



# Asentia

## **API Manual**

June 2017

© 2002-2017 ICS Learning Group

## Disclaimer

ICS Learning Group makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. ICS Learning Group reserves the right to revise this publication and to make changes to its content at any time, without obligation to notify any person or entity of such revisions or changes.

Further, ICS Learning Group makes no representations or warranties with respect to any ICS Learning Group product, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. ICS Learning Group reserves the right to make changes to any and all parts of ICS Learning Group products at any time, without obligation to notify any person or entity of such changes.

## Trademarks

Asentia™ and ICS Learning Group are registered trademarks of Interactive Communications Solutions Group, Inc.

Other brands and their products are trademarks or registered trademarks of their respective holders.

## Product Support

If you have purchased a support package and have any questions during the use of Asentia™ that are not addressed in this guide, please visit our support site at: <http://support.isclearninggroup.com/>

Or contact us at:

ICS Learning Group  
8221 Ritchie Highway  
Suite 303  
Pasadena, MD 21122

Phone: 410-975-9440

Fax: 410-975-9445

<http://www.icslearninggroup.com>

## Overview

The Asentia™ API enables customers to integrate their third-party applications with the LMS. Due to the range in variation of third-party applications, support for this API is limited:

- If the Asentia™ feature, method, dataset, etc. is not listed in this document, it is not available via the API.
- Per above, reporting features/methods are not listed and, thus, not supported. We understand this is a popular request but the possibility for unfavorable system impact is too great if an improper or excessive request is made. There are other methods to obtain automated reports (i.e. automated subscriptions).
- Similar to any LMS customization requests, any actual coding assistance is not covered by support and must be scoped as a billable project at a 2-hour minimum.
- Asentia™ Support is with our direct customers. You may request us to work directly with your customer's developers, support, or representatives, but such contact will be scoped as a billable project at a 2-hour minimum.

In sum, ICS will support clarification of what the API offers access to and verbal examples of how people use the API. However, any actual coding or detailed technical assistance will require a billable project to be initiated.

## Table of Contents

Overview .....	3
Configuring API Access .....	5
API Method Reference .....	5
GetCourse.....	6
GetCourseCatalog.....	10
GetEnrollment.....	13
EnrollCourse .....	16
EnrollLearningPath.....	19
GetGroup .....	21
SaveGroup .....	24
AttachUserToGroups .....	27
RemoveUserFromGroups .....	29
GetUser .....	31
SaveUser .....	36
GetSSOToken.....	42

## Configuring API Access

Click the System administrator widget on the left side of the screen, then click “API.”

Under “API Access,” select “On.” Make up an Authentication Key with at least 10 characters (can contain letters, numbers and/or symbols, and is case sensitive), and put it in the Authentication Key box. This key is like a password for the API on your account: every API request made must include your account name and the key you enter here in order to be successfully executed. Our example in this manual will be “GoWithAsentia” moving forward.

Click “Save Changes.”

The API URL is noted under “URL.” The appropriate path for each method is noted in this document and should be combined with this URL to create the full URL for each request.

## API Method Reference

A complete list of available API methods follows.

XSD schemas are provided for all XML requests listed in this reference.

All dates and/or times in all requests should be provided in ISO date or ISO datetime format as specified in the XSD schema and normalized to GMT. Additionally, all dates and times that are included in responses will be in ISO datetime format and normalized to GMT.

Requests POSTed to the API should use the parameter “data” for the payload (request) string.

**NOTE:** Some special characters may need to be URL encoded in order to pass properly to the API interface. Whether this is the case or not depends on what data you’re passing. If you get a ‘fail’ status, check your data values and URL encode any special characters (i.e. try using “%40” instead of “@”).

In fact, using your programming language’s “URLEncode” method is often the best approach by forming the string, then using the above method to URL encode that value. However, be sure to **ONLY** encode the string itself, **NOT** the entire request payload.

## GetCourse

URL:

`/_util/API/default.aspx?op=GetCourse`

JSON Schema:

`/_util/API/Schema/GetCourse.json`

XSD Schema:

`/_util/API/Schema/GetCourse.xsd`

## Function

This method is used to retrieve the properties of a course.

## Options

You are able to return course data and properties by ID, Course Code, or Full Text Search. Note that Course Code, and Full Text Search performs a search, and may return data for more than one course.

Note that if multiple parameters are specified, the search will use the following order of precedence. If ID is specified, all other search parameters will be ignored and only the course with that ID will be returned. If Course Code is specified, Full Text Search will be ignored and only the course(s) with that Course Code will be returned. If your intent is to perform a Full Text Search, specify only that parameter.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "id": 10580,
      "code": null,
      "fts": null
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <getCourse>
```

```

        <id>10580</id>
        <code></code>
        <fts></fts>
    </getCourse>
</payload>
</request>

```

Returned sample course object in XML:

