-00043 - INVALID SECRET LENGTH
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Edit User Web Service

The Add User web service allows you to edit an existing user in Webassessor™.

Request Parameters

These are the allowable field names (* indicates an optional field):

Edit User Request Parameters

requestType (must be set to EDIT USER)	securityToken
login (user's login name, must be unique)	addressStreet1*
firstName*	addressStreet2*
lastName*	addressCity*
email*	addressState*
notes*	addressPostalCode*
homePhone*	addressCountry*
custom fields*	sendEmail
returnFormat	

Calling the Edit User Web Service with JSON

The edit user API resides in the external-webservices project. Prior to release 18, the client sends all the fields mentioned below as part of the web service request in order to edit a single user field.

Request:

```
{
  "requestType": "EDIT USER",
  "securityToken": "0dhcg632-ld99-26nc-1ld9-chsdcg9m823hh",
  "login": "testadd",
  "firstName": "Greg",
  "lastName": "White11",
  "email": "prashsdf@gmail.com",
  "returnFormat": "JSON",
  "addressCity": "Columbia",
  "addressCountry": "USA",
  "addressPostalCode": "21044",
  "addressState": "US-MD",
  "addressStreet1": "11000 Broken Land Parkway East",
  "addressStreet2": "",
  "homePhone": "",

  "customFields":
  [
    {"id": "2092028", "stringValue": "No"}
  ]
}
```

```
}  
  
...  
  
{  
  "addressCity": "Columbia",  
  "addressCountry": "United States",  
  "addressPostalCode": "21044",  
  "addressState": "Maryland",
```



```
    "id": "1",
    "name": "Kryterion Default Region"
  },
  "customFieldValue": [
    {
      "customFieldId": "831",
      "customFieldText": "Approved to test",
      "intValue": "0",
      "stringValue": "No"
    },
    {
      "customFieldId": "2089230",
      "customFieldText": "Date Custom Field",
      "dateValue": "2018-12-23T00:00:00-07:00",
      "intValue": "0"
    },
    {
      "customFieldId": "2089229",
      "customFieldText": "Integer Custom Field",
      "intValue": "1"
    },
    {
      "customFieldId": "2099419",
      "customFieldText": "test",
      "intValue": "0"
    },
    {
      "customFieldId": "2089231",
      "customFieldText": "Numeric Custom Field",
      "intValue": "0",
      "numberValue": "1235.0"
    },
    {
      "customFieldId": "2092028",
      "customFieldText": "Test View",
      "intValue": "0",
      "stringValue": "No"
    }
  ],
  "email": "prashsdf@gmail.com",
  "firstName": "Greg",
  "homePhone": [],
  "id": "5766445",
  "lastName": "White11",
  "login": "testadd"
}
```

Edit User Single Field Web Service

Allow clients to update a single user or custom field without altering the remaining fields.

Update Last Name only

Request:

```
{
  "requestType": "EDIT USER",
  "securityToken": "0d****[REDACTED]****",
  "returnFormat": "JSON",
  "login": "testtaker1",
  "lastName": "Edison"
}
```

Request:

```
{
  "addressCity": "Phoenix",
  "addressCountry": "United States",
  "addressPostalCode": "85048",
  "addressState": "Arizona",
  "addressStreet1": "4486 Dye Street",
  "addressStreet2": "",
  "clientRegion": {
    "id": "1866",
    "name": "SomeClientRegionName"
  },
  "customFieldValue": [
    {
      "intValue": "0",
      "stringValue": "Private"
    },
    {
      "intValue": "0",
      "stringValue": "Yes"
    }
  ],
  "email": "jvwmxv3cvsc@fakemailgenerator.net",
  "firstName": "Dorothy",
  "homePhone": "4807594113",
  "id": "981863",
  "lastName": "Edison",
  "login": "testtaker1"
}
```

```
"id": "981863",  
"lastName": "Edison",  
"login": "testtaker1"  
}
```

Update First Name Only

Request:

```
{  
  "requestType": "EDIT USER",  
  "securityToken": "0d****[REDACTED]****",  
  "returnFormat": "JSON",  
  "login": "testtaker1",  
  "firstName": "Freddie"  
}
```

Request:

```
{  
  "addressCity": "Phoenix",  
  "addressCountry": "United States",  
  "addressPostalCode": "85048",  
  "addressState": "Arizona",  
  "addressStreet1": "4486 Dye Street",  
  "addressStreet2": "",  
  "clientRegion": {  
    "id": "1866",  
    "name": "SomeClientRegionName"  
  },  
  "customFieldValue": [  
    {  
      "intValue": "0",  
      "stringValue": "Private"  
    },  
    {  
      "intValue": "0",  
      "stringValue": "Yes"  
    }  
  ],  
  "email": "jvwmxv3cvsc@fakemailgenerator.net",  
  "firstName": "Freddie",  
  "homePhone": "4807594113",  
  "id": "981863",  
  "lastName": "Edison",  
  "login": "testtaker1"  
}
```

Note: Prior to release 18, sending only the interested field(s) as part of the request will cause the web service to throw validation errors like "You must supply a last name and should not be more than 50-character length."

Update Custom Field Values

The edit user API allows clients to add just the custom fields to the payload without having to include all the user fields.

Request:

```
{
  "requestType": "EDIT USER",
  "securityToken": "0d****[REDACTED]****",
  "returnFormat": "JSON",
  "login": "testtaker1",
  "customFields":
  [
    {"id": "2085399", "stringValue": "Private"},
    {"id": "2085398", "stringValue": "No"}
  ]
}
```

Request:

```
{
  "addressCity": "Phoenix",
  "addressCountry": "United States",
  "addressPostalCode": "85048",
  "addressState": "Arizona",
  "addressStreet1": "4486 Dye Street",
  "addressStreet2": "",
  "clientRegion": {
    "id": "1866",
    "name": "SomeClientRegionName"
  },
  "customFieldValue": [
    {
      "customFieldId": "2085399",
      "customFieldText": "Your role focuses primarily on which sector?",
      "intValue": "0",
      "stringValue": "Private"
    },
    {
      "customFieldId": "2085398",
      "customFieldText": "Verification Opt-In:",
      "intValue": "0",
      "stringValue": "No"
    }
  ]
}
```

```
],  
  "email": "jvwmxv3cvsc@fakemailgenerator.net",  
  "firstName": "Freddie",  
  "homePhone": "4807594113",  
  "id": "981863",  
  "lastName": "Edison",  
  "login": "testtaker1"  
}
```

Update only a single custom field value

Clients can update a single custom field value without sending all the custom field values.

Request:

```
{  
  "requestType": "EDIT USER",  
  "securityToken": "0d****[REDACTED]****",  
  "returnFormat": "JSON",  
  "login": "testtaker1",  
  "customFields":  
    [  
      {"id": "2085398", "stringValue": "Yes"}  
    ]  
}
```

Request:

```
{  
  "addressCity": "Phoenix",  
  "addressCountry": "United States",  
  "addressPostalCode": "85048",  
  "addressState": "Arizona",  
  "addressStreet1": "4486 Dye Street",  
  "addressStreet2": "",  
  "clientRegion": {  
    "id": "1866",  
    "name": "SomeClientRegionName"  
  },  
  "customFieldValue": [  
    {  
      "customFieldId": "2085399",  
      "customFieldText": "Your role focuses primarily on which sector?",  
      "intValue": "0",  
      "stringValue": "Private"  
    },  
    {  
      "customFieldId": "2085398",
```

```
        "customFieldText": "Verification Opt-In:",  
        "intValue": "0",  
        "stringValue": "Yes"  
    }  
],  
    "email": "jvwmxv3cvsc@fakemailgenerator.net",  
    "firstName": "Freddie",  
    "homePhone": "4807594113",  
    "id": "981863",  
    "lastName": "Edison",  
    "login": "testtaker1"  
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00002 - NULL LOGIN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00018 - INVALID COUNTRY
WAWSE-00019 - INVALID STATE/PROVINCE
WAWSE-00021 - INVALID FAX
WAWSE-00022 - INVALID PHONE
WAWSE-00024 - INVALID LOGIN FORMAT
WAWSE-00026 - INVALID POSTAL CODE
WAWSE-00027 - INVALID NAME
WAWSE-00029 - INVALID STREET ADDRESS
WAWSE-00034 - INVALID PROVINCE FOR COUNTRY
WAWSE-00035 - INVALID ADDRESS CITY
WAWSE-00039 - INVALID FAX LENGTH
WAWSE-00040 - INVALID PHONE LENGTH
WAWSE-00043 - INVALID SECRET LENGTH
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR
OCCURRED
WAWSE-00068 - INVALID CITIZENSHIP COUNTRY

Get User Web Service

The Get User web service allows you to retrieve user information from Webassessor™.

Request Parameters

These are the allowable field names (* indicates an optional field):

Get User Request Parameters

requestType (must be set to GET USER)
login

securityToken
returnFormat

Calling the Get User Web Service with JSON

Here is an example of JSON object:

```
}  
{  
  "requestType": "GET_USER",  
  "securityToken": "12345678-1234-1234-1234-123456789012",  
  "login": "JohnDoe",  
  "returnFormat": "JSON"  
}
```

Example Get User Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of a Get User web service call in JSON.

```
{  
  "addressCity": "Chicago",  
  "addressCountry": "United States",  
  "addressPostalCode": "60013-0001",  
  "addressState": "Illinois",  
  "addressStreet1": "555 Main Street",  
  "addressStreet2": "Suite 99", "clientRegion":  
  {  
    "id": "1234",  
    "name": "Acme Co. North America"  
  },  
  "email": "john@beatles.com",  
  "fax": "(480) 555-1212",  
  "firstName": "John",  
  "homePhone": "911-9999",  
  "id": "425660",  
  "lastName": "Lennon",  
  "login": "JohnLennon",  
  "customFields": [  
    {  
      "id": "181",  
      "intValue": "0",  
      "numberValue": "0",  
      "stringValue": "xy2434"  
    },  
    {  
      "id": "182",  
      "intValue": "0",  
      "numberValue": "0",  
      "stringValue": "ABCDEFGH"  
    }  
  ]  
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00002 - NULL LOGIN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00008 - INVALID LOGIN
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00024 - INVALID LOGIN FORMAT
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Get User by ID Web Service

The Get User by ID web service allows you to retrieve user info from Webassessor™ by the user object's numerical ID.

Request Parameters

These are the allowable field names (* indicates an optional field):

Get User by ID Request Parameters

requestType (must be set to GET USER BY ID) securityToken
userId ReturnFormat

Calling the Get User by ID Web Service with JSON

Here is an example of JSON object:

```
{
  "requestType": "GET USER BY ID ",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "userId": 1234,
  "returnFormat": "JSON"
}
```

Example Get User by ID Web Service Call JSON Return Value

The following is an example of what the return value would look like upon successful completion of a Get User by ID web service call in JSON.

```
{
  "addressCity": "Chicago",
  "addressCountry": "United States",
  "addressPostalCode": "60013-0001",
  "addressState": "Illinois",
  "addressStreet1": "555 Main Street",
}
```



```
"addressStreet2": "Suite 99", "clientRegion":  
{  
  "id": "1234",  
  "name": "Acme Co. North America"  
},  
"email": "john@beatles.com",  
"fax": "(480) 555-1212",  
"firstName": "John",  
"homePhone": "911-9999",  
"id": "425660",  
"lastName": "Lennon",  
"login": "JohnLennon",  
"customFields": [  
  {  
    "id": "181",  
    "intValue": "0",  
    "numberValue": "0",  
    "stringValue": "xy2434"  
  },  
  {  
    "id": "182",  
    "intValue": "0",  
    "numberValue": "0",  
    "stringValue": "ABCDEFGH"  
  }  
]
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00010 - INVALID USER ID
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Get Users by Date Range Web Service

The Get Users by Date Range web service allows you to retrieve user info from Webassessor™ by within a specified date range.

Request Parameters

These are the allowable field names (* indicates an optional field):

Get Users by Date Range Request Parameters

requestType (set to GET USERS BY Date Range)

securityToken

startDateYYYY-MM-DD 00:00:00

endDateYYYY-MM-DD 00:00:00

ReturnFormat

Calling the Get Users by Date Range Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "GET USERS BY DATE RANGE ",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "startDate": "2022-01-01",
  "endDate": "2022-12-31",
  "returnFormat": "JSON"
}
```

Example Get Users by Date Range Web Service Call JSON Return Value

The following is an example of what the return value would look like upon successful completion of a Get Users by Date Range web service call in JSON.

```
{
  "addressCity": "Chandler",
  "addressCountry": "United States",
  "addressPostalCode": "85286",
  "addressState": "Arizona",
  "addressStreet1": "12345",
  "addressStreet2": [],
  "clientRegion": {
    "id": "16674",
    "name": "Jon_Testing"
  },
  "customFieldValue": {
    "customFieldId": "2099980",
    "customFieldText": "Test",
    "intValue": "0",
    "stringValue": "yes"
  },
  "email": "jdoe@kryteriononline.com(666666666666)",
}
```

```
"firstName": "Jon",  
  "homePhone": [],  
  "id": "5866340",  
  "lastName": "Doe",  
  "login": "jon_tester",  
  "dateCreated":  
  "dateModified":  
  "Active/Inactive":  
  "userDefined1": []  
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00041 - INVALID DATE RANGE CODE
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Delete User Web Service

The Delete User web service allows you to delete a Test Taker in Webassessor™.

Please note that for safety reasons, the record is not actually deleted, but rather inactivated.

Request Parameters

These are the allowable field names (*indicates an optional field)

Delete User Request Parameters

requestType (must be set to GET USER BY ID)	securityToken
login	ReturnFormat

Calling the Delete User Web Service with JSON

Here is an example JSON object:

```
{  
  "requestType": "DELETE USER",  
  "securityToken": "12345678-1234-1234-1234-123456789012",  
  "login": "JohnDoe",  
  "returnFormat": "JSON"  
}
```

Example Delete User Web Service Call JSON Return Value

The following is an example of what the return value would look like upon a successful completion of a Delete User web service call in JSON.

```
{  
  "id": "0",  
  "login": "JohnLennon"  
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00002 - NULL LOGIN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00008 - INVALID LOGIN
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00024 - INVALID LOGIN FORMAT
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Get Custom Fields Web Service

The Get Custom Fields web service allows you to retrieve a list of all available custom fields so that you may look up the correct custom field id to be used in other web services.

For instance, in order to use the Add User web service, you must supply custom field ids for any custom fields that you associate with the user. This web service allows you to look them up.

