ConcourseConnect

User API

http://www.concursive.com
Revised August 12, 2014

Overview

The ConcourseConnect User API includes RESTful methods intended for 3rd party applications to authenticate as a guest or registered user and interact with aspects of the ConcourseConnect back-end. The API is full-featured and allows users to register and manage their user accounts all through the API without the need to use the web site. Registrations do grant the user access to both the API and ConcourseConnect web site.

API use requires a developer key obtained from the site owner. Most applications will require further back-end configuration. Content configuration and management is mostly done via the web site except for specific end-user capabilities.

For example, the API is initially used in guest mode so that the user can view guest-level data. Most site's allow user registration via the API. Once registered and logged in, the registered user can interact with available data exposed through the services.

There are two forms of security:

1. SSL - SSL is the most basic way to keep communication secure; SSL must be used when talking to the API

2. Digest authentication - The user's username and password do not need to be stored on the device as plaintext, instead the information is hashed either for storage or before transmission

The API is accessed by appending /api to the ConcourseConnect URL.
http://www.example.com/api

When authentication fails, the web server returns error 401: This request requires HTTP authentication () or often a status of 1 in the response instead of 0.

Underlying data has object-level access permissions so not all data returned will be the same for each user.

The services are defined and enabled in Connect's back-end api/core_services.xml configuration file on the server.

Authorization

Every request must include the client's assigned app id. This is done either by adding an additional request header, or by appending key=XYZ (for testing) to the URL. The client platform (ios|android) must also be supplied so that the backend push notifications scheme can be configured correctly.

```
xhr.setRequestHeader('X-Concursive-Key', 'key=' + authKey);
xhr.setRequestHeader('X-Concursive-Platform', 'ios|android');
xhr.setRequestHeader('X-Concursive-Model', 'iphone5,etc.');
```

Response Notes

The response uses JSON or XML. Specify format=json or format=xml (the default)

Authentication Notes

Digest authentication requires that the authorization credentials are specified in the HTTP Request Header. See http://en.wikipedia.org/wiki/Digest_access_authentication

```
Set Authorization to (all on one line with a space after Digest):

Digest
username="username",realm="realm",nonce="nonce",uri="uri",response="respon
se"
```

Where the placeholders in quotes are substituted with the following values:

```
realm = "ConcourseConnect";
uri = "/api/site";
ha1 = md5HexDigest(username + ":" + realm + ":" + password);
ha2 = md5HexDigest("POST:" + uri);
nonce = new Date().getTime();
response = md5HexDigest(ha1 + ":" + nonce + ":" + ha2);
```

The client can store the ha1 value and the actual username, the password must not be stored. The client will use various URIs depending on the service being accessed. The “site” service above returns information about the site. Use POST/GET for the method as required by the api services.

The server may inspect nonce attributes submitted by clients, to prevent replay attacks

Typical Use Cases

Guest authentication can be used until a valid response has been obtained using user credentials. At that point the user information is stored by the client for future use. If the API ever returns an error, it may be possible that the user information is no longer valid and the user should try logging in again.

The first time the app starts up, the app must access the main site service (to get user data, image and permissions) and then also optionally retrieves some dynamic content to show the user. At some point the app may request push notification access and then that information is sent to the server too.

GET /api/site?format=json&deviceId=X&sandbox=true

GET /api/wiki?format=json&profile=main-profile&subject=Mobile+Participation

Display the ‘html’ value of the result set

The user fills out a form and registers

POST /api/registration

The user fills out the login form, the authentication is tested by getting the site and user information:

GET /api/site?format=json

An example of the app retrieving a list of records… like available topics…

GET /api/profileList?format=json&category=Topics

REST API Methods

GET /albumList

Returns a list of photo albums that the user has access to. If the requested profile does not have any albums, then the default albums are created and returned.

URL

http://social.preview.concursive.com/api/albumList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

- **profile**
  Show the albums for a specific profile
- **category**
  Show the albums for a specific category name
- **view**
  mine or public
- **hasFiles**
  Only return albums that have files

Example request and response

/albumList?items=1

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="albumList" total="13">
      <record name="album">
        <albumId>488</albumId>
        <title>Activities</title>
        <uniqueId>showing-off-features</uniqueId>
        <profile>Showing off features</profile>
        <profileLink>http://social.preview.concursive.com/show/showing-
off-features</profileLink>
        <photoCount>0</photoCount>
        <imageUrl>http://social.preview.concursive.com/image/
2008051912-64-194-75x75-0a725de3ae6d31237492297760-760000000-2/
Sample.png</imageUrl>
        <imageWidth>75</imageWidth>
        <imageHeight>75</imageHeight>
        <createdDate>1324040695317</createdDate>
        <createdText>3 weeks ago</createdText>
        <postPhotos>true</postPhotos>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /blog/:blog-id

Returns the specified blog.

URL

http://social.preview.concursive.com/api/blog/:blog-id

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- **id**
  The record id of the blog to display

Optional

none

Example request and response

/blog/11

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="blogList" total="1">
      <record name="blog">
        <id>11</id>
        <subject>Vieques, Puerto Rico</subject>
        <intro>&lt;p&gt;Curious if anyone has been here... sounds
fascinating.&lt;/p&gt;&#13;
```

```
&lt;p&gt;&lt;img src="http://social.preview.concursive.com/show/matt-rajkowski/blog-image/vieques.jpg" alt="vieques.jpg" width="513" height="383" /&gt;&lt;/p&gt;</intro>
        <message/>
        <publishDate>1276739400000</publishDate>
        <publishText>1 year ago</publishText>
        <publishedBy>Matt Rajkowski</publishedBy>
        <publishedByUniqueId>matt-rajkowski</publishedByUniqueId>
        <publishedByImageUrl>http://social.preview.concursive.com/image/2008090514-1-171-45x45-0a725de3ae6d31237492297760-760000000-2/matt-headshot.png</publishedByImageUrl>
        <publishedByImageWidth>45</publishedByImageWidth>
        <publishedByImageHeight>45</publishedByImageHeight>
        <ratingAverage>0.0</ratingAverage>
        <ratingCount>0</ratingCount>
        <ratingValue>0</ratingValue>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /blogList

Returns a list of blog posts.