```

<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload>
    <Course>
      <Id>10580</Id>
      <IdSite>2</IdSite>
      <Title><![CDATA[Computer and Internet Introduction:]]></Title>
      <CourseCode>P001</CourseCode>
      <RevCode>F9589699GKI9BLAAHI6IC9E5JICI9DEI</RevCode>
      <Avatar />
      <Cost />
      <Credits />
      <EstimatedLengthInMinutes />
      <ShortDescription><![CDATA[Computer and Internet
Introduction]]></ShortDescription>
      <LongDescription />
      <Objectives />
      <SocialMedia />
      <IsPublished>true</IsPublished>
      <IsClosed>false</IsClosed>
      <IsLocked>false</IsLocked>
      <IsPrerequisiteAny>true</IsPrerequisiteAny>
      <IsFeedActive>true</IsFeedActive>
      <IsFeedModerated>false</IsFeedModerated>
      <IsFeedOpenSubscription>false</IsFeedOpenSubscription>
      <DtCreated>2017-01-06T20:30:45.12</DtCreated>
      <DtModified>2017-03-10T20:09:29</DtModified>
      <IsDeleted>false</IsDeleted>
      <DtDeleted />
      <Order />
      <DisallowRating>false</DisallowRating>
      <Rating />
      <Votes />
      <ForceLessonCompletionInOrder>false</ForceLessonCompletionInOrder
>
      <SelfEnrollmentDueInterval />
      <SelfEnrollmentExpiresFromStartInterval />
      <SelfEnrollmentExpiresFromFirstLaunchInterval />
    </Course>
  </payload>
</response>

```

```

<SearchTags />
<IsCallingUserEnrolled>>false</IsCallingUserEnrolled>
<LanguageSpecificProperties>
  <LangString>en-US</LangString>
  <Title><![CDATA[Computer and Internet Introduction:]]></Title>
  <ShortDescription><![CDATA[Computer and Internet
Introduction]]></ShortDescription>
  <LongDescription />
  <Objectives />
  <SearchTags />
</LanguageSpecificProperties>
<SocialMediaProperties />
</Course>
</payload>
</response>

```

Returned sample course object in JSON:

```

{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
      "Id": 10580,
      "IdSite": 2,
      "Title": "Computer and Internet Introduction:",
      "CourseCode": "P001",
      "RevCode": "F9589699GKI9BLAAHI6IC9E5JICI9DEI",
      "Avatar": null,
      "Cost": null,
      "Credits": null,
      "EstimatedLengthInMinutes": null,
      "ShortDescription": "Computer and Internet Introduction",
      "LongDescription": null,
      "Objectives": null,
      "SocialMedia": null,
      "IsPublished": true,
      "IsClosed": false,
      "IsLocked": false,
      "IsPrerequisiteAny": true,
      "IsFeedActive": true,
      "IsFeedModerated": false,
      "IsFeedOpenSubscription": false,
      "DtCreated": "2017-01-06T20:30:45.12",
      "DtModified": "2017-03-10T20:09:29",
      "IsDeleted": false,
      "DtDeleted": null,
      "Order": null,
      "DisallowRating": false,
      "Rating": null,
      "Votes": null,
    }
  }
}

```



```

    "ForceLessonCompletionInOrder": false,
    "SelfEnrollmentDueInterval": null,
    "SelfEnrollmentDueTimeframe": null,
    "SelfEnrollmentExpiresFromStartInterval": null,
    "SelfEnrollmentExpiresFromStartTimeframe": null,
    "SelfEnrollmentExpiresFromFirstLaunchInterval": null,
    "SelfEnrollmentExpiresFromFirstLaunchTimeframe": null,
    "SearchTags": "",
    "IsCallingUserEnrolled": false,
    "LanguageSpecificProperties":
    [
      {
        "LangString": "en-US",
        "Title": "Computer and Internet Introduction:",
        "ShortDescription": "Computer and Internet
Introduction",
        "LongDescription": "",
        "Objectives": "",
        "SearchTags": ""
      }
    ],
    "SocialMediaProperties": [ ]
  }
}

```

## GetCourseCatalog

URL:

`/_util/API/default.aspx?op=GetCourseCatalog`

JSON Schema:

`/_util/API/Schema/GetCourseCatalog.json`

XSD Schema:

`/_util/API/Schema/GetCourseCatalog.xsd`

### Function

This method is used to retrieve course catalog listings.

### Options

You are able to return course catalog data by ID, or Full Text Search. Note that Full Text Search performs a search, and may return data for more than one course catalog.

Note that if multiple parameters are specified, the search will use the following order of precedence. If ID is specified, all other search parameters will be ignored and only the course catalog with that ID will be returned. If your intent is to perform a Full Text Search, specify only that parameter.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "id": 123,
      "fts": null
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <getCourseCatalog>
      <id>123</id>
      <fts></fts>
    </getCourseCatalog>
  </payload>
</request>
```

```
    </getCourseCatalog>
  </payload>
</request>
```

Returned sample course object in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload>
    <Catalogs>
      <Catalog>
        <IdCatalog>123</IdCatalog>
        <IdSite>2</IdSite>
        <IdParent />
        <Title><![CDATA[Computer Essentials]]></Title>
        <Order />
        <ShortDescription><![CDATA[Basic computer
courses.]]></ShortDescription>
        <LongDescription />
        <IsPublished />
        <IsPrivate>>false</IsPrivate>
        <Cost>0</Cost>
        <CalculatedCost>0</CalculatedCost>
        <CostType>1</CostType>
        <IsClosed>>true</IsClosed>
        <DtCreated>2017-01-11T16:47:29.36</DtCreated>
        <DtModified>2017-01-11T16:47:29.36</DtModified>
        <SearchTags />
        <NumberEnrollableCourses>2</NumberEnrollableCourses>
        <LanguageSpecificProperties>
          <LangString>en-US</LangString>
          <Title><![CDATA[Computer Essentials]]></Title>
          <ShortDescription><![CDATA[Basic computer
courses.]]></ShortDescription>
          <LongDescription />
          <SearchTags />
        </LanguageSpecificProperties>
        <CourseIds>10580,10581</CourseIds>
      </Catalog>
    </Catalogs>
  </payload>
</response>
```

Returned sample course object in JSON:

```
{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
      "IdCatalog": 123,
      "IdSite": 0,
      "IdParent": null,
      "Title": "Computer Essentials",
      "Order": null,
      "ShortDescription": "Basic computer courses.",
      "LongDescription": null,
      "IsPublished": null,
      "IsPrivate": false,
      "Cost": 0.0,
      "CalculatedCost": 0.0,
      "CostType": 1,
      "IsClosed": true,
      "DtCreated": "2017-01-11T16:47:29.36",
      "DtModified": "2017-01-11T16:47:29.36",
      "SearchTags": "",
      "NumberEnrollableCourses": 2,
      "LanguageSpecificProperties": [
        {
          "LangString": "en-US",
          "Title": "Computer Essentials",
          "ShortDescription": "Basic computer courses.",
          "LongDescription": "",
          "SearchTags": "" }
      ],
      "CourseIds": "10580,10581"
    }
  }
}
```

## GetEnrollment

URL:

`/_util/API/default.aspx?op=GetEnrollment`

JSON Schema:

`/_util/API/Schema/GetEnrollment.json`

XSD Schema:

`/_util/API/Schema/GetEnrollment.xsd`

### Function

This method is used to retrieve data for a user's enrollment in a course.

### Options

You are able to return enrollment data for user course enrollments by enrollment ID, user ID, or course ID. Note that querying data by user ID or course ID returns all enrollments for that user or course.

You may only use one parameter to retrieve course enrollment data, `idEnrollment`, `idUser`, or `idCourse`.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "idEnrollment": 123456
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <getEnrollment>
      <id>123456</id>
    </getEnrollment>
  </payload>
</request>
```

Returned sample enrollment object in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload>
    <Enrollments>
      <Enrollment>
        <Id>123456</Id>
        <IdSite>2</IdSite>
        <IdCourse>0</IdCourse>
        <IdUser>152645</IdUser>
        <IdGroupEnrollment />
        <IdRuleSetEnrollment />
        <IdLearningPathEnrollment />
        <IdTimezone>15</IdTimezone>
        <IsLockedByPrerequisites>false</IsLockedByPrerequisites>
        <Code />
        <Title><![CDATA[Workplace Environmental Concerns]]></Title>
        <DtStart>2013-01-06T19:30:00</DtStart>
        <DtDue />
        <DtExpiresFromStart />
        <DtExpiresFromFirstLaunch />
        <DtEffectiveExpires />
        <DtFirstLaunch />
        <DtCompleted>2013-01-07T19:30:00</DtCompleted>
        <DtCreated>2017-01-17T11:05:10.877</DtCreated>
        <DtLastSynchronized />
        <DueInterval />
        <DueTimeframe />
        <ExpiresFromStartInterval />
        <ExpiresFromStartTimeframe />
        <ExpiresFromFirstLaunchInterval />
        <ExpiresFromFirstLaunchTimeframe />
        <IdActivityImport></IdActivityImport>
      </Enrollment>
    </Enrollments>
  </payload>
</response>
```

Returned sample enrollment object in JSON:

```
{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
      "Id": 123456,
      "IdSite": 2,
```