Request Parameters

These are the allowable field names (* indicates an optional field):

Get Custom Fields Request Parameters

requestType (must be GET CUSTOM FIELDS)

returnFormat
securityToken

Calling the Get Custom Fields Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "GET_CUSTOM_FIELDS",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "returnFormat": "JSON"
}
```

Example Get Custom Fields Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of a Get Custom Fields web service call in JSON.

```
{
  "id": "123456789",
  "required": "false",
  "resourceKeyName": "Some custom field:", "type": "STRING"
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Get Transcripts by User Web Service

The Get Transcripts by User web service allows you to retrieve all transcripts for a given user.

Note: **Possible values for the below values in the response for registration object:**

Progress: COMPLETED, SCHEDULED, IN_PROGRESS, SCHEDULEDSUSPENDED

Status: NORMAL, PAUSED, SUSPENDED, CANCELLED, PENDING_SALE

Request Parameters

These are the allowable field names (* indicates an optional field):

Get Transcripts by User Request Parameters

requestType must be set to GET TRANSCRIPTS BY USER
(login or candidateld)
SecurityToken
returnFormat

Calling the Get Transcripts by User Web Service with JSON

Here is an example JSON object using “login”:

```
{
  "requestType": "GET TRANSCRIPTS BY USER",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "returnFormat": "JSON",
  "login": "johnDoe"
}
```

Here is an example JSON object using “candidateld”:

```
{
  "requestType": "GET TRANSCRIPTS BY USER",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "returnFormat": "JSON",
  "candidateld": "12345"
}
```

Example Get Transcripts by User Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of a Get Transcripts by User web service call in JSON.

```
{
  "certNumber": "0",
  "date": "2008-06-24T08:49:42-07:00",
  "id": "46381",
  "maxScore": "148",
  "passFail": "NA",
  "scaledScore": "0",
  "score": "115.0",
  "simpleRegistration": {
    "candidate": "12345",
    "id": "99999",
    "productId": "9876",
    "progress": "COMPLETED",
    "registrationDate": "2008-06-24T06:43:42-07:00",
    "status": "NORMAL",
    "testName": "Example Test",
    "testingCenterId": "0",
    "timeElapsedInMillis": "0",
    "timeRemainingInMillis": "0"
  }
}
```

```
    },
    "status": "ACTIVE",
    "topicScores": [

      {
        "code": "ABCD00001",
        "maxScore": "15.0"
        "name": "1. Basic Knowledge",
        "score": "13.0"
      },
      "testRescoreId": "0"
    ],
  },
  {
    "certNumber": "0",
    "date": "2008-06-25T08:49:42-07:00",
    "id": "46382",
    "maxScore": "184",
    "passFail": "Pass",
    "scaledScore": "0",
    "score": "151.0",
    "simpleRegistration": {
      "candidate": "12345",
      "id": "88888",
      "productId": "5432",
      "progress": "COMPLETED",
      "registrationDate": "2008-06-25T06:43:42-07:00",
      "status": "NORMAL",
      "testName": "Example Test Part II",
      "testingCenterId": "0",
      "timeElapsedInMillis": "0",
      "timeRemainingInMillis": "0"
    },
    "status": "ACTIVE",
    "testRescoreId": "0"
  }
]

[
  {
    "code": "ABCD00001",
    "maxScore": "15.0"
    "name": "1. Basic Knowledge",
    "score": "13.0"
  },
  "testRescoreId": "0"
],
{
  "certNumber": "0",
  "date": "2008-06-25T08:49:42-07:00",
  "id": "46382",
  "maxScore": "184",
  "passFail": "Pass",
  "scaledScore": "0",
```

```
"score": "151.0",
"simpleRegistration": {
  "candidate": "12345",
  "id": "88888",
  "productId": "5432",
  "progress": "COMPLETED",
  "registrationDate": "2008-06-25T06:43:42-07:00",
  "status": "NORMAL",
  "testName": "Example Test Part II",
  "testingCenterId": "0",
  "timeElapsedInMillis": "0",
  "timeRemainingInMillis": "0"
},
"status": "ACTIVE",
"testRescoreId": "0"
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00002 - NULL LOGIN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED
WAWSE-00049 - DATE RANGE TOO LONG
WAWSE-00088 - TOO MANY FIELDS PROVIDED

Get Transcripts by Date Range Web Service

The Get Transcripts by Date Range web service allows you to retrieve all transcripts for a given date range. The date range between startDate and endDate is now limited to 7 days.

If there are more than ~800 records, The API returns 404 Error.

Note: Possible values for the below values in the response for registration object:

Progress: COMPLETED, SCHEDULED, IN_PROGRESS, SCHEDULEDSUSPENDED **Status:** NORMAL, PAUSED, SUSPENDED, CANCELLED, PENDING_SALE

Request Parameters

These are the allowable field names (* indicates an optional field):

Get Transcripts by Date Range Request Parameters

requestType (must be set to TRANSCRIPTS BY DATE RANGE)
startDate
endDate

SecurityToken
returnFormat

Calling the Get Transcripts by Date Range Web Service with JSON

Here is an example of JSON object:

```
{
  "requestType": "GET TRANSCRIPTS BY DATE RANGE",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "returnFormat": "JSON",
  "startDate": "2010-06-01 00:00:00",
  "endDate": "2010-06-08 00:00:00"
}
```

Example Get Transcripts by Date Range Web Service Call JSON Return Value

The following is an example of what the return value will look like upon the successful completion of a Get Transcripts by Date Range web service call in JSON.

```
{
  "certNumber": "0",
  "date": "2008-06-24T08:49:42-07:00",
  "id": "46381",
  "maxScore": "148",
  "passFail": "NA",
  "scaledScore": "0",
  "score": "115.0",
  "simpleRegistration": {
    "candidate": "12345",
    "id": "99999",
    "productId": "9876",
    "progress": "COMPLETED",
    "registrationDate": "2008-06-24T06:43:42-07:00",
    "status": "NORMAL",
    "testName": "Example Test",
    "testingCenterId": "0",
    "timeElapsedInMillis": "0",
    "timeRemainingInMillis": "0"
  },
  "status": "ACTIVE",
  "topicScores": [
    {
      "code": "ABCD00001",
      "maxScore": "15.0",
      "name": "1. Basic Knowledge",
      "score": "13.0"
    }
  ],
  "testRescoreId": "0"
},
{
  "certNumber": "0",
```

```
"date": "2008-06-25T08:49:42-07:00",
"id": "46382",
"maxScore": "184",
"passFail": "Pass",
"scaledScore": "0",
"score": "151.0",
"simpleRegistration": {
  "candidate": "12345",
  "id": "88888",
  "productId": "5432",
  "progress": "COMPLETED",
  "registrationDate": "2008-06-25T06:43:42-07:00",
  "status": "NORMAL",
  "testName": "Example Test Part II",
  "testingCenterId": "0",
  "timeElapsedInMillis": "0",
  "timeRemainingInMillis": "0"
},
"status": "ACTIVE",
"testRescoreId": "0"
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00038 - NULL DATE
WAWSE-00042 - UNPARSEABLE DATE
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED
WAWSE-00049 - DATE RANGE TOO LONG

Add Transcript (single topic) Web Service

The Add Transcript web service allows you to add a transcript with a single topic for a given user and registration.

Request Parameters

These are the allowable fields (*indicates an optional field):

Add Transcript Request Parameters

requestType (must be set to ADD TRANSCRIPT)

score

securityToken

transcriptDate

maxScore

passFail

transcriptStatus

registrationId

returnFormat

certNumber

Calling the Add Transcript Web Service with single topic with JSON

Here is an example JSON object:

```
{
  "requestType": "ADD TRANSCRIPT",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "transcriptDate": "2013-06-17 07:35:52",
  "maxScore": "2",
  "passFail": "PASS",
  "score": "2",
  "transcriptStatus": "ACTIVE",
  "registrationId": "123456",
  "returnFormat": "JSON",
  "sendEmail": true
}
```

Example Add Transcript with (Single Topic) Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of an Add Transcript web service call in JSON.

```
{
  "certNumber": "0",
  "date": "2014-04-17T07:35:52-07:00",
  "id": "1234567",
  "maxScore": "100",
  "passFail": "PASS",
  "scaledScore": "0",
  "score": "95",
  "emailSent": true,
  "simpleRegistration": {
    "candidate": {
      "addressCity": "Phoenix",
      "addressCountry": "United States",
      "addressPostalCode": "85044",
      "addressState": "Arizona",
      "addressStreet1": "555 Main Street",
      "addressStreet2": "Suite 1b",
      "clientRegion": {
        "id": "1234",
        "name": "Acme Co. North America"
      }
    },
    "email": "jro@randomemail.com", "fax": [],
    "firstName": "Jane",
    "homePhone": [],
    "id": "7777",
    "lastName": "Roe",
    "login": "jro@randomemail.com", "userDefined1": [],
    "userDefined2": [],
    "userDefined3": []
  }
}
```

```
        "userDefined4": [],
        "userDefined5": []
    },
    "id": "987654",
    "productCode": "PR000001",
    "productId": "7777",
    "progress": "COMPLETED",
    "registrationDate": "2014-03-12T18:09:01-07:00",
    "status": "NORMAL",
    "testName": "Test",
    "testingCenterId": "0",
    "timeElapsedInMillis": "0",
    "timeRemainingInMillis": "0"
},
"status": "ACTIVE"
}
```

Add Transcript (multitopic) Web Service

The Add Transcript web service allows you to add a transcript with multiple topics for a given user and registration

Calling the Add Transcript Web Service with Multi topic with JSON

```
{
  "registrationId": "24234324",
  "transcriptStatus": "ACTIVE",
  "score": 48,
  "sendEmail": false,
  "requestType": "ADD TRANSCRIPT",
  "maxScore": 50,
  "returnFormat": "JSON",
  "transcriptDate": "2014-11-25 16:31:47",
  "passFail": "PASS",
  "securityToken": "3123123123123123",
  "topicScore":
  [
    { "code": "TP000332", "score": 3, "maxScore": 3},
    { "code": "TP000333", "score": 3, "maxScore": 3},
    { "code": "TP000334", "score": 10, "maxScore": 10},
    { "code": "TP000335", "score": 3, "maxScore": 3},
    { "code": "TP000336", "score": 2, "maxScore": 2},
    { "code": "TP000337", "score": 8, "maxScore": 8},
    { "code": "TP000338", "score": 8, "maxScore": 8},
    { "code": "TP000339", "score": 6, "maxScore": 8},
    { "code": "TP000340", "score": 3, "maxScore": 3},
    { "code": "TP000341", "score": 2, "maxScore": 2}
  ]
}
```

Possible Error Messages

WAWSE-00012 - INVALID REGISTRATION ID

The registration id was not found

WAWSE-00052 - TRANSCRIPT STATUS IS UNKNOWN

The transcript status message is invalid

WAWSE-00051 - PASS FAIL VALUE IS UNKNOWN

The pass fail value is unknown

WAWSE-00000 - NULL SECURITY TOKEN

WAWSE-00001 - INVALID SECURITY TOKEN

WAWSE-00003 - NULL REQUEST TYPE

WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR

WAWSE-00044 - INVALID REQUEST TYPE

WAWSE-00050 - PASS FAIL INDICATION IS NULL

WAWSE-00051 - TRANSCRIPT STATUS IS NULL

WAWSE-00053 - TRANSCRIPT DATE IS NULL

WAWSE- 00060 - AN UNEXPECTED ERROR OCCURRED

Get Registrations Web Service

The Get Registrations web service allows you to retrieve all transcripts for a given user.

Note: **Possible values for the below values in the response:**

Progress: COMPLETED, SCHEDULED, IN_PROGRESS, SCHEDULEDSUSPENDED

Status: NORMAL, PAUSED, SUSPENDED, CANCELLED, PENDING_SALE

Request Parameters

These are the allowable field names (*indicates an optional field):

Get Registrations Request Parameters

requestType (must be set to GET REGISTRATIONS BY USER)

login

securityToken

returnFormat

Calling the Get Registrations Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "GET REGISTRATIONS",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "returnFormat": "JSON",
  "login": "johndoe"
}
```

Example Get Registrations Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of a Get Registrations web service call in JSON.

[

```
{
  "authCode": "ywgeyp",
  "candidate": {
    "addressCity": "New Orleans",
    "addressCountry": "United States",
    "addressPostalCode": "70130",
    "addressState": "Louisiana",
    "addressStreet1": "650 Poydras St. ",
    "addressStreet2": [],
    "clientRegion": { "id": "4427",
      "name": "INFORMS"
    },
  },
  "email": "test@kryterion.xom",
  "fax": [],
  "firstName": "John",
  "homePhone": [],
  "id": "683715",
  "lastName": "Smith683715",

  "login": "jarmcbez"
},
"eventDate": "2014-05-03T00:00:00-07:00",
"id": "2135201",
"notes": [],
"productCode": "PR000001",
"productId": "19220",
"progress": "COMPLETED",
"regTransactionId": "0",
"registrationDate": "2014-04-21T08:44:11-07:00",
"status": "NORMAL",
"testName": "CAP Form B",
"testingCenterCity": "LaPlace",
"testingCenterCountry": "United States",
"testingCenterId": "4265",
"testingCenterName": "The Testing Center, Inc.",
"testingCenterState": "Louisiana",
"timeElapsedInMillis": "0",
"timeRemainingInMillis": "0",
"timeZone": "America/Chicago"
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY
TOKEN
WAWSE-00002 - NULL LOGIN

WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR
OCCURRED
WAWSE-00049 - DATE RANGE TOO LONG

Get Registrations by Product Code Web Service

The Get Registrations by Product Code web service allows you to retrieve all transcripts for a given product. Now the Date range between *startDate* and *endDate* is limited to 7 days.

If there are more than ~800 records, The API returns 404 Error.

Note: **Possible values for the below values in the response:**

Progress: COMPLETED, SCHEDULED, IN_PROGRESS, SCHEDULEDSUSPENDED

Status: NORMAL, PAUSED, SUSPENDED, CANCELLED, PENDING_SALE

Request Parameters

These are the allowable field names (* indicates an optional field):

Get Registrations by Product Code Request Parameters

requestType (must be set to GET REGISTRATIONS BY PRODUCT CODE)
productCode
SecurityToken
startDate
endDate
returnFormat

Calling the Get Registrations by Product Code Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "GET REGISTRATIONS BY PRODUCT CODE",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "productCode": "PR000001",
  "startDate": "2008-06-15 00:00:00",
  "endDate": "2008-06-15 00:00:00",
  "returnFormat": "JSON"
}
```

Example Get Registrations by Product Code Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of a Get Registrations by Product Code web service call in JSON.