URL

http://social.preview.concursive.com/api/blogList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

- **profile**
  Limit the records to the specified profile's unique id
- **category**
  Limit the records to the specified category
- **view**
  Specify **mine** to receive the user's blog posts
- **items**
- **page**

Example request and response

/blogList?items=1

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="blogList" total="4">
      <record name="blog">
        <id>11</id>
        <uniqueId>matt-rajkowski</uniqueId>
        <subject>Vieques, Puerto Rico</subject>
        <publishDate>1276739400000</publishDate>
        <publishText>1 year ago</publishText>
        <publishedBy>Matt Rajkowski</publishedBy>
        <publishedByUniqueId>matt-rajkowski</publishedByUniqueId>
        <publishedByImageUrl>http://social.preview.concursive.com/image/2008090514-1-171-45x45-0a725de3ae6d31237492297760-760000000-2/matt-headshot.png</publishedByImageUrl>
        <publishedByImageWidth>45</publishedByImageWidth>
        <publishedByImageHeight>45</publishedByImageHeight>
        <ratingAverage>0.0</ratingAverage>
        <ratingCount>0</ratingCount>
        <ratingValue>0</ratingValue>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /categoryList

Returns a list of valid categories for the site.

URL

http://social.preview.concursive.com/api/categoryList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

- **category**
  Show the subcategories of a category; otherwise just the top-level categories are displayed.

Example request and response

/categoryList

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="9" name="profileCategoryList" total="9">
      <record name="profileCategory">
        <id>63</id>
        <description>Challenges</description>
        <label>Challenges</label>
      </record>
      <record name="profileCategory">
        <id>62</id>
        <description>Deals</description>
        <label>Deals</label>
      </record>
      <record name="profileCategory">
        <id>64</id>
        <description>Meet-Ups</description>
        <label>Meet-Ups</label>
      </record>
      <record name="profileCategory">
        <id>36</id>
        <description>Organizations</description>
        <label>Organizations</label>
      </record>
      <record name="profileCategory">
        <id>37</id>
        <description>Groups</description>
        <label>Groups</label>
      </record>
      <record name="profileCategory">
        <id>47</id>
        <description>People</description>
        <label>People</label>
      </record>
      <record name="profileCategory">
        <id>48</id>
        <description>Places</description>
        <label>Places</label>
      </record>
      <record name="profileCategory">
        <id>49</id>
        <description>Events</description>
        <label>Events</label>
      </record>
      <record name="profileCategory">
        <id>51</id>
        <description>Ideas</description>
        <label>Ideas</label>
      </record>
    </recordSet>
  </response>
</concursive>
```

POST /checkin

Checks the user into the specified profile.

URL

http://social.preview.concursive.com/api/checkin

Supported formats

xml, json

Supported request methods

POST

Requires authentication

true

Parameters

Required

- **profile**
- **latitude**
- **longitude**

Optional

- accuracy
- altitude
- heading
- speed
- message
- foursquareToken
- <image attachments>

GET /checkinList

Returns the checkin points, over time

URL

http://social.preview.concursive.com/api/checkinList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- **profile**

Optional

- minTeamMemberLevel
- startDate
- endDate

GET /classified/:classified-id

Returns the specified classified ad.

URL

http://social.preview.concursive.com/api/classified/:classified-id

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- **id**

Optional

none

Example request and response

/classified/9

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="classifiedList" total="1">
      <record name="classified">
        <id>9</id>
        <enteredBy>Matt Rajkowski</enteredBy>
        <enteredByUniqueId>matt-rajkowski</enteredByUniqueId>
        <enteredByImageUrl>http://social.preview.concursive.com/image/
2008090514-1-139-45x45/DSC01277.jpg</enteredByImageUrl>
```

```
        <enteredByImageWidth>45</enteredByImageWidth>
        <enteredByImageHeight>45</enteredByImageHeight>
        <title>Have an old car you want to get rid of?</title>
        <description>We're looking for scrap steel. Let us know what you
have so we can make you an offer.</description>
        <categoryId>107</categoryId>
        <category>Services</category>
        <ratingAverage>0.0</ratingAverage>
        <ratingCount>0</ratingCount>
        <ratingValue>0</ratingValue>
        <expirationDate>1295813375172</expirationDate>
        <expirationText>1 month from now</expirationText>
        <active>true</active>
        <expired>false</expired>
        <upcoming>false</upcoming>
        <hasFiles>false</hasFiles>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /classifiedList

Returns a list of classified ads.

URL

http://social.preview.concursive.com/api/classifiedList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

none

Example request and response

/classifiedList

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="classifiedList" total="1">
      <record name="classifiedAd">
        <classifiedId>9</classifiedId>
        <projectId>107</projectId>
        <description>Have an old car you want to get rid of?</description>
      </record>
    </recordSet>
```

```
     </response>
   </concursive>
```

GET /event/:event-id

Returns the specified event.