```
    "IdCourse": 0,  
    "IdUser": 152645,  
    "IdGroupEnrollment": null,  
    "IdRuleSetEnrollment": null,  
    "IdLearningPathEnrollment": null,  
    "IdTimezone": 15,  
    "IsLockedByPrerequisites": false,  
    "Code": null,  
    "RevCode": null,  
    "Title": "Workplace Environmental Concerns",  
    "DtStart": "2013-01-06T19:30:00",  
    "DtDue": null,  
    "DtExpiresFromStart": null,  
    "DtExpiresFromFirstLaunch": null,  
    "DtEffectiveExpires": null,  
    "DtFirstLaunch": null,  
    "DtCompleted": "2013-01-07T19:30:00",  
    "DtCreated": "2017-01-17T11:05:10.877",  
    "DtLastSynchronized": null,  
    "DueInterval": null,  
    "DueTimeframe": null,  
    "ExpiresFromStartInterval": null,  
    "ExpiresFromStartTimeframe": null,  
    "ExpiresFromFirstLaunchInterval": null,  
    "ExpiresFromFirstLaunchTimeframe": null,  
    "IdActivityImport": null  
  }  
}  
}
```

## EnrollCourse

### URL:

`/_util/API/default.aspx?op=EnrollCourse`

### JSON Schema:

`/_util/API/Schema/EnrollCourse.json`

### XSD Schema:

`/_util/API/Schema/EnrollCourse.xsd`

## Function

This method is used to enroll a user into a course.

## Properties

This method can be used to create a single course enrollment for a single user. The properties that need to be sent in the data request are below:

- **idUser (required)**  
Type: integer  
Specifies the ID of the user account that the enrollment is to be assigned to.
- **idCourse (required)**  
Type: integer  
Specifies the ID of the course that the enrollment is for.
- **isLockedByPrerequisites**  
Type: boolean  
Indicates the prerequisite enforcement of the enrollment. Enter “true” to indicate an enrollment that can be locked by course prerequisites, and “false” to indicate an unlocked enrollment.
- **dtStart (required)**  
Type: datetime  
Specifies the start date and time of the enrollment.
- **idTimezone (required)**  
Type: integer  
Specifies the ID of the timezone that the enrollment is for.
- **dueInterval**  
Type: integer  
Specifies the due date interval (refer to the following element description for more details).
- **dueTimeframe**  
Type: restricted value: (“yyy”, “m”, “ww”, or “d”) Specifies the timeframe of the due date interval. A due interval of “1” with a due timeframe of “m” indicates that the due date of the enrollment shall be exactly 1 month after the start date.
- **expiresFromStartInterval**  
Type: integer  
Specifies expires from start date interval (refer to the following element description for more details).



- expiresFromStartTimeframe**  
 Type: restricted value: ("yyyy", "m", "ww", or "d")  
 Specifies the timeframe of expires from start date interval. Expires from start date interval of "6" with expires from start date timeframe of "ww" indicates that the course enrollment shall expire exactly 6 weeks after the start date.
- expiresFromFirstLaunchInterval**  
 Type: integer  
 Specifies expires from first launch date interval (refer to the following element description for more details).
- expiresFromFirstLaunchTimeframe**  
 Type: restricted value: ("yyyy", "m", "ww", or "d")  
 Specifies the timeframe of expires from first launch date interval. Expires from first launch date interval of "6" with expires from first launch date timeframe of "ww" indicates that the course enrollment shall expire exactly 6 weeks after the user first launches it.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "idUser": 25,
      "idCourse": 10580,
      "isLockedByPrerequisites": false,
      "dtStart": "2017-01-01T09:00:00",
      "idTimezone": 15,
      "dueInterval": 2,
      "dueTimeframe": "ww",
      "expiresFromStartInterval": 1,
      "expiresFromStartTimeframe": "m",
      "expiresFromFirstLaunchInterval": 6,
      "expiresFromFirstLaunchTimeframe": "ww"
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <courseEnrollment>
      <idUser>25</idUser>
    </courseEnrollment>
  </payload>
</request>
```

```
<idCourse>10580</idCourse>
<isLockedByPrerequisites>>false</isLockedByPrerequisites>
<dtStart>2017-01-01T09:00:00</dtStart>
<idTimezone>15</idTimezone>
<dueInterval>2</dueInterval>
<dueTimeframe>ww</dueTimeframe>
<expiresFromStartInterval>1</expiresFromStartInterval>
<expiresFromStartTimeframe>m</expiresFromStartTimeframe>
<expiresFromFirstLaunchInterval>6
</expiresFromFirstLaunchInterval>
  <expiresFromFirstLaunchTimeframe>ww
</expiresFromFirstLaunchTimeframe>
  </courseEnrollment>
</payload>
</request>
```

Returned sample response in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload />
</response>
```

Returned sample response in JSON:

```
{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
    }
  }
}
```

## EnrollLearningPath

URL:

`/_util/API/default.asmx?op=EnrollLearningPath`

JSON Schema:

`/_util/API/Schema/EnrollLearningPath.json`

XSD Schema:

`/_util/API/Schema/EnrollLearningPath.xsd`

### Function

This method is used to enroll a user into a learning path.

### Properties

This method can be used to create a single learning path enrollment for a single user. The properties that need to be sent in the data request are below:

- **idUser (required)**  
Type: integer  
Specifies the ID of the user account that the enrollment is to be assigned to.
- **idLearningPath (required)**  
Type: integer  
Specifies the ID of the course that the enrollment is for.
- **dtStart (required)**  
Type: datetime  
Specifies the start date and time of the enrollment.
- **idTimezone (required)**  
Type: integer  
Specifies the ID of the timezone that the enrollment is for.
- **dueInterval**  
Type: integer  
Specifies the due date interval (refer to the following element description for more details).
- **dueTimeframe**  
Type: restricted value: ("yyy", "m", "ww", or "d") Specifies the timeframe of the due date interval. A due interval of "1" with a due timeframe of "m" indicates that the due date of the enrollment shall be exactly 1 month after the start date.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "idUser": 25,
      "idLearningPath": 1234,

```

```

        "dtStart": "2017-01-01T09:00:00",
        "idTimezone": 15,
        "dueInterval": 2,
        "dueTimeframe": "ww"
    }
}
}

```

**Input data sample in XML:**

```

<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <learningPathEnrollment>
      <idUser>25</idUser>
      <idLearningPath>1234</idLearningPath>
      <dtStart>2017-01-01T09:00:00</dtStart>
      <idTimezone>15</idTimezone>
      <dueInterval>2</dueInterval>
      <dueTimeframe>ww</dueTimeframe>
    </learningPathEnrollment>
  </payload>
</request>

```

**Returned sample response in XML:**

```

<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload />
</response>

```

**Returned sample response in JSON:**

```

{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
    }
  }
}

```

## GetGroup

URL:

`/_util/API/default.aspx?op=GetGroup`

JSON Schema:

`/_util/API/Schema/GetGroup.json`

XSD Schema:

`/_util/API/Schema/GetGroup.xsd`

### Function

This method is used to retrieve properties for a group.