```
[
  {
```

```
"authCode": "ywgeyp",
"candidate": {
  "addressCity": "New Orleans",
  "addressCountry": "United States",
  "addressPostalCode": "70130",
  "addressState": "Louisiana",
  "addressStreet1": "650 Poydras St.",
  "addressStreet2": [],
  "clientRegion": { "id": "4427",
    "name": "INFORMS"
  },
},
"email": "test@kryterion.xom",
"fax": [],
"firstName": "John",

"homePhone": [],
"id": "683715",
"lastName": "Smith683715",
"login": "jarmcbez"
},
"eventDate": "2014-05-03T00:00:00-07:00",
"id": "2135201",
"notes": [],
"productCode": "PR000001",
"productId": "19220",
"progress": "COMPLETED",
"regTransactionId": "0",
"registrationDate": "2014-04-21T08:44:11-07:00",
"status": "NORMAL",
"testName": "CAP Form B",
"testingCenterCity": "LaPlace",
"testingCenterCountry": "United States",
"testingCenterId": "4265",
"testingCenterName": "The Testing Center, Inc.",
"testingCenterState": "Louisiana",
"timeElapsedInMillis": "0",
"timeRemainingInMillis": "0",
"timeZone": "America/Chicago"
},
{
  "authCode": "xypegd",
  "candidate": {
    "email": "test@kryterion.xom",
    "fax": [],
    "firstName": "John",
```



```
"homePhone": [],  
"id": "683844",  
"lastName": "Smith683844",  
"login": "gbennett"  
},  
"eventDate": "2014-04-25T00:00:00-07:00",  
"id": "2135548",  
"notes": [],  
"productCode": "PR000001",  
"productId": "19220",  
"progress": "COMPLETED",  
"regTransactionId": "0",  
"registrationDate": "2014-04-21T12:38:34-07:00",  
"status": "NORMAL",  
"testName": "CAP Form B",  
"testingCenterCity": "Columbia",  
"testingCenterCountry": "United States",  
"testingCenterId": "12341",  
"testingCenterName": "Phoenix TS",  
"testingCenterState": "Maryland",  
"timeElapsedInMillis": "0",  
"timeRemainingInMillis": "0",  
"timeZone": "America/New_York"  
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00038 - NULL DATE
WAWSE-00041 - INVALID DATE RANGE CODE
WAWSE-00042 - UNPARSEABLE DATE
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00048 - NULL PRODUCT CODE
WAWSE-00049 - DATE RANGE TOO LONG
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Get Registrations by Date Range Web Service

The Get Registrations by Product Code web service allows you to retrieve all transcripts for a given product. The start date and end date are the modified date and not the created date. Now event date and location details are added in the response. Now the Date range between startDate and endDate is limited to 7 days.

If there are more than ~800 records, the API returns 404 Error.

Note: **Possible values for the below values in the response:**

Progress: COMPLETED, SCHEDULED, IN_PROGRESS, SCHEDULEDSUSPENDED
Status: NORMAL, PAUSED, SUSPENDED, CANCELLED, PENDING_SALE

Request Parameters

These are the allowable field names (* indicates an optional field):

Get Registrations by Date Range Request Parameters

requestType (must be set to GET REGISTRATIONS BY DATE RANGE)
startDate
endDate
securityToken
returnFormat

Calling the Get Registrations by Date Range Code Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "GET REGISTRATIONS BY DATE RANGE",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "startDate": "2017-06-01 00:00:00",
  "endDate": "2017-06-03 00:00:00",
  "returnFormat": "JSON"
}
```

Example Get Registrations by Date Range Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of a Get Registrations by Date Range web service call in JSON.

```
[
  {
    "authCode": "cqsdpx",
    "candidate": {
      "addressCity": "HTP5D6QYLV",
      "addressCountry": "United States",
      "addressPostalCode": "85044",
      "addressState": "Arizona",
      "addressStreet1": "XNGZOA8XFP368OC",
      "addressStreet2": [],
      "citizenship": "United States",
      "clientRegion": {
        "id": "1",
        "name": "Kryterion Default Region"
      },
    },
    "email": "test@kryterion.xom(5)",
    "fax": [],
    "firstName": "John22222222222222222222222222222222",
    "homePhone": [],
  }
]
```

```
    "id": "27889",  
    "lastName": "Smith278892222222222222222222222222222",  
    "login": "thomasbrun",  
    "userDefined1": []  
  },  
  "eventDate": "2017-05-31T00:00:00-07:00",  
  "id": "3758273",  
  "notes": "SYSTEM CANCELLED",  
  "productCode": "PR000115",  
  "productId": "16930",  
  "progress": "SCHEDULED",  
  "regTransactionId": "0",  
  "registrationDate": "2017-05-16T13:42:06-07:00",  
  "status": "CANCELLED",  
  "testingCenterCity": "Phoenix",  
  "testingCenterCountry": "United States",  
  "testingCenterId": "1241",  
  "testingCenterName": "Compatibility Testing Only",  
  "testingCenterState": "Arizona",  
  "timeElapsedInMillis": "0",  
  "timeRemainingInMillis": "0",  
  "timeZone": "America/Phoenix"  
} ]
```

Possible Error Messages

WAWSE-00001 - INVALID SECURITY
TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE
ERROR
WAWSE-00038 - NULL DATE
WAWSE-00042 - UNPARSEABLE DATE
WAWSE-00044 - INVALID REQUEST
TYPE
WAWSE-00048 - NULL PRODUCT
CODE
WAWSE-00060 - AN UNEXPECTED ERROR
OCCURRED
WAWSE-00041 - INVALID DATE RANGE CODE

Cancel Registration Web Service

The Cancel Registration web service allows you to cancel a registration by registration ID.

Note: Admin can offer full or no refund based on the flag "issueNoRefund" .If that is set to false, Full refund amount is calculated and refunded to test taker and the registration will be cancelled.

If it is set to true, Registration will be cancelled but test taker will not get any refund

Request Parameters

These are the allowed field names (* indicates an optional field):

Cancel Registration Request Parameters

requestType	
securityToken	
returnFormat	
login : "johnsmith1" [A valid user login]	issueNoRefund: false,[Admin can offer full or no refund based on the flag]
registrationId : "99999999" [A valid registration ID]	registrationCancelReason : [Reason for registration cancellation. The field value cannot have more than 60 characters]

Calling the Cancel Registration Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "CANCEL REGISTRATION",
  "securityToken": "*****[REDACTED]*****",
  "returnFormat": "JSON",
  "login": "johnsmith1",
  "registrationId": "99999999",
  "issueNoRefund": false,
  "registrationCancelReason": "The test taker is out of town"
}
```

Example Cancel Registration Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of the "Cancel Registration" web service call in JSON.

```
{
  "candidate": {
    "addressCity": "Honolulu",
    "addressCountry": "United States",
    "addressPostalCode": "96818",
    "addressState": "Hawaii",
    "addressStreet1": "4334 Arron Smith Drive",
    "addressStreet2": [],
  }
}
```

```
"clientRegion": {
  "id": "1",
  "name": "Kryterion Default Region"
},
"email": "johnsmith1@localhost.com",
"firstName": "John1111111111",
"homePhone": [],
"id": "27896",
"lastName": "Smith2789699999999",
"login": "johnsmith1",
"userDefined1": []
},
"id": "99999999",
"notes": [],
"productCode": "PR000017",
"productId": "2974",
"progress": "SCHEDULED",
"regTransactionId": "0",
"registrationDate": "2020-09-17T09:48:05-07:00",
"status": "NORMAL",
"timeElapsedInMillis": "0",
"timeRemainingInMillis": "0"
}
```

Please Note: The payment calls made during cancellation are asynchronous, and it usually takes from 4 to 5 seconds to complete. Since you get the response immediately in an API call, most likely, you will not see the updated status. The API user can get the updated status by making a "GET REGISTRATIONS" API call within a few seconds after the first "CANCEL REGISTRATION" call.

Possible Error Messages

WAWSE-00008 - INVALID LOGIN
WAWSE-00012 - INVALID REGISTRATION ID
WAWSE-RE0014 - INVALID REG CANCEL REASON LENGTH
WAWSE-RE0015 - CANNOT CANCEL REGISTRATION

Get Products Web Service

The Get Products web service allows you to retrieve a list of all of the products which are active.

Request Parameters

These are the allowable field names (* indicates an optional field):

Get Products Request Parameters

requestType (must be set to GET PRODUCTS) ReturnFormat

SecurityToken

Calling the Get Products Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "GET_PRODUCTS",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "returnFormat": "JSON"
}
```

Example Get Products Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of a Get Products web service call in JSON.

```
[
  {
    "code": "PR000001",
    "currency": "USD",
    "deliveryType": "NOT_PROCTORED",
    "id": "3839",
    "longDesc": [],
    "name": "Salesforce.com Certified Administrator",
    "price": "0.00",
    "registrationEndDate": "9999-12-31T00:00:00-07:00",
    "registrationStartDate": "2008-06-18T00:00:00-07:00",
    "status": "ACTIVE",
    "testAids": [],
    "testWindowEndDate": "9999-12-31T00:00:00-07:00",
    "testWindowStartDate": "2008-06-18T00:00:00-07:00"
  }
]
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Get Cert Status by User Web Service

The Get Cert Status by User web service allows you to retrieve the certification status for a given user.

Request Parameters

These are the allowable field names (* indicates an optional field):

Get Cert Status by User Request Parameters

requestType (must be set to GET CERT INFO)	login
securityToken	returnFormat

Calling the Get Certification Status by User Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "GET CERT INFO",
  "securityToken": "12345678-1234-1234-1234-123456789012",
  "returnFormat": "JSON",
  "login": "johndoe"
}
```

Example Get User by I Get Certification Status by User D Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of a Get Certification by User web service call in JSON.

```
{
  "certs": [
    {
      "certificationStatus": "AT_RISK",
      "certificationType": "Certified Admin",
      "description": "Acme Certified Admin track",
      "trackName": "Certified Admin"
    },
    {
      "certificationStatus": "AT_RISK",
      "certificationType": "Certified Manager",
      "description": "Acme Certified Manager track",
      "trackName": "Certified Manager"
    }
  ],
  "user": {
    "id": "123456",
    "login": "johndoe"
  }
}
```

The following is an example of what the return value would look like when Users have no certifications

```
{
```

```
"note": "No cert status found for user",  
"user": {  
  "id": "289962",  
  "login": "gwhite_admin"  
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00002 - NULL LOGIN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Get Cert Info by Date Range Web Service

Get certification status for multiple candidates using this API. Date range is based on the update/change of the certification status.

Calling the GET CERT INFO BY DATE RANGE Web Service with JSON

Here is an example JSON object:

```
{  
  "requestType": "GET CERT INFO BY DATE RANGE",  
  "securityToken": "2k2v0lmh-9xtv-4adn-qozi-vthhgwp22ts1",  
  "startDate": "2017-10-24 00:00:00",  
  "endDate": "2017-10-27 17:00:00",  
  "returnFormat": "JSON"  
}
```

Sample JSON response for GET CERT INFO BY DATE RANGE Web Service with JSON

```
[  
  {  
    "certs": [  
      {  
        "certificationStatus": "CURRENT",  
        "certificationType": "Certified Administrator",  
        "description": "Salesforce Certified Administrator track",  
        "trackName": "Certified Administrator"  
      },  
      {  
        "certificationStatus": "CURRENT",  
        "certificationType": "Certified Programmer I"  
      }  
    ]  
  }  
]
```



```

    },
    {
      "certificationStatus": "CURRENT",
      "certificationType": "Certified Technical Architect"
    }
  ],
  "user": {
    "id": "xxxx",
    "login": "xxxxxxx"
  }
},
{
  "certs": {
    "certificationStatus": "CURRENT",
    "certificationType": "Certified Technical Architect"
  },
  "user": {
    "id": "aaaa",
    "login": "sf_aaaa_taker"
  }
}
]

```

Get Client Regions Web Service

The Get Client Regions web service allows you to retrieve a list of all the client regions related to the given security token.

Request Parameters

These are the allowable fields (* indicates an optional field):

Get Certification Status by User Request Parameters

requestType (must be set to GET CLIENT REGIONS) returnFormat
securityToken

Calling the Get Client Regions Web Service with JSON

Here is an example JSON object:

```

{
  "requestType": "GET CLIENT REGIONS",
  "securityToken": "12345678-1234-1234-1234-123456789012",

```

```
"returnFormat": "JSON",  
}
```

Example Get Client Regions Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of a Get Client Region web service call in JSON.

```
{  
  "clientRegions": [  
    {  
      "id": "1234",  
      "name": "Acme Co. North America",  
    },  
    {  
      "id": "5678",  
      "name": "Acme Co. Europe",  
    },  
    {  
      "id": "9876",  
      "name": "Acme Co. Asia",  
    }  
  ]  
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY
TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE
ERROR
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

Add Registration Web Service

The ADD REGISTRATION web service allows you to register the exams for test takers. It supports only PROCTORED (only private testing centers), NOT_PROCTORED and PAPER_BASED. Facility code is mandatory only for PROCTORED delivery type exams.

Request Parameters

These are the allowable fields (* indicates an optional field):

ADD REGISTRATION ELIGIBILITY by User Request Parameters

requestType (must be set to ADD REGISTRATION) deliveryType

securityToken
login
facilityCode(for PROCTORED type exams and not
for
PAPER_BASED and NOT_PROCTORED)
examCode

scheduledDateTime(
sendEmail
returnFormat

Calling the Add Registration Web Service with JSON

Here is an example JSON object: Proctored Exam

```
{  
  "requestType": "ADD REGISTRATION",  
  "securityToken": "12345678-1234-1234-1234-1234567890126",  
  "login": "Noel.Kendall@roche.com",  
  "facilityCode": "CTC0026",  
  "examCode": "PR001049",  
  "deliveryType": "PROCTORED",  
  "scheduledDateTime": "2018-22-2 4:145:00",  
  "sendEmail": true,  
  "returnFormat": "JSON"  
}
```

Example Add REGISTRATION Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of an ADD REGISTRATION for Proctored Exam web service call in JSON.