URL

http://social.preview.concursive.com/api/event/:event-id

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- **id**

Optional

none

Example request and response

/event/1

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="eventList" total="1">
      <record name="event">
        <id>1</id>
        <title>Tidewater Woodworkers Guild Meeting</title>
        <location>Woodcraft Store, Janaf Shopping Center</location>
        <description/>
        <startDate>1237849200000</startDate>
        <endDate>1237813200000</endDate>
        <startDateText>2 years ago</startDateText>
        <endDateText>2 years ago</endDateText>
        <isWebMeeting>false</isWebMeeting>
        <isWebcast>false</isWebcast>
        <profile>Tidewater Woodworkers Guild Meeting</profile>
        <profileImageUrl>http://social.preview.concursive.com/image/
2008051912-49-11-45x45-0a725de3ae6d31237492297760-760000000-2/Icons-
Event-100x100.png</profileImageUrl>
        <profileImageWidth>45</profileImageWidth>
        <profileImageHeight>45</profileImageHeight>
        <createdBy>Tom Manos</createdBy>
        <createdByUniqueId>tom-manos</createdByUniqueId>
        <createdByImageUrl>http://social.preview.concursive.com/image/
2008090514-6-60-45x45-0a725de3ae6d31237492297760-760000000-2/captain.jpg</
createdByImageUrl>
        <createdByImageWidth>45</createdByImageWidth>
        <createdByImageHeight>45</createdByImageHeight>
      </record>
    </recordSet>
  </response>
</concursive>
```

POST /eventImage

Stores or replaces an event's image

URL

http://social.preview.concursive.com/api/eventImage

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- eventId
- <image attachment>

Optional

none

GET /eventList

Returns a list of upcoming events.

URL

http://social.preview.concursive.com/api/eventList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

- **category**
- **profile**
- **view**
  mine

Example request and response

/eventList?items=1

```
  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

POST /idea

Submits an idea (comment), linked to a topic

URL

http://social.preview.concursive.com/api/idea

Supported formats

xml, json

Supported request methods

POST

Requires authentication

true

Parameters

Required

- **profile**
- **title**

Optional

- shortDescription

POST /issue

Submits a new issue/ticket

URL

http://social.preview.concursive.com/api/issue

Supported formats

xml, json

Supported request methods

POST

Requires authentication

true

Parameters

Required

- **problem**
  The text of the issue being submitted

- **profile**
  The profile to store the issue in

Optional

none

Example request and response

/issue

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
  </response>
</concursive>
```

GET /memberList

Returns a list of team members of a profile

URL

http://social.preview.concursive.com/api/memberList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- **profile**
  The profile to return the members list for

Optional

- **view**
  The default is **approved**; specify **all** to include the pending members

Example request and response

/memberList?profile=spring-kick-off

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="2" name="memberList" total="2">
      <record name="member">
        <id>2012</id>
        <role>Member</role>
        <isAdded>true</isAdded>
        <uniqueId>luke-smith</uniqueId>
        <name>Luke Smith</name>
        <location>Southington CT 06489</location>
        <imageUrl>https://www.ctihub.com/image/2008090514-8100-1108-45x45-4e0343a0486ff1329334667607-607000000-324/snapshot.jpg</imageUrl>
        <imageWidth>45</imageWidth>
        <imageHeight>45</imageHeight>
      </record>
      <record name="member">
        <id>2092</id>
        <role>Admin</role>
        <isAdded>true</isAdded>
        <uniqueId>matt-rajkowski</uniqueId>
        <name>Matt Rajkowski</name>
        <location>Norfolk VA 23510</location>
        <imageUrl>https://www.ctihub.com/image/2008090514-45-333-45x45-4e0343a0486ff1329334667607-607000000-324/matt-headshot.png</imageUrl>
        <imageWidth>45</imageWidth>
        <imageHeight>45</imageHeight>
      </record>
    </recordSet>
  </response>
</concursive>
```

POST /notification

Submits a notification message and sends a push notification to registered iOS and Android devices. The app client must ask the user through the mobile SDK and retrieve the device id. The device id is always sent to the ‘site’ service when logging the user in.

URL

http://social.preview.concursive.com/api/notification

Supported formats

xml, json

Supported request methods

POST

Requires authentication

true

Parameters

Required

- **profile**
  The profile’s unique id to store the message under; the default is the client’s main-profile

- **title**
  The album to create if albumId is not specified

Optional

- **type**
  default

- pushNotification
  set to “true” to send as a push notification to subscribed devices

- sandbox
  set to “true” to send push notification to non-production devices

POST /photo

Posts a photo to the specified album.

URL

http://social.preview.concursive.com/api/photo

Supported formats

xml, json

Supported request methods

POST

Requires authentication

true

Parameters

Required

- ***files***
  Uploaded files must be .jpg, .jpeg, .png, .gif or .bmp

- **albumId**
  The album to store the photo in

- **albumName**
  The album to create if albumId is not specified

- **body**
  The caption of the photo

Optional

- **profile**
  The profile's unique id to create the album under; the default is the current user's profile