### Options

You are able to return group data and properties by ID, or Full Text Search. Note that Full Text Search performs a search, and may return data for more than one group.

Note that if multiple parameters are specified, the search will use the following order of precedence. If ID is specified, all other search parameters will be ignored and only the group with that ID will be returned. If your intent is to perform a Full Text Search, specify only that parameter.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "id": 123,
      "fts": null
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <getGroup>
      <id>123</id>
    </getGroup>
  </payload>
</request>
```

```
</payload>
</request>
```

Returned sample group object in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload>
    <Groups>
      <Group>
        <Id>123</Id>
        <IdSite>2</IdSite>
        <Name><![CDATA[ICS Employees]]></Name>
        <PrimaryGroupToken />
        <ObjectGUID />
        <DistinguishedName />
        <Avatar />
        <LanguageSpecificProperties>
          <LangString>en-US</LangString>
          <Name><![CDATA[ICS Employees]]></Name>
          <ShortDescription><![CDATA[A group of ICS
employees.]]></ShortDescription>
          <LongDescription><![CDATA[]]></LongDescription>
          <SearchTags />
        </LanguageSpecificProperties>
        <IsSelfJoinAllowed>true</IsSelfJoinAllowed>
        <MembershipIsPublicized>true</MembershipIsPublicized>
        <IsFeedActive>true</IsFeedActive>
        <IsFeedModerated>true</IsFeedModerated>
        <ShortDescription><![CDATA[A group of ICS
employees.]]></ShortDescription>
        <LongDescription><![CDATA[]]></LongDescription>
        <SearchTags />
      </Group>
    </Groups>
  </payload>
</response>
```

Returned sample group object in JSON:

```
{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
      "Id": 123,
      "IdSite": 2,
      "Name": "ICS Employees",
```

```

"PrimaryGroupToken": null,
"ObjectGUID": null,
"DistinguishedName": null,
"Avatar": null,
"LanguageSpecificProperties": [
  {
    "LangString": "en-US",
    "Name": "ICS Employees",
    "ShortDescription": "A group of ICS employees.",
    "LongDescription": null,
    "SearchTags": "" },
],
"IsSelfJoinAllowed": true,
"MembershipIsPublicized": true,
"IsFeedActive": true,
"IsFeedModerated": true,
"ShortDescription": "A group of ICS employees.",
"LongDescription": null,
"SearchTags": null
}
}
}

```

## SaveGroup

URL:

[/\\_util/API/default.aspx?op=SaveGroup](#)

JSON Schema:

[/\\_util/API/Schema/SaveGroup.json](#)

XSD Schema:

[/\\_util/API/Schema/SaveGroup.xsd](#)

### Function

This method is used to create a new group or modify an existing group.

### Properties

This method can be used to create or modify a single group. The properties that need to be sent in the data request are below:

- **-id (required) (XML attribute)**  
Type: integer  
Specifies the ID of the group to be saved, use 0 if creating a new group.
- **name (required)**  
Type: integer  
Specifies the name of the group.
- **isSelfJoinAllowed (required)**  
Type: boolean  
Indicates whether users can join themselves to a group (self-joinable groups are called communities). Enter "true" or "false".
- **wallsActive (required)**  
Type: boolean  
Indicates whether the group's discussion board is active. Enter "true" or "false".
- **wallsModerated (required)**  
Type: boolean  
Indicates whether the group's discussion board is moderated. Enter "true" or "false".
- **shortDescription**  
Type: string  
A short description of the group to be displayed in search results.
- **longDescription**  
An HTML formatted description of the group to be displayed when a self-joinable group is presented in the system catalog.

Input data sample in JSON:

```
{  
  "request": {  
    "securityContext": { "key": "GoWithAsentia" },
```



```

"responseFormat": "json",
"payload": {
  "-id": 0,
  "name": [
    {
      "-lang": "en-US",
      "#text": "New Group"
    }
  ],
  "isSelfJoinAllowed": true,
  "wallIsActive": true,
  "wallIsModerated": true,
  "shortDescription": [
    {
      "-lang": "en-US",
      "#text": "Short Description"
    }
  ],
  "longDescription": [
    {
      "-lang": "en-US",
      "#text": "Long Description"
    }
  ]
}
}
}
}
}

```

Input data sample in XML:

```

<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <group id='0'>
      <name>
        <value lang='en-US'>New Group</value>
      </name>
      <isSelfJoinAllowed>true</isSelfJoinAllowed>
      <wallIsActive>true</wallIsActive>
      <wallIsModerated>true</wallIsModerated>
      <shortDescription>
        <value lang='en-US'>Short Description</value>
      </shortDescription>
      <longDescription>
        <value lang='en-US'>Long Description</value>
      </longDescription>
    </group>
  </payload>
</request>

```

```
        </longDescription>
    </group>
</payload>
</request>
```

Returned sample response in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload />
</response>
```

Returned sample response in JSON:

```
{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
    }
  }
}
```

## AttachUserToGroups

URL:

`/_util/API/default.aspx?op=AttachUserToGroups`

JSON Schema:

`/_util/API/Schema/AttachUserToGroups.json`

XSD Schema:

`/_util/API/Schema/AttachUserToGroups.xsd`

### Function

This method is used to join a single user to one or more groups.

### Properties

This method can be used to join a user to one or more groups. The properties that need to be sent in the data request are below:

- **-id (required) (XML attribute)**  
Type: integer  
Specifies the ID of the user to be joined to group(s).
- **groups (required)**  
Type: integer(s)  
A collection of one or more group IDs to join the user to.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "-id": 123456,
      "groups": {
        "group": [
          { "-id": 120 },
          { "-id": 213 }
        ]
      }
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
```

```
<account></account>
  <key>GoWithAsentia</key>
</securityContext>
<responseFormat>xml</responseFormat>
<payload>
  <user id='123456'>
    <groups>
      <group id='120' />
      <group id='213' />
    </groups>
  </user>
</payload>
</request>
```

Returned sample response in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload />
</response>
```

Returned sample response in JSON:

```
{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
    }
  }
}
```

## RemoveUserFromGroups

URL:

`/_util/API/default.aspx?op=RemoveUserFromGroups`

JSON Schema:

`/_util/API/Schema/RemoveUserFromGroups.json`

XSD Schema:

`/_util/API/Schema/RemoveUserFromGroups.xsd`

### Function

This method is used to remove a single user from one or more groups.

### Properties

This method can be used to remove a user from one or more groups. The properties that need to be sent in the data request are below:

- **-id (required) (XML attribute)**  
Type: integer  
Specifies the ID of the user to be joined to group(s).
- **groups (required)**  
Type: integer(s)  
A collection of one or more group IDs to join the user to.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "-id": 123456,
      "groups": {
        "group": [
          { "-id": 120 },
          { "-id": 213 }
        ]
      }
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
```

```
<account></account>
  <key>GoWithAsentia</key>
</securityContext>
<responseFormat>xml</responseFormat>
<payload>
  <user id='123456'>
    <groups>
      <group id='120' />
      <group id='213' />
    </groups>
  </user>
</payload>
</request>
```

Returned sample response in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload />
</response>
```

Returned sample response in JSON:

```
{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
    }
  }
}
```

## GetUser

URL:

`/_util/API/default.aspx?op=GetUser`

JSON Schema:

`/_util/API/Schema/GetUser.json`

XSD Schema:

`/_util/API/Schema/GetUser.xsd`

## Function

This method is used to retrieve the properties of a user.

## Options

You are able to return user data and properties by ID, Username, or Full Text Search. Note that Full Text Search performs a search, and may return data for more than one user.

Note that if multiple parameters are specified, the search will use the following order of precedence. If ID is specified, all other search parameters will be ignored and only the user with that ID will be returned. If Username is specified, Full Text Search will be ignored and only the user with that Username will be returned. If your intent is to perform a Full Text Search, specify only that parameter.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "id": null,
      "username": "jsmith",
      "fts": null
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <getCourse>
```

```
<id></id>
  <username>jsmith</username>
  <fts></fts>
</getCourse>
</payload>
</request>
```

Returned sample user object in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload>
    <User>
      <Id>123456</Id>
      <FirstName><![CDATA[John]]></FirstName>
      <MiddleName><![CDATA[]]></MiddleName>
      <LastName><![CDATA[Smith]]></LastName>
      <DisplayName><![CDATA[Smith, John]]></DisplayName>
      <Avatar>avatar.jpg</Avatar>
      <Email><![CDATA[john.smith@email.com]]></Email>
      <Username><![CDATA[jsmith]]></Username>
      <IdSite>2</IdSite>
      <IdTimezone>15</IdTimezone>
      <ObjectGUID />
      <ActivationGUID />
      <DistinguishedName />
      <IsActive>true</IsActive>
      <MustChangePassword>>false</MustChangePassword>
      <DtCreated>2013-09-30T17:51:18.083</DtCreated>
      <DtExpires />
      <IsDeleted>>false</IsDeleted>
      <DtDeleted>2014-10-11T11:30:41.903</DtDeleted>
      <IdLanguage>57</IdLanguage>
      <LanguageString>en-US</LanguageString>
      <DtModified>2014-08-07T20:10:16.76</DtModified>
      <DtLastLogin />
      <EmployeeId />
      <Company><![CDATA[ICS Learning Group]]></Company>
      <Address><![CDATA[8221 Ritchie Hwy. Suite 303]]></Address>
      <City><![CDATA[Pasadena]]></City>
      <Province><![CDATA[Maryland]]></Province>
      <PostalCode>21122</PostalCode>
      <Country><![CDATA[USA]]></Country>
      <PhonePrimary>4109759440</PhonePrimary>
      <PhoneWork />
      <PhoneFax />
      <PhoneHome />
```