```
{  
  "authCode": "12345",  
  "candidate": {  
    "addressCity": "abc",  
    "addressCountry": "United States",  
    "addressPostalCode": "12345",  
    "addressState": "New Mexico",  
    "addressStreet1": "abc street",  
    "addressStreet2": "Canada",  
    "citizenship": "abc",  
    "email": "abc@yahoo.com",  
    "fax": [],  
    "firstName": "abc",  
    "homePhone": "12345",  
    "id": "7",
```

```
"lastName": "def",
"login": "abcdef",
"userDefined1": []
},
"event": {
  "id": "123",
  "created": "2017-07-21T16:37:05.367-07:00",
  "creatorId": "1",
  "modified": "2017-07-21T16:37:05.367-07:00",
  "modifierId": "1",
  "blockedOut": "false",
  "center": {
    "id": "1241",
    "created": "2007-04-13T14:03:14-07:00",
    "creatorId": "9352",
    "modified": "2017-03-07T14:06:01-07:00",
    "modifierId": "535290",
    "address": {
      "id": "14223",
      "city": "Phoenix",
      "country": "United States",
      "postalCode": "85044",
      "province": "Arizona",
      "street1": "7776 South Pointe Parkway West",
      "street2": []
    }
  },
  "availability": "ASSIGNED",
  "code": "abc",
  "contactEmail": [],
  "contactFax": [],
  "contactName": "William Wilkins",
  "contactPhone": "602.659.4660",
  "delayedScheduleHours": "0",
  "futureScheduleDays": "90",
  "kcpWhoSentTheConfirmationForm": [],
  "name": "Compatibility Testing Only",
  "notes": [],
  "parkingInfo": [],
  "timezone": "America/Phoenix",
  "type": "private"
},
"dateEnd": [],
"dateStart": [],
"emailSentFlag": "false",
"registration": {
  "id": "12345",
}
```

```
}
```

Example not successful result:

```
{  
  "candidateId": "0",  
  "certNumber": "0",  
  "certTypeId": "0",  
  "clientId": "0",  
  "clientRegionId": "0",  
  "deliveryType": "PROCTORED",  
  "errorCode": "WAWSE-RE0003 - THE LOGIN IS NOT VALID",  
  "errorMessage":  
    "com.kryterion.webassessor.webservices.exception.WebServicesException: The  
    candidate login is not a test taker or does not belong to the client region  
    provided",  
  "examCode": "PR001049",  
  "facilityCode": "CTC0026",  
  "login": "Noel.Kendall@roche.com",  
  "maxScore": "0",  
  "productId": "0",  
  "randomizeResponse": "false",  
  "registrationId": "0",  
  "requestType": "ADD REGISTRATION",  
  "returnFormat": "JSON",  
  "scaledScore": "0",  
  "scheduledDate": "2019-10-02 06:25:00",  
  "securityToken": "orwfjhb6-x6b0-wz3p-f4e5-3o36ceaiv926",  
  "sendEmail": "true",  
  "syncLoginWithEmail": "false",  
  "testingCenterId": "0",  
  "userId": "0"  
}
```

Here is an example JSON object: Proctored Exam:

```
{  
  "requestType": "ADD REGISTRATION",  
  "securityToken": "csp_QA_candidate10",  
  "login": "candidate",  
  "examCode": "PR000028",  
  "facilityCode": "",  
  "scheduledDateTime": "2020-10-2 23:45:00",  
  "sendEmail": true,  
  "deliveryType": "PROCTORED",  
  "returnFormat": "JSON"  
}
```

```
}
```

The following is an example of what the return value would look like upon the successful completion of an ADD REGISTRATION for Non-Proctored Exam web service call in JSON.

```
{  
  "candidate": {  
    "addressCity": "Tempe",  
    "addressCountry": "United States",  
    "addressPostalCode": "85282",  
    "addressState": "Arizona",  
    "addressStreet1": "1673 East Avenue",  
    "addressStreet2": [],  
    "clientRegion": {  
      "id": "1",  
      "name": "Kryterion Default Region"  
    },  
    "email": "ssaa@outlook.com",  
    "firstName": "John",  
    "homePhone": "4801112222",  
    "id": "7",  
    "lastName": "Smith",  
    "login": "candidate",  
    "userDefined1": []  
  },  
  "deliveryType": "NOT_PROCTORED",  
  "emailSent": "true",  
  "id": "7102115",  
  "notes": [],  
  "productCode": "PR000028",  
  "productId": "4835",  
  "progress": "SCHEDULED",  
  "regTransactionId": "0",  
  "registrationDate": "2019-09-03T09:48:54.646-07:00",  
  "status": "NORMAL",  
  "timeElapsedInMillis": "0",  
  "timeRemainingInMillis": "0",  
  "transaction": {  
    "id": "7006112",  
    "billingAddress": {  
      "id": "1",  
      "city": "Tempe",
```

```
"country": "United States",
"postalCode": "85282",
"province": "Arizona",
"street1": "1673 East Avenue",
"street2": []
},
"candidate": {
  "id": "7",
  "created": "2005-04-07T11:01:06-07:00",
  "creatorId": "1",
  "modified": "2019-08-30T10:17:59-07:00",
  "modifierId": "4676143",
  "clientRegion": {
    "id": "1",
    "created": "2005-04-07T11:01:05-07:00",
    "creatorId": "1",
    "modified": "2019-07-30T16:23:56-07:00",
    "modifierId": "535290", "defaultLocale": [],
    "agreement": "No Shows and Cancellations: I understand I cannot
reschedule or cancel my scheduled test session with less than seventy-
two (72) hours' notice and rescheduling or cancelling will require
contacting my test sponsor. I understand that if I do reschedule or cancel
my registration with less than 72 hours' notice, I may be required to pay
an additional fee. Special Accommodations: I understand it is my
responsibility to request any special accommodations I may require
through my test sponsor for test sponsor approval and test center
arrangement prior to scheduling my test session. If you require a special
accommodation and have not already made arrangements with your test
sponsor, please click the Cancel button below and contact your test
sponsor. I understand that some special accommodations, such as a
translator or professional reader, may require an additional fee.",
    "brand": "DEFAULT",
    "defaultBcc": [],
    "defaultCc": [],
    "defaultFrom": "dev2@kryteriononline.com",
    "disableLogin": "false",
    "enableExamEligibility": "false",
    "enableMultipleExamPrice": "false",
    "enableSelfCancel": "false",
    "enableSelfResched": "false",
    "enabledForTax": "true",
    "erv": "0",
    "name": "Kryterion Default Region",
```

```
"notes": [],
"olpClientId": [],
"path": "/kryterion1",
"paymentManagerId": "15",
"sendWelcomeEmail": "false",
"shortName": "DEFAULT",
"signature": [],
"status": "ACTIVE",
"taxCompanyCode": "KRY",
"useCertificationStatus": "false",
"useKryPaymentService": "true",
"voucherOnly": "false"
},
"email": "ssaa@outlook.com",
"firstName": "John",
"homePhone": "4801112222",
"lastName": "Smith",
"login": "candidate",
"mailAddress": {
  "id": "1",
  "city": "Tempe",
  "country": "United States",
  "postalCode": "85282",
  "province": "Arizona",
  "street1": "1673 East Avenue",
  "street2": []
},
"notes": "test senneca.bcprod@gmail.com!",
"role": {
  "id": "7",
  "description": "Candidate",
  "functionalRole": "CANDIDATE",
  "name": "Candidate"
},
"status": "ACTIVE",
"esv": "0",
"olpGuid": "c6734a55-1976-428d-98de-c9ed0c98926b",
"showAgreement": "false",
"userDefined1": []
},
"candidateFirstName": "John",
"candidateLastName": "Smith",
"candidatePhone": "4801112222",
```



```
"code": "REGISTRATION WS",  
"creatorId": "1",  
"currency": "USD",  
"modifierId": "1",  
"paymentType": "OTHER",  
"status": "ACTIVE",  
"total": "0.0",  
"transDate": "2019-09-03T09:48:54.746-07:00",  
"vendorCode": "FREE"  
}  
}
```

Here is an example JSON object: Paper Based Exam:

```
{  
  "requestType": "ADD REGISTRATION",  
  "securityToken": "csp_QA_candidate10",  
  "login": "candidate",  
  "examCode": "PR000059",  
  "facilityCode": "CTC00005",  
  "scheduledDateTime": "2021-10-2 23:45:00",  
  "sendEmail": true,  
  "deliveryType": "PAPER_BASED",  
  "returnFormat": "JSON"  
}
```

The following is an example of what the return value would look like upon the successful completion of an ADD REGISTRATION for Paper Based Exam web service call in JSON.

```
{  
  "candidate": {  
    "addressCity": "Tempe",  
    "addressCountry": "United States",  
    "addressPostalCode": "85282",  
    "addressState": "Arizona",  
    "addressStreet1": "1673 East Avenue",  
    "addressStreet2": [],  
    "clientRegion": {  
      "id": "1",  
      "name": "Kryterion Default Region"  
    },  
    "email": "ssaa@outlook.com",  
    "firstName": "John",
```

```
"homePhone": "4801112222",
"id": "7",
"lastName": "Smith",
"login": "candidate",
"userDefined1": [],
},
"deliveryType": "PAPER_BASED",
"emailSent": "true",
"id": "7102117",
"notes": [],
"productCode": "PR000059",
"productId": "10438",
"progress": "SCHEDULED",
"regTransactionId": "0",
"registrationDate": "2019-09-03T10:03:36.994-07:00",
"status": "NORMAL",
"timeElapsedInMillis": "0",
"timeRemainingInMillis": "0",
"transaction": {
  "id": "7006114",
  "billingAddress": {
    "id": "1",
    "city": "Tempe",
    "country": "United States",
    "postalCode": "85282",
    "province": "Arizona",
    "street1": "1673 East Avenue",
    "street2": []
  },
},
"candidate": {
  "id": "7",
  "created": "2005-04-07T11:01:06-07:00",
  "creatorId": "1",
  "modified": "2019-08-30T10:17:59-07:00",
  "modifierId": "4676143",
  "clientRegion": {
    "id": "1",
    "created": "2005-04-07T11:01:05-07:00",
    "creatorId": "1",
    "modified": "2019-07-30T16:23:56-07:00",
    "modifierId": "535290",
    "defaultLocale": [],
  },
},
}
```

"agreement": "No Shows and Cancellations: I understand I cannot reschedule or cancel my scheduled test session with less than seventy-two (72) hours' notice and rescheduling or cancelling will require contacting my test sponsor. I understand that if I do reschedule or cancel my registration with less than 72 hours' notice, I may be required to pay an additional fee. Special Accommodations: I understand it is my responsibility to request any special accommodations I may require through my test sponsor for test sponsor approval and test center arrangement prior to scheduling my test session. If you require a special accommodation and have not already made arrangements with your test sponsor, please click the Cancel button below and contact your test sponsor. I understand that some special accommodations, such as a translator or professional reader, may require an additional fee."

```
"brand": "DEFAULT",
"defaultBcc": [],
"defaultCc": [],
"defaultFrom": "dev2@kryteriononline.com",
"disableLogin": "false",
"enableExamEligibility": "false",
"enableMultipleExamPrice": "false",
"enableSelfCancel": "false",
"enableSelfResched": "false",
"enabledForTax": "true",
"erv": "0",
"name": "Kryterion Default Region",
"notes": [],
"olpClientId": [],
"path": "/kryterion1",
"paymentManagerId": "15",
"sendWelcomeEmail": "false",
"shortName": "DEFAULT",
"signature": [],
"status": "ACTIVE",
"taxCompanyCode": "KRY",
"useCertificationStatus": "false",
"useKryPaymentService": "true",
"voucherOnly": "false"
},
"email": "ssaa@outlook.com",
"firstName": "John",
"homePhone": "4801112222",
"lastName": "Smith",
"login": "candidate",
```

```
    "mailAddress": {
      "id": "1",
      "city": "Tempe",
      "country": "United States",
      "postalCode": "85282",
      "province": "Arizona",
      "street1": "1673 East Avenue",
      "street2": []
    },
    "notes": "test senneca.bcprod@gmail.com",
    "role": {
      "id": "7",
      "description": "Candidate",
      "functionalRole": "CANDIDATE",
      "name": "Candidate"
    },
    "status": "ACTIVE",
    "esv": "0",
    "olpGuid": "c6734a55-1976-428d-98de-c9ed0c98926b",
    "showAgreement": "false",
    "userDefined1": []
  },
  "candidateFirstName": "John",
  "candidateLastName": "Smith",
  "candidatePhone": "4801112222",
  "code": "REGISTRATION WS",
  "creatorId": "1",
  "currency": "USD",
  "modifierId": "1",
  "paymentType": "OTHER",
  "status": "ACTIVE",
  "total": "0.0",
  "transDate": "2019-09-03T10:03:36.999-07:00",
  "vendorCode": "FREE"
}
}
```

The following is an example of what the return value would look like upon the unsuccessful completion of ADD Registration web service call in JSON.