Example request and response

/photo

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
  </response>
</concursive>
```

GET /photoList

Returns a list of photo records.

URL

http://social.preview.concursive.com/api/photoList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

One (1) of the following is required:

- **profile**
  Limit the list of photos to a specific profile

- **category**
  Limit the list of photos to a specify category

- **albumId**
  Limit the list of photos to a specific album

Optional

- **view**
  Specify **mine** to get the user's photos; specify **connected** to get the photos for the user's connections; specify **public** to get photos that are publicly accessible; specify **friends** to get photos that me and my friends have created for public and connected profiles I have access to

- **items**

  Set the value to the number of records to return, the max value is 200.

- **page**

  Set the value to the page number to return when paging through records.

Example request and response

/photoList?items=1

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="photoList" total="32">
      <record name="photo">
        <photoId>200</photoId>
        <caption>samuel.jpg</caption>
        <clientFileName>samuel.jpg</clientFileName>
        <uniqueId>jair-domingues</uniqueId>
        <profile>jair domingues</profile>
        <profileLink>http://social.preview.concursive.com/show/jair-domingues</profileLink>
        <profileImageUrl>http://social.preview.concursive.com/image/2008090514-234-201-45x45-0a725de3ae6d31237492297760-760000000-2/100_0881.JPG</profileImageUrl>
        <profileImageWidth>45</profileImageWidth>
        <profileImageHeight>45</profileImageHeight>
        <imageUrl>http://social.preview.concursive.com/image/702101604-234-200-75x75-0a725de3ae6d31237492297760-760000000-2/samuel.jpg</imageUrl>
        <largeImageUrl>http://social.preview.concursive.com/image/702101604-234-200-640x480-0a725de3ae6d31237492297760-760000000-2/samuel.jpg</largeImageUrl>
        <createdDate>1324314223647</createdDate>
        <createdText>3 weeks ago</createdText>
        <createdBy>174</createdBy>
        <contactText>jair domingues</contactText>
        <contactUniqueId>jair-domingues</contactUniqueId>
        <contactUrl>http://social.preview.concursive.com/show/jair-domingues</contactUrl>
        <contactImageUrl>http://social.preview.concursive.com/image/2008090514-234-201-75x75-0a725de3ae6d31237492297760-760000000-2/100_0881.JPG</contactImageUrl>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /profile

GET /profileForm

POST /profileForm

POST /profileBannerImage

POST /profileImage

POST /profileInvitation

GET /profileList

Returns a list of profiles matching any specified criteria. This service is limited to 200 records per request and supports record paging.

URL

http://social.preview.concursive.com/api/profileList?view=mine&query=X&location=X&category=X&latitude=X&longitude=X&radius=X&fixed=true

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

- **fields**

  Specify the field names as a comma-separated list
- **view**

  Set the value to **mine** to return only the profiles the user is connected to, otherwise any profile the user has access to will be returned (except People).
- **query**

  A plain-text string to query the profiles with. The query engine uses stemming to find like terms, so “story” and “stories” in the profile will be returned if only “store” is specified.
- **location**

  A plain-text string to limit the results to the profile’s location. Can be a city, state, zip code and country.
- **category**

  A plain-text string to limit the category of profiles returned. For example **people**, **groups**, **businesses**
- **latitude**

  Find records near the given latitude. Also requires longitude to be set. If this value is set then only records with a location will be returned.
- **longitude**

  Find records near the given longitude. Also requires latitude to be set. If this value is set then only records with a location will be returned.
- **radius**

  Find profiles (except People) within the radius specified in miles. Requires latitude and longitude to be set. The results will include the distance to the profile when latitude and longitude are specified.

- **fixed**

  Set the value to **true** to disable radius expansion when no profiles are found. By default the radius will expand incrementally by 5 miles up to 100 miles.

- **items**

  Set the value to the number of records to return, the max value is 200.

- **page**

  Set the value to the page number to return when paging through records.

- **sort**

  Set the value to **new** to return the newly added profiles first; set the value to **highest_rated** to return the highest rated profiles first; set the value to **most_reviewed** to return the most reviewed profiles first; set the value to **alpha** to return the results in alphabetical order; the default sort order is “best match”

Example request and response

http://social.preview.concursive.com/api/profileList?view=mine&latitude=40.7&longitude=-85.8&radius=1000&category=events

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="2" name="profileList" total="2">
      <record name="profile">
        <id>117</id>
        <category>Events</category>
        <categoryLabel>Events</categoryLabel>
        <subcategory/>
        <uniqueId>macon-georgias-international-cherry-blossom-festival</
uniqueId>
        <title>Macon Georgia's International Cherry Blossom Festival</
title>
        <location>Macon GA 31201</location>
        <longitude>-83.6029116</longitude>
        <latitude>32.7918963</latitude>
        <distance>559.6896828600834</distance>
        <distanceKm>900.7332329727781</distanceKm>
        <description>Enjoy gracious southern hospitality as we celebrate
in grand style the magnificent beauty of our 300,000 flowering Yoshino
cherry trees.</description>
        <imageUrl>http://social.preview.concursive.com/image/
2008051912-49-11-45x45-0a725de3ae6d31237492297760-760000000-2/Icons-
Event-100x100.png</imageUrl>
```

```
        <imageWidth>45</imageWidth>
        <imageHeight>45</imageHeight>
        <image100Url>http://social.preview.concursive.com/image/
2008051912-49-11-100x100-0a725de3ae6d31237492297760-760000000-2/Icons-
Event-100x100.png</image100Url>
        <image200Url>http://social.preview.concursive.com/image/
2008051912-49-11-200x200-0a725de3ae6d31237492297760-760000000-2/Icons-
Event-100x100.png</image200Url>
        <webPage>http://www.cherryblossom.com/</webPage>
        <approved/>
        <closed/>
        <guests>1</guests>
        <participants/>
        <membership>0</membership>
        <ratingCount/>
        <ratingValue/>
        <ratingAverage/>
        <isActiveMember>true</isActiveMember>
        <isPendingMember>false</isPendingMember>
        <canJoin>false</canJoin>
        <canRequestToJoin>false</canRequestToJoin>
        <canLeave>true</canLeave>
        <memberCount>1</memberCount>
      </record>
      <record name="profile">
        <id>24</id>
        <category>Events</category>
        <categoryLabel>Events</categoryLabel>
        <subcategory/>
        <uniqueId>early-bird-kayak-paddle</uniqueId>
        <title>Early Bird Kayak Paddle</title>
        <location>Virginia Beach VA 23456</location>
        <longitude>-75.924558</longitude>
        <latitude>36.691451</latitude>
        <distance>599.7171671581813</distance>
        <distanceKm>965.1512246630161</distanceKm>
        <description>This paddle is for those early birds who like to see
the birds or that just want a cooler morning paddle! $16 per person.</
description>
        <imageUrl>http://social.preview.concursive.com/image/
2008090514-24-62-45x45-0a725de3ae6d31237492297760-760000000-2/fal.jpg</
imageUrl>
        <imageWidth>45</imageWidth>
        <imageHeight>45</imageHeight>
        <image100Url>http://social.preview.concursive.com/image/
2008090514-24-62-100x100-0a725de3ae6d31237492297760-760000000-2/fal.jpg</
image100Url>
        <image200Url>http://social.preview.concursive.com/image/
2008090514-24-62-200x200-0a725de3ae6d31237492297760-760000000-2/fal.jpg</
image200Url>
        <webPage>http://www.dcr.virginia.gov/state_parks/fal.shtml</
webPage>
        <approved/>
        <closed/>
```

```
           <guests>1</guests>
           <participants/>
           <membership>0</membership>
           <ratingCount/>
           <ratingValue/>
           <ratingAverage/>
           <isActiveMember>true</isActiveMember>
           <isPendingMember>false</isPendingMember>
           <canJoin>false</canJoin>
           <canRequestToJoin>false</canRequestToJoin>
           <canLeave>true</canLeave>
           <memberCount>6</memberCount>
         </record>
       </recordSet>
     </response>
   </concursive>
```