```

    <PhoneMobile />
    <PhonePager />
    <PhoneOther />
    <Department />
    <Division />
    <Region />
    <JobTitle />
    <JobClass />
    <Gender>M</Gender>
    <Race />
    <DtDOB>1983-07-26T00:00:00</DtDOB>
    <DtHire />
    <DtTerm />
    <Field00 />
    <Field01 />
    <Field02 />
    <Field03 />
    <Field04 />
    <Field05 />
    <Field06 />
    <Field07 />
    <Field08 />
    <Field09 />
    <Field10 />
    <Field11 />
    <Field12 />
    <Field13 />
    <Field14 />
    <Field15 />
    <Field16 />
    <Field17 />
    <Field18 />
    <Field19 />
    <IsUserACourseExpert>false</IsUserACourseExpert>
    <IsUserASupervisor>false</IsUserASupervisor>
    <IsUserAWallModerator>false</IsUserAWallModerator>
    <IsUserAnILTInstructor>false</IsUserAnILTInstructor>
    <GroupMemberships>
      <int>198</int>
    </GroupMemberships>
  </User>
</payload>
</response>

```

Returned sample user object in JSON:

```

{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {

```

```
"Id": 123456,
"FirstName": "John",
"MiddleName": null,
"LastName": "Smith",
"DisplayName": "Smith, John ",
"Avatar": "avatar.jpg",
"Email": "john.smith@email.com",
"Username": "jsmith",
"Password": null,
"IdSite": 2,
"IdTimezone": 15,
"ObjectGUID": null,
"ActivationGUID": null,
"DistinguishedName": "",
"IsActive": true,
"MustChangePassword": false,
"DtCreated": "2013-09-30T17:51:18.083",
"DtExpires": null,
"IsDeleted": false,
"DtDeleted": "2014-10-11T11:30:41.903",
"IdLanguage": 57,
"LanguageString": "en-US",
"DtModified": "2014-08-07T20:10:16.76",
"DtLastLogin": null,
"EmployeeId": null,
"Company": "ICS Learning Group",
"Address": "8221 Ritchie Hwy. Suite 303",
"City": "Pasadena",
"Province": "Maryland",
"PostalCode": "21122",
"Country": "USA",
"PhonePrimary": "4109759440",
"PhoneWork": null,
"PhoneFax": null,
"PhoneHome": null,
"PhoneMobile": null,
"PhonePager": null,
"PhoneOther": null,
"Department": null,
"Division": null,
"Region": null,
"JobTitle": null,
"JobClass": null,
"Gender": "M",
"Race": null,
"DtDOB": "1983-07-26T00:00:00",
"DtHire": null,
"DtTerm": null,
"Field00": null,
"Field01": null,
```