```
{
  "candidateId": "0",
```

```
"certTypeId": "0",
"clientId": "0",
"clientRegionId": "0",
"deliveryType": "PAPER_BASED",
"errorCode": "WAWSE-RE0005 -REGISTRATION ALREADY EXISTS.",
"errorMessage":
"com.kryterion.webassessor.webservices.exception.WebServicesException: Registration already exists for the same user for the same product and testing center",
"examCode": "PR000135",
"facilityCode": "CTC00005",
"login": "candidate",
"maxScore": "0",
"productId": "0",
"randomizeResponse": "false",
"registrationId": "0",
"requestType": "ADD REGISTRATION",
"returnFormat": "JSON",
"scaledScore": "0",
"scheduledDateTime": "2018-10-02 23:45:00",
"securityToken": "csp_QA_candidate10",
"sendEmail": "true",
"syncLoginWithEmail": "false",
"testingCenterId": "0",
"userId": "0"
}
```

Possible Error Messages and Codes.

WAWSE-RE0001 - FACILITY CODE SHOULD BE PRIVATE

The facility code is either unavailable or it may not belong to the client testing network

WAWSE-RE0002 - PRODUCT CODE IS INCORRECT

The Delivery type for the product should be either PROCTORED or NOT_PROCTORED or PAPER_BASED

WAWSE-RE0003 - THE LOGIN IS NOT VALID

The candidate login is not a test taker or does not belong to the client region provided

WAWSE-RE0004 - DELIVERY TYPE SHOULD NOT BE NULL

Delivery type for the product should be provided

WAWSE-RE0005 -REGISTRATION ALREADY EXISTS

Registration already exists for the same user for the same product and testing center

WAWSE-RE0006 - THE DELIVERY TYPE DOES NOT EXIST FOR THE PRODUCT

The product with the delivery type provided is not present in our system

WAWSE-RE0007 - THE DATE CODE SHOULD BE FUTURE DATE OR IS INVALID

The date should be greater than current date

WAWSE-RE0008 - THE FACILITY CODE CANNOT BE NULL

Facility code is mandatory

WAWSE-RE0009 - THE DELIVERY TYPE SHOULD BE PROCTORED, NOT_PROCTORED and PAPER_BASED Delivery type should be correct

WAWSE-RE0010 - PRODUCT CODE DOES NOT BELONG TO CLIENT REGION OR CODE IS INVALID

Either product code not available or Inactive or it may belong to a different client region

WAWSE-RE0011 - SEND EMAIL SHOULD BE EITHER TRUE OR FALSE

Send email flag should be either true or false

WAWSE-00048 - NULL PRODUCT CODE

The product code cannot be null

WAWSE-00002 - NULL LOGIN

You must supply a login name for the user

WAWSE-00000 - NULL SECURITY TOKEN

WAWSE-00001 - INVALID SECURITY TOKEN

WAWSE-00003 - NULL REQUEST TYPE

WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR

WAWSE-00044 - INVALID REQUEST TYPE

WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

WAWSE-Ex0001-PRODUCT ID NULL

WAWSE-Ex0002-CANDIDATE ID NULL

WAWSE-Ex0003-AUTH CODE NULL

WAWSE-Ex0004-STATUS NULL(ACTIVE/INACTIVE)

WAWSE-Ex0005-INVALID AUTH CODE (The authorization code provided must contain only letters, numbers, or these special characters (. @ _).

WAWSE-Ex0006-The candidate ID provided doesn't exist or belongs to a different region

WAWSE-Ex0007-The candidate ID provided is not a test taker

WAWSE-Ex0008-The product ID provided does not exist or belongs to a different region.

WAWSE-Ex0009-The certification type provided does not exist.

Add Registration with Validation Web Service

The Add Registration with Validation web service allows you to register the exams for test takers. It supports only NOT_PROCTORED and PAPER_BASED. The difference between Add Registration and Add Registration with Validation is that this request will work like UI where all pre-requisites and retake rules are taken into consideration.

Request Parameters

These are the allowable fields (* indicates an optional field):

ADD REGISTRATION WITH VALIDATION by User Request Parameters

requestType (must be set to ADD REGISTRATION)

securityToken

deliveryType
login
examCode
returnFormat

Calling the Add Registration Web Service with JSON
Here is an example JSON object: Proctored Exam

```
{  
  "requestType": "ADD REGISTRATION WITH VALIDATION",  
  "securityToken": "12345678-1234-1234-1234-1234567890126",  
  "login": "Noel.Kendall@roche.com",  
  "examCode": "PR001049",  
  "deliveryType": "PROCTORED",  
  "returnFormat": "JSON"  
}
```

Example Add REGISTRATION Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of an ADD REGISTRATION WITH VALIDATION for NOT_Proctored Exam web service call in JSON.

```
{  
  "candidate": {  
    "addressCity": "abc",  
    "addressCountry": "United States",  
    "addressPostalCode": "12345",  
    "addressState": "New Mexico",  
    "addressStreet1": "abc street",  
    "addressStreet2": "Canada",  
    "citizenship": "abc",  
    "email": "abc@yahoo.com",  
    "fax": [],  
    "firstName": "abc",  
    "homePhone": "12345",  
    "id": "7",  
    "lastName": "def",  
    "login": "abcdef",  
    "userDefined1": []  
  },  
  "deliveryType": "NOT_PROCTORED",  
  "id": "2660501",  
  "notes": []  
}
```

```
    "productCode": "PR000014",  
    "productId": "14614",  
    "progress": "SCHEDULED",  
    "regTransactionId": "0",  
    "registrationDate": "2021-03-19T10:20:03-07:00",  
    "status": "NORMAL",  
    "timeElapsedInMillis": "0",  
    "timeRemainingInMillis": "0"  
}
```

Example not successful result:

```
{  
  "candidateId": "0",  
  "certNumber": "0",  
  "certTypeId": "0",  
  "clientId": "0",  
  "clientRegionId": "0",  
  "deliveryType": "PROCTORED",  
  "errorCode": "WAWSE-RE0003 - THE LOGIN IS NOT VALID",  
  "errorMessage":  
    "com.kryterion.webassessor.webservices.exception.WebServicesException: The  
    candidate login is not a test taker or does not belong to the client region  
    provided",  
  "examCode": "PR001049",  
  "facilityCode": "CTC0026",  
  "login": "Noel.Kendall@roche.com",  
  "maxScore": "0",  
  "productId": "0",  
  "randomizeResponse": "false",  
  "registrationId": "0",  
  "requestType": "ADD REGISTRATION WITH VALIDATION",  
  "returnFormat": "JSON",  
  "scaledScore": "0",  
  "scheduledDate": "2019-10-02 06:25:00",  
  "securityToken": "orwfjhb6-x6b0-wz3p-f4e5-3o36ceaiv926",  
  "sendEmail": "true",  
  "syncLoginWithEmail": "false",  
  "testingCenterId": "0",  
  "userId": "0"  
}
```

Here is an example JSON object: NOT_Proctored Exam:

```
{  
  "requestType": "ADD REGISTRATION",
```



```
"status": "NORMAL",  
"timeElapsedInMillis": "0",  
"timeRemainingInMillis": "0"  
}
```

Possible Error Messages and Codes.

WAWSE-RE0001 - FACILITY CODE SHOULD BE PRIVATE

The facility code is either unavailable or it may not belong to the client testing network

WAWSE-RE0002 - PRODUCT CODE IS INCORRECT

The Delivery type for the product should be either PROCTORED or NOT_PROCTORED or PAPER_BASED

WAWSE-RE0003 - THE LOGIN IS NOT VALID

The candidate login is not a test taker or does not belong to the client region provided

WAWSE-RE0004 - DELIVERY TYPE SHOULD NOT BE NULL

Delivery type for the product should be provided

WAWSE-RE0005 -REGISTRATION ALREADY EXISTS

Registration already exists for the same user for the same product and testing center

WAWSE-RE0006 - THE DELIVERY TYPE DOES NOT EXIST FOR THE PRODUCT

The product with the delivery type provided is not present in our system

WAWSE-RE0007 - THE DATE CODE SHOULD BE FUTURE DATE OR IS INVALID

The date should be greater than current date

WAWSE-RE0008 - THE FACILITY CODE CANNOT BE NULL

Facility code is mandatory

WAWSE-RE0009 - THE DELIVERY TYPE SHOULD BE PROCTORED, NOT_PROCTORED and PAPER_BASED Delivery type should be correct

WAWSE-RE0010 - PRODUCT CODE DOES NOT BELONG TO CLIENT REGION OR CODE IS INVALID

Either product code not available or Inactive or it may belong to a different client region

WAWSE-RE0011 - SEND EMAIL SHOULD BE EITHER TRUE OR FALSE

Send email flag should be either true or false

WAWSE-00048 - NULL PRODUCT CODE

The product code cannot be null

WAWSE-00002 - NULL LOGIN

You must supply a login name for the user

WAWSE-00000 - NULL SECURITY TOKEN

WAWSE-00001 - INVALID SECURITY TOKEN

WAWSE-00003 - NULL REQUEST TYPE

WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR

WAWSE-00044 - INVALID REQUEST TYPE

WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED

WAWSE-Ex0001-PRODUCT ID NULL

WAWSE-Ex0002-CANDIDATE ID NULL

WAWSE-Ex0003-AUTH CODE NULL

WAWSE-Ex0004-STATUS NULL(ACTIVE/INACTIVE)

WAWSE-Ex0005-INVALID AUTH CODE (The authorization code provided must contain only letters, numbers, or these special characters (. @ _).
 WAWSE-Ex0006-The candidate ID provided doesn't exist or belongs to a different region
 WAWSE-Ex0007-The candidate ID provided is not a test taker
 WAWSE-Ex0008-The product ID provided does not exist or belongs to a different region.
 WAWSE-Ex0009-The certification type provided does not exist.
 WAWSE-00087 - PRODUCT PRE-REQUISITES NOT MET

Add Item Web Service

The Add Item web service allows you to add items into the mentioned items folders for each client. The allowed Item Type are MC, MS and TF. The allowed number of responses for TF is Response A,B and for MC and MS it is Response A,B,C,D,E,F,G,H,I,J with A and B being mandatory.

The Item folders will be created under the parent folder with today's date. The client ID must be unique across the region. The required question custom fields must be added in the request.

Request Parameters

These are the allowable fields (* indicates an optional field):

ADD ITEM Request Parameters

requestType (must be set to ADD ITEM)	responseF*
	responseG*
	responseH*
itemBank (each folder should be separated by “^”)	
clientDefinedID	correctResponse
itemType	itemPoints
itemText	randomizeResponse
responseA	itemObjective*
responseB	itemNotes*
ResponseC*	itemFeedback*
ResponseD*	customFields*
ResponseE*	returnFormat

Calling the Add Registration Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "ADD ITEM",
  "securityToken": "123-456-7891",
```

```
"itemBank": "kryterion",
"clientDefinedID": "12345",
"itemType": "MS",
"itemText": "This is the question",
"responseA": "Response A",
"responseB": "Response B",
"responseC": "Response C",
"responseD": "Response D",
"responseE": "Response E",
"responseF": "Response F",
"responseG": "Response G",
"responseH": "Response H",
"responseI": "Response I",
"responseJ": "Response J",
"correctResponse": "A,C",
"itemPoints": "1",
"randomizeResponse": "true",
"itemObjective": "objective text",
"itemNotes": "notes text",
"itemFeedback": "feedback text",
"returnFormat": "JSON",
"customFields":
  [
    {"id": "2089229", "intValue": "12345"},
    {"id": "2089231", "numberValue": "12345.34"},
    {"id": "2099445", "stringValue": "Hello"},
    {"id": "2089230", "dateValue": "YYYY-MM-DD"},
    {"id": "2089230", "optionValue": "stringValue"}
  ]
}
```

Example Add ITEM Web Service Call JSON Return Value

```
{
  "clientDefinedID": "12345",
  "clientRegion": {
    "id": "123",
    "name": "abc "
  },
  "correctResponse": "AB",
  "customFieldValue": [
    {
      "customFieldId": "2089230",
      "customFieldText": "Date Custom Field",
      "dateValue": "2019-10-02T00:00:00-07:00",
      "intValue": "0"
    }
  ]
}
```

```

    },
    {
      "customFieldId": "2089229",
      "customFieldText": "Integer Custom Field",
      "intValue": "12345"
    },
    {
      "customFieldId": "2089231",
      "customFieldText": "Numeric Custom Field",
      "intValue": "0",
      "numberValue": "12345.34"
    },
    {
      "customFieldId": "2099445",
      "customFieldText": "test2",
      "intValue": "0",
      "stringValue": "Hello"
    }
  ],
  "feedback": "feedback text",
  "id": "706057",
  "itemBank": " kryterion^1.1",
  "itemPoints": "1",
  "itemText": "This is the question",
  "itemType": "MS",
  "notes": "notes text",
  "objective": "objective text",
  "randomizeResponse": "true",
  "responseA": "Response A",
  "responseB": "Response B",
  "responseC": "Response C",
  "responseD": "Response D",
  "responseE": "Response E",
  "responseF": "Response F",
  "responseG": "Response G",
  "responseH": "Response H",
  "responseI": "Response I",
  "responseJ": "Response J"
}

```

Example not successful result:

```

{
  "candidateId": "0",
  "certTypeId": "0",
  "clientDefinedID": "IA2a-3A_1",
  "clientId": "0",
  "clientRegionId": "0",

```

```
"correctResponse": "A",
"customFields": [
  {
    "id": "466",
    "intValue": "0",
    "stringValue": "1. Recall"
  },
  {
    "id": "467",
    "intValue": "0",
    "stringValue": "Yes"
  }
],
"errorCode": "WAWSE-00067 - ITEM CLIENT ID EXISTS",
"errorMessage":
  "com.kryterion.webassessor.webservices.exception.WebServicesException
  : WAWSE00067 - ITEM CLIENT ID EXISTS",
"itemBank": "custom_field^options^1.1",
"itemFeedback": [],
"itemNotes": "notes",
"itemObjective": "objective",
"itemPoints": "4",
"itemText": "Item Text?",
"itemType": "MC",
"maxScore": "0",
"productId": "0",
"randomizeResponse": "false",
"registrationId": "0",
"requestType": "ADD ITEM",
"responseA": "Response A",
"responseB": "Response B",
"responseC": "Response C",
"responseD": "Response D",
"responseE": "E",
"responseF": "F",
"responseG": "G",
"responseH": "H",
"responseI": "I",
"responseJ": "J",
"returnFormat": "JSON",
"scaledScore": "0",
"securityToken": "csp_QA_candidate10",
"sendEmail": "false",
"syncLoginWithEmail": "false",
"testingCenterId": "0",
"userId": "0"
}
```

Possible Error Messages and codes

- WAWSE-00067 - ITEM CLIENT ID EXISTS The Client ID must be unique.
- WAWSE-00062 - INVALID ITEM TYPE The possible Item Type are MC,MS,TF
- WAWSE-00031 - MISSING REQUIRED CUSTOM FIELDS The required custom fields must be added in the request.
- WAWSE-00030 - INVALID CUSTOM FIELD ID The Custom Field ID is invalid/incorrect
- WAWSE-00063 - RESPONSE IS NULL The Response are missing
- WAWSE-00073 - NULL ITEM TEXT The Item Text is NULL
- WAWSE-00064 - CORRECT OPTION IS NULL The Correct Option is Required
- WAWSE-00065 - INVALID CORRECT OPTION The Correct Option is INVALID. It can have values A,B,C,D,E,F,G,H,I,J provided that those response are existing.
- WAWSE-00066 - INVALID ITEM OBJECTIVE LENGTH The Objective length cannot exceed 150 characters.
- WAWSE-00074 - NULL ITEM POINTS The Item points must have a value
- WAWSE-00071 - INVALID RESPONSES FOR TF ITEM TYPE The TF items must have true and false responses.
- WAWSE-00044 - INVALID REQUEST TYPE
- WAWSE-00000 - NULL SECURITY TOKEN
- WAWSE-00060 - AN UNEXPECTED ERROR OCCURREDWAWSE-00001 - INVALID SECURITY TOKEN
- WAWSE-00003 - NULL REQUEST TYPE
- WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR

Edit Item Web Service

The Edit Item web service allows you to edit items each client. The supported Item Types are Multiple Choice (MC), Multiple Select (MS) and True/False (TF). The number of responses for an Item cannot change.

The client ID must be unique across the region. The required question custom fields must be added to the request.