POST /profileManagement

Profile membership settings

POST /profileMembership

Action to join or leave a profile

URL

http://social.preview.concursive.com/api/profileMembership

Supported formats

xml, json

Supported request methods

POST

Requires authentication

true

Parameters

Required

- **uniqueId**
  The profile to join or leave

- **action**
  join, leave

Optional

none

Example request and response

/profileMembership

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
  </response>
</concursive>
```

POST /profileRequest

Approve another user’s request to become a friend or member

URL

http://social.preview.concursive.com/api/profileRequest

Supported formats

xml, json

Supported request methods

POST

Requires authentication

true

Parameters

Required

- **id**
  The unique team member id

- **approve**
  true, false

Optional

none

Example request and response

/profileRequest

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
  </response>
</concursive>
```

GET /profileRequestList

Returns a user’s requests to become friends.

URL

http://social.preview.concursive.com/api/profileRequestList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

none

Example request and response

/profileRequestList

```
  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

GET /promotion/:promotion-id

Returns the specified promotion.

URL

http://social.preview.concursive.com/api/promotion/:promotion-id

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- **id**

Optional

none

Example request and response

/promotion/4

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="promotionList" total="1">
      <record name="promotion">
        <id>4</id>
        <enteredBy>Matt Rajkowski</enteredBy>
        <enteredByUniqueId>matt-rajkowski</enteredByUniqueId>
        <enteredByImageUrl>http://social.preview.concursive.com/image/2008090514-1-171-45x45-0a725de3ae6d31237492297760-760000000-2/matt-headshot.png</enteredByImageUrl>
        <enteredByImageWidth>45</enteredByImageWidth>
        <enteredByImageHeight>45</enteredByImageHeight>
        <heading>Sale Items</heading>
        <briefDescription1>There are 30 items for sale</briefDescription1>
        <briefDescription2>Today only!</briefDescription2>
        <content>When these items are gone, so is our sale! Hurry in.</content>
        <ratingAverage>0.0</ratingAverage>
        <ratingCount>0</ratingCount>
        <ratingValue>0</ratingValue>
        <webPage/>
        <destinationUrl/>
        <expirationDate>1326405600000</expirationDate>
        <expirationText>7 hours from now</expirationText>
        <active>true</active>
        <expired>false</expired>
        <upcoming>false</upcoming>
        <uniqueId>steelorbis</uniqueId>
        <profile>SteelOrbis</profile>
        <profileImageUrl>http://social.preview.concursive.com/image/2008051912-36-6-45x45-0a725de3ae6d31237492297760-760000000-2/Icons-Organization-100x100.png</profileImageUrl>
        <profileImageWidth>45</profileImageWidth>
        <profileImageHeight>45</profileImageHeight>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /promotionList

Returns a list of current promotions.

URL

http://social.preview.concursive.com/api/promotionList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

- **query**
- **location**
- **category**
- **latitude**
- **longitude**
- **radius**

Example request and response

/promotionList

```
  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="promotionList" total="1">
      <record name="promotion">
        <id>4</id>
        <heading>Sale Items</heading>
        <expirationDate>1326405600000</expirationDate>
        <expirationText>7 hours from now</expirationText>
        <active>true</active>
        <expired>false</expired>
        <upcoming>false</upcoming>
        <uniqueId>steelorbis</uniqueId>
        <profile>SteelOrbis</profile>
        <profileImageUrl>http://social.preview.concursive.com/image/
2008051912-36-6-45x45-0a725de3ae6d31237492297760-760000000-2/Icons-
Organization-100x100.png</profileImageUrl>
        <profileImageWidth>45</profileImageWidth>
        <profileImageHeight>45</profileImageHeight>
      </record>
    </recordSet>
  </response>
</concursive>
```

POST /registration

Approve another user’s request to become a friend or member

URL

http://social.preview.concursive.com/api/registration

Supported formats

xml, json

Supported request methods

POST

Requires authentication

true

Parameters

Required

- **firstName**
- **lastName**
- **email**
- **password**

Optional

none

Example request and response

/registration

```
 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

POST /review

GET /reviewList

GET /settings

POST /settings

Saves the user’s settings

POST /sharePhoto

Shares a photo to Facebook

GET /show/:unique-id

Returns information about a specific profile specified by the unique-id value. The user must have access to the data.