```
    "Field02": null,  
    "Field03": null,  
    "Field04": null,  
    "Field05": null,  
    "Field06": null,  
    "Field07": null,  
    "Field08": null,  
    "Field09": null,  
    "Field10": null,  
    "Field11": null,  
    "Field12": null,  
    "Field13": null,  
    "Field14": null,  
    "Field15": null,  
    "Field16": null,  
    "Field17": null,  
    "Field18": null,  
    "Field19": null,  
    "IsUserACourseExpert": false,  
    "IsUserASupervisor": false,  
    "IsUserAWallModerator": false,  
    "IsUserAnILTInstructor": false,  
    "GroupMemberships": [198]  
  }  
}
```

## SaveUser

### URL:

`/_util/API/default.aspx?op=SaveUser`

### JSON Schema:

`/_util/API/Schema/SaveUser.json`

### XSD Schema:

`/_util/API/Schema/SaveUser.xsd`

## Function

This method is used to create a new user or modify an existing user.

## Properties

This method can be used to create or modify a single user. The properties that need to be sent in the data request are below:

- **-id (required) (XML attribute)**  
Type: integer  
Specifies the ID of the user to be saved, use 0 if creating a new user.
- **firstName (required)**  
Type: string  
Specifies the first name of the user.
- **middleName**  
Type: string  
Specifies the middle name of the user.
- **lastName (required)**  
Type: string  
Specifies the last name of the user.
- **email**  
Type: string  
Specifies the user's email address.
- **username (required) (unique)**  
Type: string  
Specifies the username of the user in the system. This is the log in name the user will use to access the system and must be unique.
- **password**  
Type: string  
Specifies the password the user will use to log into the system. Only required if creating a new user. This may be a temporary value you use to set the user up and then require them to change upon log in. See "mustChangePassword" below.
- **mustChangePassword (required)**  
Type: boolean  
Determines whether a user must change their password upon their next log in to the system. Enter "true" or "false".

- **idTimezone (required)**  
Type: integer  
Specifies the ID of the user's default time zone.
- **languageString (required)**  
Type: string  
Specifies the user's default language.
- **dtExpires**  
Type: datetime  
Specifies the date/time when the user's account expires.
- **isActive (required)**  
Type: boolean  
Determines whether the user is an active user. Enter "true" or "false".
- **company**  
Type: string  
Specifies the user's company.
- **address**  
Type: string  
Specifies the user's address.
- **city**  
Type: string  
Specifies the user's city.
- **province**  
Type: string  
Specifies the user's state/province.
- **postalCode**  
Type: string  
Specifies the user's postal code.
- **country**  
Type: string  
Specifies the user's country.
- **phonePrimary**  
Type: string  
Specifies the user's primary phone number.
- **phoneHome**  
Type: string  
Specifies the user's home phone number.
- **phoneWork**  
Type: string  
Specifies the user's work phone number.
- **phoneFax**  
Type: string  
Specifies the user's fax phone number.
- **phoneMobile**  
Type: string  
Specifies the user's mobile phone number.

- **phonePager**  
Type: string  
Specifies the user's pager number.
- **phoneOther**  
Type: string  
Specifies the user's other phone number.
- **employeeID**  
Type: string  
Specifies the user's employee ID.
- **jobTitle**  
Type: string  
Specifies the user's job title.
- **jobClass**  
Type: string  
Specifies the user's job class.
- **division**  
Type: string  
Specifies the user's division.
- **region**  
Type: string  
Specifies the user's region.
- **department**  
Type: string  
Specifies the user's department.
- **dtHire**  
Type: datetime  
Specifies the date the user was hired.
- **dtTerm**  
Type: datetime  
Specifies the date the user was terminated.
- **gender**  
Type: string  
Specifies the user's gender. Value's entered should be "m" or "f".
- **race**  
Type: string  
Specifies the user's race.
- **dtDOB**  
Type: datetime  
Specifies the user's date of birth.
- **field00**  
Type: string  
Multipurpose field00.
- **field01**  
Type: string  
Multipurpose field01.

- **field02**  
Type: string  
Multipurpose field02.
- **field03**  
Type: string  
Multipurpose field03.
- **field04**  
Type: string  
Multipurpose field04.
- **field05**  
Type: string  
Multipurpose field05.
- **field06**  
Type: string  
Multipurpose field06.
- **field07**  
Type: string  
Multipurpose field07.
- **field08**  
Type: string  
Multipurpose field08.
- **field09**  
Type: string  
Multipurpose field09.
- **field10**  
Type: string  
Multipurpose field10.
- **field11**  
Type: string  
Multipurpose field11.
- **field12**  
Type: string  
Multipurpose field12.
- **field13**  
Type: string  
Multipurpose field13.
- **field14**  
Type: string  
Multipurpose field14.
- **field15**  
Type: string  
Multipurpose field15.
- **field16**  
Type: string  
Multipurpose field16.

- **field17**  
Type: string  
Multipurpose field17.
- **field18**  
Type: string  
Multipurpose field18.
- **field19**  
Type: string  
Multipurpose field19.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "-id": 0,
      "firstName": "John",
      "lastName": "Smith",
      "username": "jsmith",
      "password": "#45ent144cc0unt!",
      "mustChangePassword": true,
      "idTimezone": 15,
      "languageString": "en-US",
      "isActive": true
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <user id ="0">
      <firstName>John</firstName>
      <middleName></middleName>
      <lastName>Smith</lastName>
      <email>john.smith@email.com</email>
      <username>johnsmith</username>
      <password>#45ent144cc0unt!</password>
      <mustChangePassword>true</mustChangePassword>
      <idTimezone>15</idTimezone>
    </user>
  </payload>
</request>
```



```
<languageString>en-US</languageString>
<dtExpires>2018-01-01T09:00:00</dtExpires>
<isActive>true</isActive>
<company>ICS Learning Group</company>
<province>Maryland</province>
<postalcode>21122</postalcode>
<country>USA</country>
<phonePrimary></phonePrimary>
<phoneHome></phoneHome>
<phoneWork>410-975-9440</phoneWork>
<phoneFax></phoneFax>
<gender>m</gender>
</user>
</payload>
</request>
```

Returned sample response in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload />
</response>
```

Returned sample response in JSON:

```
{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
    }
  }
}
```

## GetSSOToken

URL:

`/_util/API/default.aspx?op=GetSSOToken`

JSON Schema:

`/_util/API/Schema/GetSSOToken.json`

XSD Schema:

`/_util/API/Schema/GetSSOToken.xsd`

### Function

This method is used to retrieve a single sign-on (SSO) token for a user. The token that is generated allows the calling application to redirect the user to the single sign-on URL

[https://\[hostname\].asentialms.com/\\_util/DoSSO.aspx?token=\[token\]](https://[hostname].asentialms.com/_util/DoSSO.aspx?token=[token]) using the token as authentication for the user. Once re-directed, the user will be logged into the system.

Note that generated tokens expire after 60 seconds. So the redirect to the single sign-on URL needs to happen within 60 seconds after token generation.

Input data sample in JSON:

```
{
  "request": {
    "securityContext": { "key": "GoWithAsentia" },
    "responseFormat": "json",
    "payload": {
      "id": 123456,
    }
  }
}
```

Input data sample in XML:

```
<request>
  <securityContext>
    <account></account>
    <key>GoWithAsentia</key>
  </securityContext>
  <responseFormat>xml</responseFormat>
  <payload>
    <getSSOToken>
      <id>123456</id>
    </getSSOToken>
  </payload>
</request>
```

Returned sample SSO token object in XML:

```
<response>
  <status>
    <message>Success</message>
    <description></description>
  </status>
  <payload>
    <SSOToken>
      <Token>7TREGNGFT78956TOI54G756T</Token>
    </SSOToken>
  </payload>
</response>
```

Returned sample SSO token object in JSON:

```
{
  "response": {
    "status": { "message": "Success", "description": "" },
    "payload": {
      "Token": "7TREGNGFT78956TOI54G756T",
    }
  }
}
```