Request Parameters

These are the allowable fields (* indicates an optional field):

ADD ITEM Request Parameters

- | | |
|-------------------|-------------------------------|
| requestType | updateRandomizeResponse * |
| securityToken | itemNotes * |
| itemId | itemFeedback * |
| clientDefinedID * | itemObjective * |
| itemText * | gradientScoring * |
| responseA - J * | pointsGradientResponseA - J * |
| correctResponse * | customFields * |
| itemPoints * | |

Calling the Edit Item Web Service with JSON

Here is a complete example JSON object:

```
{
  "requestType": "EDIT ITEM",
  "securityToken": "ExampleSecurityToken",
  "itemId": "IIT000061",
  "clientDefinedID": "UpdatedID",
  "itemText": "edited",
  "responseA": "A response update",
  "responseB": "B response update",
  "responseC": "C response update",
  "responseD": "D response update",
  "responseE": "E response update",
  "responseF": "F response update",
  "responseG": "G response update",
  "responseH": "H response update",
  "responseI": "I response update",
  "responseJ": "J response update",
  "correctResponse": "C",
  "itemPoints": 2,
  "updateRandomizeResponse": true,
  "itemNotes": "Updated notes",
  "itemFeedback": "Updated feedback",
  "itemObjective": "Updated objective",
  "gradientScoring": true,
  "pointsGradientResponseA": 0.0,
  "pointsGradientResponseB": 1.0,
  "pointsGradientResponseC": 2.0,
  "pointsGradientResponseD": 1.5,
  "pointsGradientResponseE": 1.0,
  "pointsGradientResponseF": 0.5,
  "pointsGradientResponseG": 0.0,
  "pointsGradientResponseH": 1.1,
  "pointsGradientResponseI": 1.0,
  "pointsGradientResponseJ": 1.1,
  "customFields": [
    {"id": 1, "stringValue": "Updated"},
    {"id": 2, "numberValue": 126.9},
    {"id": 3, "dateValue": "2023-12-20"},
    {"id": 4, "intValue": 2},
    {"id": 5, "optionValue": "Option 3"}
  ]
}
```


Here is a sample JSON object for just one update:

```
{
  "requestType": "EDIT ITEM",
  "securityToken": "ExampleSecurityToken",
  "itemId": "IIT000061",
  "responseA": "A response update"
}
```

Example Edit Item Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of an Edit Item web service call in JSON.

```
<simpleQuestion>
  <clientDefinedID>IITQ32</clientDefinedID>
  <clientRegion>
    <id>16226</id>
    <name>John Brand</name>
  </clientRegion>
  <correctResponse>A</correctResponse>
  <feedback></feedback>
  <id>803831</id>
  <itemBank>IBF000026</itemBank>
  <itemPoints>1</itemPoints>
  <itemText>IITQ32</itemText>
  <itemType>TF</itemType>
  <notes></notes>
  <objective></objective>
  <randomizeResponse>false</randomizeResponse>
  <responseA>A response update</responseA>
  <responseB>false</responseB>
</simpleQuestion>
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN

WAWSE-00001 - INVALID SECURITY TOKEN

WAWSE-00003 - NULL REQUEST TYPE

WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR

WAWSE-00030 - INVALID CUSTOM FIELD ID: A custom field could not be found for the provided custom field ID

WAWSE-00031 - MISSING REQUIRED CUSTOM FIELDS: one of the custom fields is required, but is being changed to be empty

WAWSE-00044 - INVALID REQUEST TYPE

WAWSE-00062 - INVALID ITEM TYPE: The item to be edited is not one of the accepted types (MC, MS, TF), or the item type does not support the request update (boundSelection, and

- partialScoring are only used for MS, and gradientScoring is only used for MC)
- WAWSE-00064 - CORRECT OPTION IS NULL: the correct option must have a value provided when it is being changed
 - WAWSE-00065 - INVALID CORRECT OPTION: MC and TF question can only have 1 correct option, and the correct option may only be A, B, C, D, E, F, G, H, I, J, if those responses exist
 - WAWSE-00066 - ITEM CLIENT ID LENGTH EXCEEDED- MAX SIZE 50 CHARACTERS
 - WAWSE-00067 - ITEM CLIENT ID EXISTS: the provided clientDefinedID already exists, it must be unique
 - WAWSE-00068 - ITEM OBJECTIVE LENGTH EXCEEDED- MAX SIZE 150 CHARACTERS WAWSE-00073 - NULL ITEM TEXT: the item text has to have a value when included to be changed
 - WAWSE-00076 - INVALID ITEM POINTS: the item points provided are invalid, they must be 0 or a positive number
 - WAWSE-00084 - CUSTOM FIELD MAX SIZE EXCEEDED: one of the custom field values provided exceeds the max size
 - WAWSE-00089 - UNABLE TO EDIT ITEM: The item cannot be edited as it is included on an in-progress test
 - WAWSE-00090 - INVALID RESPONSE POINTS PERCENTAGE PROVIDED: one of the pointsPercentageResponse is for a response that does not exist for the item, or the value provided is invalid*
 - WAWSE-00091 - INVALID RESPONSE POINTS GRADIENT PROVIDED: one of the pointsGradientResponse is for a response that does not exist for the item, or the value provided is invalid**
 - WAWSE-00093 - UNABLE TO EDIT ITEM POINTS: points are not able to be updated for the item, due to the test being published
 - WAWSE-00094 - UNABLE TO FIND ITEM ID: An item could not be found with the provided itemId
 - WAWSE-00095 - UNABLE TO EDIT ITEM SCORING: unable to edit the items scoring (gradientScoring or partialScoring) due to the item being in a test that is in progress or has been completed
 - WAWSE-00096 - INVALID RESPONSE FOR ITEM: one of the responses to be updated does not exist for the item

Get Unassigned Vouchers by Batch Web Service

The Get Unassigned Vouchers by Batch web service allows you to retrieve a information on a voucher batch which are of type unassigned category so that you may look up the total quantity of the voucher codes and the remaining count available, voucher batch expiration date etc.,

For instance, to use the Get Unassigned Vouchers by Batch web service, you must supply voucher name, security token for the specific client region and the request type must be GET UNASSIGNED VOUCHERS BY BATCH

Request Parameters

```
{
  "requestType": "GET UNASSIGNED VOUCHERS BY BATCH",
  "securityToken": "ExampleSecurityToken",
  "voucherName": "voucher name",
  "returnFormat": "JSON"
}
```

Example Get Unassigned Vouchers by Batch Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the

successful completion of a Get Custom Fields web service call in JSON.

```
{
  "creationDate": "2024-03-27T16:10:24-07:00",
  "creator": "PraveenTumu",
  "description": [],
  "exams": "3,126,494,1354,1398,2174,70565",
  "quantity": "8",
  "remainingVouchers": "8",
  "status": "ACTIVE",
  "voucherExpirationDate": "2024-03-29T23:59:00-07:00",
  "voucherLastExamDate": "2024-03-29T23:59:00-07:00",
  "voucherName": "ptVoucher2",
  "voucherUnit": "Percentage",
  "voucherValue": "100"
}
```

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-VB0002 – THE VOUCHER NAME CANNOT BE NULL
WAWSE-VB0013 – VOUCHER BATCH DOES NOT EXIST
WAWSE-VB0015 – ONLY SUPPORTS UNASSIGNED VOUCHERS

Get Coupons Web Service

The Get Coupons web services allows a Client to obtain details about their coupons by coupon name.

Request Parameters

Here is an example JSON object:

```
{
  "requestType": "GET COUPONS",
  "securityToken": "XXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX",
  "couponCode": "samplecoupon",
  "returnFormat": "JSON"
}
```

Example Get Unassigned Vouchers by Batch Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of an API NAME web service call in JSON.

```
{
  "Coupon Code": "Coupon Code/Name",
  "Coupon Value": "Value"
  "Coupon Unit": "Percentage or Dollar"
  "Creation Date": "Coupon creation date",
}
```

```

"Total Coupon Uses": "Coupon uses"
"Creator": "Creator"
"Uses per Candidate": "Uses per candidate"
"Coupon Expiration Date": "Coupon Expiration Date"
"Status": "Coupon Status"
}

```

Possible Error Messages

```

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-CN001 - COUPON CODE IS NULL

```

Add Voucher Batch Web Service

The Add Voucher Batch web service allows you to create a new voucher batch via API.

Request Parameters

These are the allowable fields (* indicates an optional field):

ADD VOUCHER BATCH Request Parameters

```

"requestType": "ADD VOUCHER BATCH",
"securityToken": "*****[REDACTED]*****",
"voucherName": "New Voucher",
"clientName": "Salesforce",
"clientRegionName": "Salesforce",
"expirationDate": "01/05/2021",
"lastExamDate": "",*
"exams": ["PR000001", "PR000002"],
"assignment":
{"assignedVoucher": null, "unassignedAmount": 3, "basedOnTestCompletion": null, "alertAmount": null},
"description": "",*
"status": "ACTIVE",
"value": {"amount": null, "percentage": 100},
"emailDomains": [],*
"countryNames": ["United States", "Canada"],
"returnFormat": "JSON"

```

These are the allowed field names (* indicates an optional field):

```

requestType * : "ADD VOUCHER BATCH"
securityToken * : "*****[REDACTED]*****"
voucherName * : "New Voucher" [A valid string for voucher name]
clientName : "Salesforce"
clientRegionName : "Salesforce"
expirationDate * : "01/05/2021" [A valid date]
lastExamdate : "" [A valid date or empty string or NULL]

```

exams * : ["PR000001", "PR000002"] [A list of exam codes]

assignment * : {"assignedVoucher": null, "unassignedAmount": 3, "basedOnTestCompletion": null, "alertAmount": null} [only one of four options "assignedVoucher" (true/false), "unassignedAmount" (Integer), "basedOnTestCompletion" (true/false), and "alertAmount" (Integer) should be not NULL]

{"assignedVoucher": true, "unassignedAmount": null, "basedOnTestCompletion": null, "alertAmount": null} OR

{"assignedVoucher": null, "unassignedAmount": 30, "basedOnTestCompletion": null, "alertAmount": null} OR

{"assignedVoucher": null, "unassignedAmount": null, "basedOnTestCompletion": true, "alertAmount": null} OR

```
{"assignedVoucher": null, "unassignedAmount": null, "basedOnTestCompletion": null, "alertAmount": 500}
```

description : "" [A valid string or empty string]

status : "ACTIVE" [either "ACTIVE" or "INACTIVE"]

value : '{"amount": null, "percentage": 100}' [only one of options in "amount" and "percentage" should not be NULL and should have a valid value]

```
{"amount": 50, "percentage": null} OR {"amount": null, "percentage": 100}
```

emailDomains : [] [A list containing valid email domains or empty list]

countryNames : [] [An optional list containing valid country names or empty list]

returnFormat: JSON

Calling the Add Voucher Batch Web Service with JSON

An example request JSON object:

```
{
  "requestType": "ADD VOUCHER BATCH",
  "securityToken": "*****[REDACTED]*****",
  "voucherName": "New Voucher",
  "clientName": "Salesforce",
  "clientRegionName": "Salesforce",
  "expirationDate": "01/05/2021",
  "lastExamDate": "",
  "exams": ["PR000001", "PR000002"],
  "assignment":
  {"assignedVoucher": null, "unassignedAmount": 3, "basedOnTestCompletion": null, "alertAmount": null},
  "description": "",
  "status": "ACTIVE",
  "value": {"amount": null, "percentage": 100},
  "emailDomains": [],
  "countryNames": [],
  "returnFormat": "JSON"
}
```

Example Add Voucher Batch Web Service Call JSON Return Value

The following is an example of what the return value would look like upon the successful completion of the "ADD VOUCHER BATCH" web service call in JSON.

```
{
  "voucherName": "New Voucher",
  "dateCreated": "2020-10-06T21:32:46.712-07:00",
  "voucherCodes": ["AEuE2E2EuB8yr6Mi", "Ahuh2h2hutZaMiRG", "AjTj2j2jTSJUN92r"]
}
```

Please Note:

If you use assignedVoucher, basedOnTestCompletion or alertAmount as the assignment type, the response will not include the voucher codes.

Possible Error Messages

WAWSE-00000 - NULL SECURITY TOKEN
WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-00003 - NULL REQUEST TYPE
WAWSE-00044 - INVALID REQUEST TYPE
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED
WAWSE-VB0002 - THE VOUCHER NAME CANNOT BE NULL
WAWSE-VB0003 - THE EXAMS CANNOT BE NULL OR EMPTY
WAWSE-VB0004 - VOUCHER DESCRIPTION LENGTH EXCEEDED- MAX SIZE 500 CHARACTERS
WAWSE-VB0005 - THE DOMAIN NAME IS INVALID
WAWSE-VB0006 - THE CANDIDATE ASSIGNMENT IS INVALID
WAWSE-VB0007 - THE NUMBER OF ARGUMENTS IN CANDIDATE ASSIGNMENTS IS INVALID
WAWSE-VB0008- THE VOUCHER VALUE IS INVALID
WAWSE-VB0009 - THE NUMBER OF ARGUMENTS IN VALUE IS INVALID
WAWSE-VB0010 - PRODUCT CODE DOES NOT BELONG TO CLIENT REGION OR CODE IS INVALID
WAWSE-VB0011 - PROMOTION CODE EXISTS IN THE SYSTEM
WAWSE-VB0012 - THE DATE CODE SHOULD BE FUTURE DATE OR IS INVALID
WAWSE-VB0101 - INVALID COUNTRY NAME IN THE WHITELISTED COUNTRY LIST
WAWSE-VB0102 - PROBLEM OCCURRED DURING VOUCHER CREATION

Assign Voucher Batch to Candidate Users Web Service

The Assign Voucher Batch web service allows you to create voucher codes and assigns to users via API.

Request Parameters

These are the allowable fields (* indicates an optional field):

Assign Voucher Batch to Users Request Parameters

requestType (must be set to ASSIGN VOUCHER BATCH TO USERS)

securityToken

candidateLogins

voucherName

returnFormat

sendEmail * (defaulted to false)

Calling the Assign Voucher Batch Web Service with JSON

Here is an example JSON object:

```
{  
  "requestType": "ASSIGN VOUCHER BATCH TO USERS",  
  "securityToken": "19b15bbb-d85b-47c0-8f37-f65eabff1e51",  
  "voucherName": "TestVoucherBatch",  
}
```

```
"sendEmail" : true,  
"candidateLogins": [  
  "candidate",  
  "sample"  
],  
"returnFormat": "JSON"  
}
```

Example Assign Voucher Batch Web Service Call JSON Return Value

```
{  
  "users": [  
    {  
      "emailAddress": "dev1@kryteriononline.com(132454)",  
      "login": "candidate",  
      "voucherCode": "AQm22Q2QmhjPN2AQ"  
    },  
    {  
      "emailAddress": "gvepanjeri@kryteriononline.com",  
      "login": "sample",  
      "voucherCode": "AhA22h2hAVT8en3N"  
    }  
  ],  
  "voucherName": "TestVoucherBatch"  
}
```

Example not successful result