URL

http://social.preview.concursive.com/api/show/:unique-id

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- **uniqueId**

  the text id of the desired profile

  http://social.preview.concursive.com/api/show?format=xml&uniqueId=main-profile

Optional

none

Example request and response

/show?uniqueId=main-profile

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="profileList" total="1">
      <record name="profile">
        <id>2</id>
        <category>Groups</category>
        <categoryLabel>Groups</categoryLabel>
        <subcategory/>
        <uniqueId>main-profile</uniqueId>
        <title>Weekend Wanderings</title>
        <location/>
        <description>This is an example social network based on the theme
of things people do in their spare time (weekends). This a broad theme and
will allow for a wide range of content and topics. Create a group to
organize others with similar interests, get help getting started or
developing your hobby from those already in the know.</description>
        <imageUrl>http://social.preview.concursive.com/image/
2008090514-2-82-45x45-0a725de3ae6d31237492297760-760000000-2/
marriott.jpg.jpeg</imageUrl>
        <imageWidth>45</imageWidth>
        <imageHeight>45</imageHeight>
        <image100Url>http://social.preview.concursive.com/image/
2008090514-2-82-100x100-0a725de3ae6d31237492297760-760000000-2/
marriott.jpg.jpeg</image100Url>
        <image200Url>http://social.preview.concursive.com/image/
2008090514-2-82-200x200-0a725de3ae6d31237492297760-760000000-2/
marriott.jpg.jpeg</image200Url>
        <webPage/>
        <approved>true</approved>
        <closed>false</closed>
        <guests>true</guests>
        <participants>true</participants>
        <membership>false</membership>
        <ratingCount>6</ratingCount>
        <ratingValue>26</ratingValue>
        <ratingAverage>26</ratingAverage>
        <isActiveMember>true</isActiveMember>
        <isPendingMember>false</isPendingMember>
        <canJoin>false</canJoin>
        <canRequestToJoin>false</canRequestToJoin>
        <canLeave>false</canLeave>
        <memberCount>74</memberCount>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /site

Returns information about the site and the current user.

URL

http://social.preview.concursive.com/api/site

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

- deviceId (the iOS or Android device id); supplying this value along with the client's platform (ios|android) in the request header will register this device for push notifications

- sandbox (true/false)

Example request and response

/site?format=xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1">
      <record name="site">
        <title>Weekend Wanderings</title>
        <logoUrl>http://social.preview.concursive.com/image/
2009030915-0-52-300x100/logo.png</logoUrl>
        <description>This is an example social network based on the theme
of things people do in their spare time (weekends). This a broad theme and
will allow for a wide range of content and topics. Create a group to
organize others with similar interests, get help getting started or
developing your hobby from those already in the know.</description>
        <theme>bookman</theme>
        <colorScheme>dark_blue</colorScheme>
        <userName>Matt Rajkowski</userName>
        <firstName>Matt</firstName>
        <lastName>Rajkowski</lastName>
        <userUniqueId>matt-rajkowski</userUniqueId>
        <userImageUrl>http://social.preview.concursive.com/image/
2008090514-1-171-45x45-0a725de3ae6d31237492297760-760000000-2/matt-
headshot.png</userImageUrl>
        <userImageWidth>45</userImageWidth>
        <userImageHeight>45</userImageHeight>
        <userLanguage>en_US</userLanguage>
        <isGuest>false</isGuest>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /status

Returns the activity for a specific activity stream entry, including comments.

URL

http://social.preview.concursive.com/api/status?statusId=448

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- **statusId**
  A valid activity stream statusId

Optional

none

Example request and response

/status?statusId=448

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="2" name="statusService" total="2">
      <record name="status">
        <id>448</id>
        <uniqueId>flex</uniqueId>
        <message>&lt;p&gt;&lt;a class="wikiLink external" href="/show/
matt-rajkowski"&gt;Matt Rajkowski&lt;/a&gt;  @&lt;a class="wikiLink
```

```
external" href="/show/flex"&gt;Flex开发小组&lt;/a&gt;: Hi, I'm just
reviewing language encodings and thought I'd try this group.&lt;/p&gt;
</message>
        <messageText>Matt Rajkowski  @Flex开发小组: Hi, I'm just reviewing
language encodings and thought I'd try this group.</messageText>
        <linkStartDate>1268771853604</linkStartDate>
        <linkStartText>1 year ago</linkStartText>
        <relativeDate>1268771888171</relativeDate>
        <relativeText>1 year ago</relativeText>
        <commentCount>1</commentCount>
        <profileLink>http://social.preview.concursive.com/show/flex</
profileLink>
        <type>user-entry</type>
        <typeId>98</typeId>
        <profile>Flex开发小组</profile>
        <profileImageUrl>http://social.preview.concursive.com/image/
2008051912-37-7-45x45-0a725de3ae6d31237492297760-760000000-2/Icons-
Group-100x100.png</profileImageUrl>
        <profileImageWidth>45</profileImageWidth>
        <profileImageHeight>45</profileImageHeight>
        <user>Matt Rajkowski</user>
        <imageUrl>http://social.preview.concursive.com/image/
2008090514-1-171-45x45-0a725de3ae6d31237492297760-760000000-2/matt-
headshot.png</imageUrl>
        <imageWidth>45</imageWidth>
        <imageHeight>45</imageHeight>
      </record>
      <record name="statusReply">
        <id>449</id>
        <uniqueId>flex</uniqueId>
        <message>&lt;p&gt;&lt;a class="wikiLink external" href="/show/
matt-rajkowski"&gt;Matt Rajkowski&lt;/a&gt; @&lt;a class="wikiLink
external" href="/show/flex"&gt;Flex开发小组&lt;/a&gt;: I can see that the
name of the group doesn't update properly on save, but it looks ok on
refresh.&lt;/p&gt;
</message>
        <messageText>Matt Rajkowski @Flex开发小组: I can see that the name
of the group doesn't update properly on save, but it looks ok on
refresh.</messageText>
        <linkStartDate>1268771888171</linkStartDate>
        <linkStartText>1 year ago</linkStartText>
        <relativeDate>1268771888171</relativeDate>
        <relativeText>1 year ago</relativeText>
        <profile>Flex开发小组</profile>
        <user>Matt Rajkowski</user>
        <imageUrl>http://social.preview.concursive.com/image/
2008090514-1-171-45x45-0a725de3ae6d31237492297760-760000000-2/matt-
headshot.png</imageUrl>
        <imageWidth>45</imageWidth>
        <imageHeight>45</imageHeight>
      </record>
```

```
       </recordSet>
     </response>
   </concursive>
```

POST /status

Creates a new top-level activity stream entry for the indicated profile or for the current user, and optionally creates a comment for an existing activity stream entry.

URL

http://social.preview.concursive.com/api/status

Supported formats

xml, json

Supported request methods

POST

Requires authentication

true

Parameters

Required

- **body**
  The content to be written in the activity stream

Optional

- **replyParentId**
  The statusId being replied to
- **profile**
  The associated profile unique-id to post the update in
- **latitude**
  The user's latitude
- **longitude**
  The user's longitude
- ***files***
  Binary image data for posting an image with the status update

- **albumId**
  The albumId to store the image in

- **albumName**
  The name of a new album that will be created for storing the image in

Example request and response

/status?format=json

```
[
{
"status":{"code":0,"name":"statusService","count":1,"total":1},
"objectList":
[{
"recordNumber":1,
"recordName":"status",
"status":"ok",
"statusId":"403"
}]
}
]
```

GET /statusList

Returns the user's activity stream.

URL

http://social.preview.concursive.com/api/statusList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

- **category**
  Limit the records to a specific category

- **profile**
  Limit the records to a specific profile

- **view**
  Specify **mine** to get just the user's stream; specify **friends** to get the activity stream of the user's related profiles; specify **profiles** to get the publicly accessible activity stream of non-people profiles; specify **chatter** to get the site's global activity stream; the default will show all activities of all of the user's related profiles

- ~~**latitude**~~
  Find records near the given latitude. Also requires longitude to be set. If this value is set then only records with a location will be returned.

- **~~longitude~~**
  Find records near the given longitude. Also requires latitude to be set. If this value is set then only records with a location will be returned.

- **~~radius~~**
  Find activity within the radius specified in miles. Requires latitude and longitude to be set. The results will include the distance to the post when latitude and longitude are specified.

- **items**
  Set the value to the number of records to return, the default is 25, the max value is 200.

- **page**
  Set the value to the page number to return when paging through records.

Example request and response

/statusList?items=2

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="2" name="statusList" total="103">
      <record name="status">
        <id>885</id>
        <uniqueId>kun-zhang</uniqueId>
        <message>&lt;p&gt;&lt;a class="wikiLink external" href="/show/kun-
zhang"&gt;kun zhang&lt;/a&gt;: i am here&lt;/p&gt;
</message>
        <messageText>kun zhang: i am here</messageText>
        <linkStartDate>1326263226201</linkStartDate>
        <linkStartText>1 day ago</linkStartText>
        <relativeDate>1326289362042</relativeDate>
        <relativeText>1 day ago</relativeText>
        <commentCount>1</commentCount>
        <type>user-entry</type>
        <typeId>239</typeId>
        <profile>kun zhang</profile>
        <profileImageUrl>http://social.preview.concursive.com/image/
2008051912-47-8-45x45-0a725de3ae6d31237492297760-760000000-2/Icons-
People-100x100.png</profileImageUrl>
        <profileImageWidth>45</profileImageWidth>
        <profileImageHeight>45</profileImageHeight>
        <user>kun zhang</user>
        <imageUrl>http://social.preview.concursive.com/image/
2008051912-47-8-45x45-0a725de3ae6d31237492297760-760000000-2/Icons-
People-100x100.png</imageUrl>
        <imageWidth>45</imageWidth>
        <imageHeight>45</imageHeight>
      </record>
      <record name="status">
        <id>867</id>
        <uniqueId>collaboration-frontier</uniqueId>
        <message>&lt;p&gt;&lt;a class="wikiLink external" href="/show/
matt-rajkowski"&gt;Matt Rajkowski&lt;/a&gt; @&lt;a class="wikiLink
external" href="/show/collaboration-frontier"&gt;Collaboration
Frontier&lt;/a&gt;: Started a chat room. I'll probably hang out for a
bit.&lt;/p&gt;
</message>
        <messageText>Matt Rajkowski @Collaboration Frontier: Started a
chat room. I'll probably hang out for a bit.</messageText>
        <linkStartDate>1325094613267</linkStartDate>
        <linkStartText>2 weeks ago</linkStartText>
        <relativeDate>1325094613268</relativeDate>
        <relativeText>2 weeks ago</relativeText>
        <commentCount>0</commentCount>
        <type>user-entry</type>
        <typeId>237</typeId>
        <profile>Collaboration Frontier</profile>
        <profileImageUrl>http://social.preview.concursive.com/image/
2008090514-237-202-45x45-0a725de3ae6d31237492297760-760000000-2/
Collaboration-Frontier.png</profileImageUrl>
        <profileImageWidth>45</profileImageWidth>
        <profileImageHeight>45</profileImageHeight>
        <user>Matt Rajkowski</user>
        <imageUrl>http://social.preview.concursive.com/image/
2008090514-1-171-45x45-0a725de3ae6d31237492297760-760000000-2/matt-
headshot.png</imageUrl>
        <imageWidth>45</imageWidth>
        <imageHeight>45</imageHeight>
      </record>
    </recordSet>
  </response>
</concursive>
```

POST /subscription

Handles app store in-app subscriptions, including validating receipts, tracking orders, and granting additional functionality to the user.

POST /taskFolder

GET /taskFolderList

POST /task

GET /taskList

Returns a profile’s related profiles (from lists) that have a specific category.