```
{  
  "candidateId": "0",  
  "candidateLogins": [  
    "julydeploy",  
    "junedeploy"  
  ],  
  "certTypeId": "0",  
  "clientId": "0",  
  "clientRegionId": "0",  
  "errorCode": "WAWSE-VB0013 - VOUCHER BATCH DOES NOT EXIST",  
  "errorMessage":  
    "com.kryterion.webassessor.webservices.exception.WebServicesException:  
    Voucher Batch does not exist",  
  "maxScore": "0",  
  "productId": "0",  
  "randomizeResponse": "false",  
}
```



```
"requestType": "ASSIGN VOUCHER BATCH TO USERS",  
"returnFormat": "JSON",  
"scaledScore": "0",  
"securityToken": "19b15bbb-d85b-47c0-8f37-f65eabff1e51",  
"sendEmail": "true",  
"syncLoginWithEmail": "false",  
"testingCenterId": "0",  
"userId": "0",  
"voucherName": "TestVoucherInvalid"  
}
```

Possible Error Messages and codes

WAWSE-00001 - INVALID SECURITY TOKEN
WAWSE-VB0013 - VOUCHER BATCH DOES NOT EXIST
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
WAWSE-VB0002 - THE VOUCHER NAME CANNOT BE NULL
WAWSE-VB0014 - INVALID CANDIDATE CREDENTIALS

Activate Deactivate Voucher Batch Web Service

It will activate/deactivate all the voucher codes under that voucher batch and makes the voucher active/inactive based on the Type of Action provided in the request parameters.

Request Parameters

These are the allowable fields (* indicates an optional field):

Activate Deactivate Voucher Batch Request Parameters

requestType (must be set to ACTIVATE DEACTIVATE VOUCHER BATCH)	
securityToken	VoucherName
	typeOfAction
returnFormat	

Calling the Activate/Deactivate Voucher batch Web Service with JSON

Here is an example JSON object:

```
{  
"voucherName": "CSME V01-10Mar08",  
"securityToken": "0dhcg632-ld99-26nc-1lo9-chsdg9m823hh",  
"typeOfAction": "ACTIVATE",  
"requestType": "ACTIVATE DEACTIVATE VOUCHER BATCH",  
}
```

```
"returnFormat": "JSON"  
}
```

Example of a Response Object:

```
[  
  "All the Voucher codes under CSME V01-10Mar08 have been successfully activated"  
]
```

Here is an example of another scenario where type of action is DEACTVAITE:

```
{  
  "voucherName": "CSME V01-10Mar08",  
  "securityToken": "0dhcg632-ld99-26nc-1lo9-chsdg9m823hh",  
  "typeOfAction": "DEACTIVATE",  
  "requestType": "ACTIVATE DEACTIVATE VOUCHER BATCH",  
  "returnFormat": "JSON"  
}
```

Example of a Response Object:

```
[  
  "All the Voucher codes under CSME V01-10Mar08 have been successfully deactivated"  
]
```

Example of Response Object When the voucher name is not provided in request parameters

```
{  
  "candidateId": "0",  
  "certTypeId": "0",  
  "clientId": "0",  
  "clientRegionId": "0",  
  "errorCode": "WAWSE-VB0002 - THE VOUCHER NAME CANNOT BE NULL",  
  "errorMessage": "com.kryterion.webassessor.webservices.exception.WebServicesException: The voucher name  
is mandatory",  
  "inactiveProducts": "false",  
  "issueNoRefund": "false",  
  "keepActive": "false",  
  "maxScore": "0",  
  "productId": "0",  
  "randomizeResponse": "false",  
  "requestType": "ACTIVATE DEACTIVATE VOUCHER BATCH",  
  "returnFormat": "JSON",  
  "scaledScore": "0",  
  "securityToken": "0dhcg632-ld99-26nc-1lo9-chsdg9m823hh",  
  "sendEmail": "false",  
  "syncLoginWithEmail": "false",  
  "testingCenterId": "0",  
  "typeOfAction": "DEACTIVATE",  
}
```

```
"userId": "0",  
"voucherName": []
```

```
}
```

Example of Response Object when Type of Action is not provided:

```
{
```

```
"candidateId": "0",  
"certTypeId": "0",  
"clientId": "0",  
"clientRegionId": "0",  
"errorCode": "WAWSE-VB0103 - THE TYPE OF ACTION NAME CANNOT BE NULL",  
"errorMessage": "com.kryterion.webassessor.webservices.exception.WebServicesException: The TYPE OF  
ACTION name is mandatory",  
"inactiveProducts": "false",  
"issueNoRefund": "false",  
"keepActive": "false",  
"maxScore": "0",  
"productId": "0",  
"randomizeResponse": "false",  
"requestType": "ACTIVATE DEACTIVATE VOUCHER BATCH",  
"returnFormat": "JSON",  
"scaledScore": "0",  
"securityToken": "0dhcg632-ld99-26nc-1lo9-chsdg9m823hh",  
"sendEmail": "false",  
"syncLoginWithEmail": "false",  
"testingCenterId": "0",  
"typeOfAction": [],  
"userId": "0",  
"voucherName": "CSME V01-10Mar08"
```

```
}
```

Example of Response Object when Type of Action is Invalid

```
{
```

```
"candidateId": "0",  
"certTypeId": "0",  
"clientId": "0",  
"clientRegionId": "0",  
"errorCode": "WAWSE-VB0104 - THE TYPE OF ACTION NAME IS INVALID",  
"errorMessage": "com.kryterion.webassessor.webservices.exception.WebServicesException: The TYPE OF  
ACTION CAN BE EITHER ACTIVATE OR DEACTIVATE",  
"inactiveProducts": "false",  
"issueNoRefund": "false",  
"keepActive": "false",  
"maxScore": "0",  
"productId": "0",  
"randomizeResponse": "false",  
"requestType": "ACTIVATE DEACTIVATE VOUCHER BATCH",
```

```
"returnFormat": "JSON",  
"scaledScore": "0",  
"securityToken": "0dhcg632-ld99-26nc-1lo9-chsdg9m823hh",  
"sendEmail": "false",  
"syncLoginWithEmail": "false",  
"testingCenterId": "0",  
"typeOfAction": "3",  
"userId": "0",  
"voucherName": "CSME V01-10Mar08"  
}
```

Example of Response Object when Voucher batch doesn't Exist:

```
{  
  "candidateId": "0",  
  "certTypeId": "0",  
  "clientId": "0",  
  "clientRegionId": "0",  
  "errorCode": "WAWSE-VB0013 - VOUCHER BATCH DOES NOT EXIST",  
  "errorMessage": "com.kryterion.webassessor.webservices.exception.WebServicesException: Voucher Batch does  
not exist",  
  "inactiveProducts": "false",  
  "issueNoRefund": "false",  
  "keepActive": "false",  
  "maxScore": "0",  
  "productId": "0",  
  "randomizeResponse": "false",  
  "requestType": "ACTIVATE DEACTIVATE VOUCHER BATCH",  
  "returnFormat": "JSON",  
  "scaledScore": "0",  
  "securityToken": "0dhcg632-ld99-26nc-1lo9-chsdg9m823hh",  
  "sendEmail": "false",  
  "syncLoginWithEmail": "false",  
  "testingCenterId": "0",  
  "typeOfAction": "DEACTIVATE",  
  "userId": "0",  
  "voucherName": "CSME V01-10Mar0"  
}
```

Example of Response object when no voucher codes are not present for the provided voucher and when type of action is DEACTIVATE:

```
[  
  "The Voucher CSME V01-10Mar09 is deactivated"  
]
```

Example of Response object when no voucher codes are not present for the provided voucher and when type of action is ACTIVATE:

```
[  
  "The Voucher CSME V01-10Mar09 is activated"
```

]

Possible Error Messages and codes

WAWSE-VB0103 - THE TYPE OF ACTION NAME CANNOT BE NULL

The TYPE OF ACTION name is mandatory

WAWSE-VB0104 - THE TYPE OF ACTION NAME IS INVALID

The TYPE OF ACTION CAN BE EITHER ACTIVATE OR DEACTIVATE

WAWSE-VB0013 - VOUCHER BATCH DOES NOT EXIST

Voucher Batch does not exist

WAWSE-VB0013 - VOUCHER BATCH DOES NOT EXIST

Voucher Batch does not exist

WAWSE-VB0002 - THE VOUCHER NAME CANNOT BE NULL

The voucher name is mandatory

Get Registration by Hash Key Web Service

The Get Registration by Hash key web service returns registration details (including transcript if applicable) when clients call this API by passing in the unique hash key.

Request Parameters

These are the allowable fields (* indicates an optional field):

Get Registration by Hash key Request Parameters

requestType (must be set to GET REGISTRATION BY HASH KEY)

securityToken

hashkey

returnFormat

Calling the Get Registration by Hash key Web Service with JSON

Here is an example JSON object:

```
{
  "requestType": "GET REGISTRATION BY HASH KEY",
  "securityToken": "s13werxy-2xum-s8c9-ofpk-5pq1ghfp7u99",
  "hashKey": "jarmcbez12rtytya",
  "returnFormat": "JSON"
}
```

Example Response 1: Get Registration by Hash key Web Service Call JSON Return response when the exam is already completed and has a transcript

```
{
  "id": "21472228",
  "date": "2020-09-02T20:25:31-07:00",
}
```

```
"maxScore": "10",
"passFail": "FAIL",
"scaledScore": "0",
"score": "1.0",
"status": "ACTIVE",
"examName": "NP random",
"simpleRegistration": {
  "candidate": {
    "addressCity": "Phoenix",
    "addressCountry": "United States",
    "addressPostalCode": "85085",
    "addressState": "Arizona",
    "addressStreet1": "123 main street",
    "addressStreet2": [],
    "clientRegion": {
      "id": "1",
      "name": "Kryterion Default Region"
    }
  },
  "email": "dev1@kryteriononline.com",
  "firstName": "John",
  "homePhone": [],
  "id": "5642412",
  "lastName": "Smith5642412",
  "login": "csp_qa_candidate",
  "userDefined1": []
},
"id": "8930201",
"notes": [],
"productCode": "PR000261",
"productId": "61333",
"progress": "COMPLETED",
"regTransactionId": "0",
"registrationDate": "2020-09-02T20:24:23-07:00",
"status": "NORMAL",
"testName": "Load_Test",
"timeElapsedInMillis": "0",
"timeRemainingInMillis": "0"
},
"testFormCode": "TF000024",
"topicScores": {
  "code": "TP000020",
  "description": "Load_Test",
  "maxScore": "10.0",
  "name": "Load_Test",
  "score": "1.0"
}
}
```

Example Response 2: Get Registration by Hash key Web Service Call JSON Return response when the exam is still to be taken and does not have a transcript

```
{
  "candidate": {
    "addressCity": "Dallas",
    "addressCountry": "United States",
    "addressPostalCode": "77056",
    "addressState": "Arizona",
    "addressStreet1": "7776 South Pointe Parkway West Suite 200",
    "addressStreet2": [],
    "clientRegion": {
      "id": "1",
      "name": "Kryterion Default Region"
    },
  },
  "email": "dev1@kryteriononline.com(122222)",
  "firstName": "John",
  "homePhone": [],
  "id": "7",
  "lastName": "Smith7",
  "login": "candidate",
  "userDefined1": []
},
"id": "9024000",
"notes": [],
"productCode": "PR000015",
"productId": "2912",
"progress": "SCHEDULED",
"regTransactionId": "0",
"registrationDate": "2021-02-19T10:49:14-07:00",
"status": "NORMAL",
"timeElapsedInMillis": "0",
"timeRemainingInMillis": "0"
}
```

Example not successful result

```
{
  "candidateId": "0",
  "certTypeId": "0",
  "clientId": "0",
  "clientRegionId": "0",
  "errorCode": "WAWSE-00001 - INVALID SECURITY TOKEN",
}
```

```

    "errorMessage":
    "com.kryterion.webassessor.webservices.exception.WebServicesException: The
    security token submitted for this request is not valid",
    "hashKey": "jarmcbez12rtytya",
    "maxScore": "0",
    "productId": "0",
    "randomizeResponse": "false",
    "requestType": "GET REGISTRATION BY HASH KEY",
    "returnFormat": "JSON",
    "scaledScore": "0",
    "securityToken": "s13werxy-2xum-s8c9-ofpk-5pq1ghfp7u99",
    "sendEmail": "false",
    "syncLoginWithEmail": "false",
    "testingCenterId": "0",
    "userId": "0"
}

```

Update Lab Progress Web Service

Now that we have only one exam/registration/transcript for the (MC + lab) exam, there is a need to create a new "UPDATE LAB PROGRESS" API that clients can call to update the same transcript with the lab exam result. Since we are using the new values in the registration "progress" and transcript "pass_fail" enums, based on the result of the lab, the transcript and registration records are updated.

Scenario	Registration "progress" enum	Transcript "pass_fail" enum	Client Sends (labprogress)
Candidate completes MC and passes (lab not started)	MC_Completed	MC_Pass	
Candidate completes MC and fails (lab not started)	Completed	Fail	
Candidate completes MC and passes (lab in progress)	Lab_inprogress	MC_pass	LAB_START
Candidate completes MC and passes and completes lab (lab results delayed)	Completed	MC_pass	LAB_COMPLETE
Candidate completes MC and completes lab (lab result pass)	Completed	Pass	LAB_PASS

Candidate completes MC and completes lab (lab result fail)	Completed	MC_Pass_Lab_fail	LAB_FAIL
--	-----------	------------------	----------

Example of Request JSON:

```
{
  "registrationId": "10715771",
  "securityToken": "0dhcg632-ld99-26nc-1lo9-chsdg9m823hh",
  "requestType": "UPDATE LAB PROGRESS",
  "labProgress": "LAB_COMPLETE",
  "sendEmail": false,
  "returnFormat": "JSON"
}
```

Example Response:

```
{
  "certNumber": "0",
  "date": "2022-11-10T11:49:10.637-07:00",
  "id": "11052221",
  "maxScore": "8",
  "passFail": "PASS",
  "scaledScore": "0",
  "score": "1.0",
  "simpleRegistration": {
    "id": "2660656",
    "productCode": "PR000035",
    "productId": "14843",
    "progress": "COMPLETED",
    "regTransactionId": "0",
    "registrationDate": "2022-10-14T10:53:35-07:00",
    "status": "NORMAL",
    "testingCenterCity": "phoenix",
    "testingCenterCountry": "United States",
    "testingCenterId": "13",
    "testingCenterName": "new private event",
    "testingCenterState": "Arizona",
    "timeElapsedInMillis": "0",
    "timeRemainingInMillis": "0",
    "timeZone": "America/Phoenix"
  },
  "status": "ACTIVE"
}
```

Possible Error Messages and codes

WAWSE-00001 - INVALID SECURITY TOKEN
 WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR
 WAWSE-00086 - HASH KEY VALUE IS UNKNOWN

Valid Country and Province Codes

Please click on the link below to download Kryterion’s Country and Province List:
<https://www.kryterion.com/kryterion-country-and-province-list/>.

* Please note: Opening this CSV file in Excel and other non-UTF-8 compliant programs may cause corrupted characters to appear.

Error Codes and Validation Rules

When a web service request encounters an error during execution, the return value will contain two fields: errorCode and errorMessage. The error message can change depending on the context, but the error code field will remain stable, allowing the calling application to test for particular errors.

For example:

If an invalid security token is passed into a web service call, the error message might be, “You must provide a valid security token,” or “This security token is invalid for this request”.

Although the error messages were different, in both instances, the error code will be “WAWSE00001 – INVALID SECURITY TOKEN” as noted in the following table.

Code	Explanation	Validation Rule
WAWSE-00000 - NULL SECURITY TOKEN	The security token provided in the request is missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00001 - INVALID SECURITY TOKEN	The security token provided in the request is invalid.	Must match the security token value stored in the database.
WAWSE-00002 - NULL LOGIN	The login field in the request is missing or zero length.	Must be non-null and have a length greater than zero.

WAWSE-00003 - NULL REQUEST TYPE	The request type field is missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00004 - NULL/INVALID FIRST NAME	The first name field is missing or zero length or more than 35-character length	Must be non-null and have a length greater than zero and less than 35 characters.
WAWSE-00005 - NULL EMAIL ADDRESS	The email field is missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00006 - NULL PASSWORD	The password field is missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00007 - NULL PASSWORD SECRET	The secret field is missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00008 - INVALID LOGIN	The login field is invalid.	The user does not exist in the database.
WAWSE-00009 - UNEXPECTED PERSISTENCE ERROR	An unexpected error occurred while accessing the database.	
WAWSE-00010 - INVALID USER ID	The user id field is invalid.	Must be a value greater than zero and be associated with a user belonging to the client region associated with the given security token.
WAWSE-00012 - INVALID REGISTRATION ID	The registration id field is invalid.	Must be a value greater than zero and associated with a registration belonging to a user in the client region associated with the given

		security token.
WAWSE-00013 – NULL/INVALID LAST NAME	The last name field is missing or zero length or more than 45 character in length	Must be non-null and have a length greater than zero and less than 45 characters
WAWSE-00014 - NULL ADDRESS STREET	The street address field is missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00015 - NULL ADDRESS CITY	The city field is missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00016 - NULL ADDRESS POSTAL CODE	The postal code field is missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00017 - NULL ADDRESS COUNTRY	The country is missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00018 - INVALID COUNTRY	The country or country code provided in the request is invalid.	The country must exist and be in the approved ISO Name format, or the ISO-3 3-character abbreviation.
WAWSE-00019 - INVALID STATE/PROVINCE	The state or state code provided is invalid.	The state must be existing and be in the approved ISO Name format or the ISO abbreviation format.
WAWSE-00020 - INVALID PASSWORD	The password provided is invalid.	The password must be at least 8 characters and less than 100 characters and contain at least one uppercase character, one lowercase character, one number, and one special character.
WAWSE-00021 - INVALID FAX	The fax number provided is invalid.	
WAWSE-00022 - INVALID PHONE	The phone number provided is invalid.	The phone number must only contain numbers, dashes, and parentheses.
WAWSE-00023 - EMAIL IN USE	The email address provided is already in use for the client region associated with the security token.	The email address must be unique for all users in the client region associated with the security token.
WAWSE-00024 - INVALID LOGIN FORMAT	The login provided is in an invalid format.	The login must be at least 6 characters long and less than 50 characters and contain only letters, numbers, the @

		symbol, underscores and periods.
WAWSE-00025 - INVALID EMAIL	The email provided is invalid.	The email address must be in proper email address format.
WAWSE-00026 - INVALID POSTAL CODE	The postal code provided is invalid.	Must be between 1 and 13 characters and contain only numbers, letters, spaces, and hyphens.
WAWSE-00027 - INVALID NAME	The first name and/or last name provided is invalid.	User names must not contain special characters other than spaces, single apostrophes, hyphens, periods, carets, and underscores.
WAWSE-00028 - LOGIN ALREADY EXISTS	The login provided already exists.	Logins must be unique for each user.
WAWSE-00029 - INVALID STREET ADDRESS	The street address provided is invalid.	Must not contain asterisks or percent signs.
WAWSE-00030 - INVALID CUSTOM FIELD ID	The custom field id provided is invalid.	The custom field id must be greater than zero and associated with a custom field assigned to the client region for the given security token.
WAWSE-00031 - MISSING REQUIRED CUSTOM FIELDS	One or more required custom fields are missing from the request.	
WAWSE-00034 - INVALID PROVINCE FOR COUNTRY	The state or state code provided is valid, but not for the country provided.	
WAWSE-00035 - INVALID ADDRESS CITY	The city provided is invalid.	The city must not contain any of the following characters: '!', '@', '#', '\$', '%', '^', '*'.
WAWSE-00037 - MISMATCH EMAIL AND LOGIN	The email address and login provided belong to different users.	
WAWSE-00038 - NULL DATE	The date provided is either missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00039 - INVALID FAX LENGTH	The fax number provided is too long.	
WAWSE-00040 - INVALID PHONE LENGTH	The phone number provided is too long.	Must be 1 to 20 characters.
WAWSE-00041 - INVALID DATE RANGE CODE	The date range provided is invalid.	The start date must be before the end date.
WAWSE-00042 - UNPARSEABLE DATE	The date provided is in an incorrect format.	Date format should be YYYY-MM-DD HH:MM:SS.

WAWSE-00043 - INVALID SECRET LENGTH	The length of the password secret provided is invalid.	Must be 12 characters or less.
WAWSE-00044 - INVALID REQUEST TYPE	The request type provided is unknown.	The request type must be a valid request type as set forth in this document.
WAWSE-00045 - ZERO REGISTRATION ID	The registration id provided is either missing or zero.	Must be a number greater than zero and associated with a user for the client region associated with the given security token.
WAWSE-00048 - NULL PRODUCT CODE	The product code field is either missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00049 - DATE RANGE TOO LONG	The date range provided is too long.	Start and end dates must not be more than 7 days apart.
WAWSE-00050 - PASS FAIL INDICATION IS NULL	The pass/fail field is either missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00051 - PASS FAIL VALUE IS UNKNOWN	The value of the pass/fail field provided is unknown.	Must be one of the following values: NA, PASS, FAIL, NOT SCORED.
WAWSE-00051 - TRANSCRIPT STATUS IS NULL	The transcript status field is either missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00052 - TRANSCRIPT STATUS IS UNKNOWN	The transcript status field value is unknown.	Must be one of the following values: ACTIVE, INACTIVE.
WAWSE-00053 - TRANSCRIPT DATE IS NULL	The transcript date field is either missing or zero length.	Must be non-null and have a length greater than zero.
WAWSE-00056 - UNABLE TO FIND TEST ID	The test id provided is not found in the database.	An incorrect state of data in the database has made saving a new transcript impossible.
WAWSE-00057 - TRANSCRIPT ALREADY EXISTS	A transcript for the given registration already exists.	
WAWSE-00058 - UNABLE TO FIND CLIENT REGION ID IN DATABASE	Can't find client region with the given id	
WAWSE-00060 - AN UNEXPECTED ERROR OCCURRED	An unexpected error occurred.	

WAWSE-00061 - UNABLE TO FIND ITEM CLIENT DEFINED ID	The Item Id is not found in database	
WAWSE-00062 - INVALID ITEM TYPE	The Item Type is invalid	The permitted Item Type is "MC" or "MS" or "TF"
WAWSE-00063 - RESPONSE IS NULL	The response A, B is required field	Must be string fields
WAWSE-00064 - CORRECT OPTION IS NULL	The correct response is blank	Must be at least one among the responses.
WAWSE-00065 - INVALID CORRECT OPTION	The correct response is incorrect	Must be at least one among the responses and should not contain any numeric and special characters except",."
WAWSE-00066 - INVALID ITEM OBJECTIVE LENGTH	The length of the objective is exceeding	Must be less than 50 characters
WAWSE-00067 - ITEM CLIENT ID EXISTS	The Item client id already exists in Database	The Item client ID must be unique.
WAWSE-00068 - INVALID CITIZENSHIP COUNTRY	The country provided is invalid. Please see documentation for valid country list.	The country provided is invalid. Please see documentation for valid country list.
WAWSE-00069 - INVALID TOPIC SCORES FOR THIS REGISTRATION	Please provide valid topic scores for this registration	Please provide valid topic scores for this registration
WAWSE-00071 - INVALID RESPONSES FOR TF ITEM TYPE	The TF item can have only 2 responses	The valid Responses are only A and B
WAWSE-00073 - NULL ITEM TEXT	The item text is blank	Must be string of any length
WAWSE-00074 - NULL ITEM POINTS	The item points is null	Must be Integer field
WAWSE-C0001 - INVALID CSME REGISTRATION CODE	Format invalid. Enter either 5 digits followed by a hyphen and 6 digits, or "+ "5 digits followed by a hyphen, the letter V (in caps) and 5 digits, or "	

	<p>+ "2 digits, followed by a capital letter (A-Z) followed by 3 digits, a hyphen "</p> <p>+ "and 6 digits or 2 digits, a capital letter followed by 3 digits, a hyphen, "</p> <p>+ "a capital letter (A-Z) followed by 5 digits (xxxxx-xxxxxx or xxxxx- Vxxxxx, "</p> <p>+ "xxAxxx-xxxxxxx, xxAxxx-Axxxxx).</p> <p>All registration codes must be 30 characters or less.</p>	
WAWSE-C0002 - NULL CSME REGISTRATION CODE	You must supply a registration code	
WAWSE-C0003 - INVALID NAME ON CERTIFICATE	The name on the certificate must be greater than 0 characters and not longer than 50 characters	
WAWSE-C0004 - INVALID EMPLOYERS NAME	The employers name must not be longer than 40 characters	
WAWSE-C0005 - INVALID CURRENT IT CERTS	The current IT certs field may not be longer than 40 characters	
WAWSE-C0006 - INVALID PREREQ CODE	The Pre-Requisite code must not be longer than 30 characters	
WAWSE-C0007 - INVALID INSTRUCTOR	The instructor name must not be longer than 30 characters	
WAWSE-C0008 - INVALID EXTERNAL CERT	The external certificate must not be longer than 40 characters	
WAWSE-A0001 - INVALID MIDDLE NAME	The middle name must not be longer than 30 characters	
WAWSE-A0002 - INVALID MEMBER NUMBER	The member number may not be longer than 30 characters	
WAWSE-A0003 - INVALID ORG NAME	The organization name may not be longer than 30 characters	
WAWSE-A0004 - INVALID SALESFORCE CONTACT ID	The Salesforce contact id may not be longer than 30 characters	
WAWSE-Ex0001	EITHER NULL PRODUCT ID OR NULL CERTTYPE ID	Either only product id or certification

WAWSE-Ex0002	NULL CANDIDATEID	type id should be provided. A candidate ID must be provided
WAWSE-Ex0003	NULL AUTH CODE	An Auth code must be provided.
WAWSE-Ex0004	NULL STATUS OR NOT A PROPER STATUS(ACTIVE/INACTIVE)	Status cannot be null and can be only either ACTIVE or INACTIVE
WAWSE-Ex0005- INVALID AUTH CODE FORMAT		The authorization code provided must contain only letters, numbers, or these special characters (. @ _
WAWSE-Ex0006	The candidate ID provided doesn't exist or belongs to a different region	Candidate Id should be present in the system or it should belong the client region provided
WAWSE-Ex0007	The candidate ID provided is not a test taker.	Candidate Id should be a test taker
WAWSE-Ex0008	The product ID provided doesn't exist or belongs to a different region	The product Id should belong to the client region provided.
WAWSE-Ex0009	The certification type provided doesn't exist.	The certification Type should be present in the system
WAWSE-RE0001 - FACILITY CODE SHOULD BE PRIVATE	The facility code is either unavailable or it may not belong to the client testing network	
WAWSE-RE0002 - PRODUCT CODE IS INCORRECT	The Delivery type for the product should be either PROCTORED or NOT_PROCTORED or PAPER_BASED	
WAWSE-RE0003 - THE LOGIN IS NOT VALID	The candidate login is not a test taker or does not belong to the client region provided	
WAWSE-RE0004 - DELIVERY TYPE SHOULD NOT BE NULL	The delivery_type for the product should be provided	
WAWSE-RE0005 REGISTRATION ALREADY EXISTS	Registration already exists for the same user for the same product and testing center	
WAWSE-RE0006 - THE DELIVERY TYPE DOES NOT	The product with the delivery type provided is not present in our system	

EXIST FOR THE PRODUCT		
WAWSE-RE0007 - THE DATE CODE SHOULD BE FUTURE DATE OR IS INVALID	The date should be greater than current date	
WAWSE-RE0008 – THE FACILITY CODE CANNOT BE NULL	Facility code is mandatory	
WAWSE-RE0009 - THE DELIVERY TYPE SHOULD BE PROCTORED or NOT_PROCTORED or PAPER_BASED	Delivery type should be correct	
WAWSE-RE0010 - PRODUCT CODE DOES NOT BELONG TO CLIENT REGION OR CODE IS INVALID	The product code is either inactive or unavailable or may belong to a different client region	
WAWSE-RE0011 - SEND EMAIL SHOULD BE EITHER TRUE OR FALSE		
WAWSE-RE0012 - Channel must be assigned to the product	The channel with the testing center is not available	
WAWSE-VB0104 - THE TYPE OF ACTION NAME IS INVALID	The TYPE OF ACTION CAN BE EITHER ACTIVATE OR DEACTIVATE	
WAWSE-VB0103 - THE TYPE OF ACTION NAME CANNOT BE NULL	The TYPE OF ACTION name is mandatory	
WAWSE-VB0013 - VOUCHER BATCH DOES NOT EXIST	Voucher Batch does not exist	
WAWSE-VB0002 - THE VOUCHER NAME CANNOT BE NULL	The voucher name is mandatory	