URL

http://social.preview.concursive.com/api/taskList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

- **profile**
- **category**

Optional

none

Example request and response

/taskList

```
  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

POST /topic

Saves a moderator’s topic so that the moderator, and optionally users, can post responses. Topics can have a type of:

default, topic, discussion, challenge

Topics, discussions, and challenges can be pushed to all app users so that users can come back in and respond.

GET /videoList

Returns a profile’s videos

URL

http://social.preview.concursive.com/api/videoList

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Parameters

Required

none

Optional

- **view**
  mine, connected, public
- **category**
  A named category
- **profile**
  A profile’s uniqueId
- **title**
  A title of the video

Example request and response

/videoList

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="videoList" total="10">
      <record name="video">
        <videoId>10</videoId>
        <title>KCOY Sales Demo</title>
        <description>From the KCOY Film Festival.  Another Baker Film.
This is almost a legitimate promo, narrated by Dave Kerley.  Featuring
appearances by Frank Stelly, Dave Baker, Andrew Hecker, Nancy Paris,
Suzanne Huffman, Karen Devan, Marty Hernandez, Alan Ross, Suda Prohaska,
Governor Jerry Brown, Dave Gonzales, Bob Morrison, Robert J. Garcia, Gail
Thornton, Marty Scala, Ginny Woods, Dan Reedy, Dennis Bornhoft and Bob
Vincent (I know I missed more), plus CBS stars Lou Ferigno, Bill Bixby,
Carol O'Connor, Alan Alda, Loretta Switt, Larry Hagman, Ted Shackelford,
Joan Van Ark, James Earl Jones, Pernell Roberts, Polly Holliday, Ed Asner,
Loni Anderson, Catherine Bach (I almost got them all).</description>
        <link>http://www.youtube.com/watch?v=biuglhtmVVU</link>
        <thumbnail>http://i.ytimg.com/vi/biuglhtmVVU/default.jpg</
thumbnail>
        <embedCode>&lt;iframe class="youtube-player" type="text/html"
width="400" height="300" src="http://www.youtube.com/embed/biuglhtmVVU"
frameborder="0"&gt;&lt;/iframe&gt;</embedCode>
        <author>CreativeStuffVideo</author>
        <uniqueId>showing-off-features</uniqueId>
        <profile>Showing off features</profile>
        <profileLink>http://social.preview.concursive.com/show/showing-
off-features</profileLink>
        <profileImageUrl>http://social.preview.concursive.com/image/
2008051912-64-194-45x45-0a725de3ae6d31237492297760-760000000-2/
Sample.png</profileImageUrl>
        <profileImageWidth>45</profileImageWidth>
        <profileImageHeight>45</profileImageHeight>
        <publishedDate>1324040826809</publishedDate>
        <publishedText>3 weeks ago</publishedText>
        <publishedBy>2</publishedBy>
        <contactText>Matt Rajkowski</contactText>
        <contactUrl>http://social.preview.concursive.com/show/matt-
rajkowski</contactUrl>
        <contactImageUrl>http://social.preview.concursive.com/image/
2008090514-1-171-75x75-0a725de3ae6d31237492297760-760000000-2/matt-
headshot.png</contactImageUrl>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /wiki/:wiki-id

Returns the specified wiki.

URL

http://social.preview.concursive.com/api/wiki/:wiki-id

Supported formats

xml, json

Supported request methods

GET

Requires authentication

true

Alternate Invocation

/wiki/:profile-unique-id?subject=:subject

Parameters

Required

- **id**

Optional

- **profile**
  The profile’s unique id

- **subject**
  The subject of the wiki or leave empty for the root wiki entry

Example requests and response

/wiki/1
/wiki?profile=main-profile&subject=

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<concursive>
  <response>
    <status>0</status>
    <errorText/>
    <recordSet count="1" name="wikiList" total="1">
      <record name="wiki">
        <wikiId>1</wikiId>
        <subject/>
        <html>&lt;div class="infobox"&gt;&lt;table
class="pagedList"&gt;&lt;tr class="containerBody"&gt;&lt;td
class="formLabel"&gt;Type&lt;/td&gt;&lt;td&gt;Social Network&lt;/
td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/div&gt;&lt;h2
id="Site_Overview"&gt;Site Overview&lt;/h2&gt;
&lt;p&gt;Provide information about what this site is for.&lt;/p&gt;
&lt;h2 id="Background__History"&gt;Background &amp;amp; History&lt;/h2&gt;
&lt;p&gt;Provide information on the background and history of the
group.&lt;/p&gt;
&lt;h2 id="Management_Team"&gt;Management Team&lt;/h2&gt;
&lt;p&gt;&lt;a class="wikiLink newWiki" href="http://
social.preview.concursive.com/modify/main-profile/wiki/Community
+Manager"&gt;Community Manager&lt;/a&gt;&lt;/p&gt;
&lt;h2 id="Philosophy__Culture"&gt;Philosophy &amp;amp; Culture&lt;/h2&gt;
&lt;p&gt;Provide information about the site's philosophy and culture.&lt;/
p&gt;
</html>
        <modifiedBy>Matt Rajkowski</modifiedBy>
        <modifiedByUniqueId>matt-rajkowski</modifiedByUniqueId>
        <modifiedByImageUrl>http://social.preview.concursive.com/image/
2008090514-1-171-45x45-0a725de3ae6d31237492297760-760000000-2/matt-
headshot.png</modifiedByImageUrl>
        <modifiedByImageWidth>45</modifiedByImageWidth>
        <modifiedByImageHeight>45</modifiedByImageHeight>
        <modifiedDate>1266248377477</modifiedDate>
        <modifiedText>1 year ago</modifiedText>
        <ratingAverage>0.0</ratingAverage>
        <ratingCount>0</ratingCount>
        <ratingValue>0</ratingValue>
      </record>
    </recordSet>
  </response>
</concursive>
```

GET /wikiList