



Social Enterprise

Instant Topic Implementation Guide

© Social Strata, Inc.

1904 Third Avenue, Suite 525 • Seattle, WA 98101

Phone 206.283.5999 • Fax 206.299.4773

Document Last Revised: January 25, 2010

Social Strata, Inc. Instant Topic Implementation Guide ("API")

Status: Implemented

Author: Jonmark Weber

Copyright 2009 Social Strata, Inc.

1904 3rd Ave, Suite 525, Seattle, WA 98101

All rights reserved.

NOTICE:

This API is protected by copyright and the information described herein may be protected by one or more U.S. patents, foreign patents, or pending applications. Except as provided under the following license, no part of this API may be reproduced in any form by any means without the prior written authorization of Social Strata, Inc. and its licensors, if any.

By viewing, downloading or otherwise copying this API, you agree that you have read, understood, and will comply with all the terms and conditions set forth herein.

THIS SPECIFICATION IS PROVIDED "AS IS". Social Strata, INC. MAKES NO REPRESENTATIONS OR WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT THAT THE CONTENTS OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE OR THAT ANY PRACTICE OR IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADE SECRETS OR OTHER RIGHTS.

THIS SPECIFICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED INTO NEW VERSIONS OF THE SPECIFICATION, IF ANY. Social Strata, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES TO THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS SPECIFICATION AT ANY TIME

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL Social Strata, INC. OR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUE, PROFITS OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO ANY FURNISHING, PRACTICING, MODIFYING OR ANY USE OF THE SPECIFICATION, EVEN IF Social Strata, INC. AND/OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You will indemnify, hold harmless, and defend Social Strata, Inc. and its licensors from any claims arising or resulting from: (i) your use of the API; and/or (ii) any claims that later versions or releases of any API furnished to you are incompatible with the API provided to you under this license.

Use, duplication, or disclosure by the U.S. Government is subject to the restrictions set forth in this license and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii)(Oct 1988), FAR 12.212(a) (1995), FAR 52.227-19 (June 1987), or FAR 52.227-14(ALT III) (June 1987), as applicable.

You may wish to report any ambiguities, inconsistencies, or inaccuracies you may find in connection with your use of the API ("Feedback"). To the extent that you provide Social Strata, Inc. with any Feedback, you hereby: (i) agree that such Feedback is provided on a non-proprietary and non-confidential basis and (ii) grant Social Strata, Inc. a perpetual, non-exclusive, worldwide, fully paid-up, irrevocable license, with the right to sublicense through multiple levels of sublicensees, to incorporate, disclose, and use without limitation the Feedback for any purpose related to the API and future versions, implementations, and test suites thereof.

Introduction to Instant Topics

You can embed a topic directly on a web page through your own javascript implementation of Eve Social's Instant Topic Feature. This gives your users the ability to use the features of an Eve Social Community's topic immediately on your own site. For example, someone visiting a news article with an Instant Topic embedded on the page would not have to navigate to a separate forum to share their thoughts. They can do it instantly, right on the article page.

Creating an Instant Topic

Instant Topics are created through a server-to-server post request. The request takes this form:

<http://community.yoursite.com/createEmbeddedTopic!execute>

Here are the parameters for the request:

Parameter Name	Required?	Description
externalID	Required	A unique alphanumeric identifier for your Instant Topic
securityKey	Required	An alphanumeric key that secures your Instant Topic against unauthorized use. This must be defined in the "Settings" section of your Social Enterprise Global Control Panel. Just put your chosen securityKey value into the "Instant Topic Key" box and click "Update" to set this value.
title	Optional	A URL-encoded title for the topic. If left undefined, the title is derived from downloading the crosslinkUrl and parsing the title element out of the head of the pages HTML.
crossLinkUrl	Optional	A URL back to the originating page. If left undefined, the URL of the form from which the instant topic was created will be used.

Once a request is made, you can implement the Instant Topic on an external website.

Implementing Instant Topics

Instant Topics are implemented through the embedding of two forms into the desired page's HTML:

The Create/Edit Reply Embedded Form

This embedded form allows users to create and edit replies to the Instant Topic. The request takes this form:

<http://community.yoursite.com/postEmbeddedTopic!execute>

Parameter Name	Required?	Description
externalID	Required if embeddedTopic is not provided	The alphanumeric identifier you provided when creating your Instant Topic. Either the externalID or embeddedTopic parameter is required for the Create/Edit Reply Embedded Form.
embeddedTopic	Required if externalID is not provided	This parameter would take the value of the EmbeddedTopicOID provided in the SUCCESS response of the Create Instant Topic Request
resultURL	Required	URL on your domain for result information. (See appendix.)
replyPublic.body	Required	The message body for the reply.
replyOid	Optional	If an existing reply is being edited by the user, its replyOid should be specified here.

The Delete Reply Embedded Form

This embedded form allows for the deletion of replies to the Instant Topic. The request takes this form:

<http://community.yoursite.com/postEmbeddedTopic!execute>

Parameter Name	Required?	Description
externalID	Required if embeddedTopic is not provided	The alphanumeric identifier you provided when creating your Instant Topic. Either the externalID or embeddedTopic parameter is required for the Delete Reply Embedded Form.
embeddedTopic	Required if externalID is not provided	This parameter would take the value of the EmbeddedTopicOID provided in the SUCCESS response of the Create Instant Topic Request
resultURL	Required	URL on your domain for result information. (See appendix.)
replyOid	Required	OID for reply, used to specify the reply when editing.

Info JsonP Requests

You may wish to get further information about an Instant Topic to fully integrate it into your site. There are two JsonP requests you can send to get further information about the Instant Topic:

Instant Topic Info

This JsonP request returns information about the topic itself, as well as the logged-in user.

It will return the following information:

- User OID of the logged-in user
- User Display Name of the user
- The URL of the user's profile page
- Topic OID of the embedded topic
- externalID of the embedded topic
- The number of replies

The request takes this form:

<http://community.yoursite.com/embeddedTopicInfo>

Parameter Name	Required?	Description
externalID	Required if embeddedTopic is not provided	The alphanumeric identifier you provided when creating your Instant Topic. Either the externalID or embeddedTopic parameter is required for the Delete Reply Embedded Form.
embeddedTopic	Required if externalID is not provided	This parameter would take the value of the EmbeddedTopicOID provided in the SUCCESS response of the Create Instant Topic Request
callback	Required	This parameter is provided automatically by the jquery jsonp ajax request framework.

Sample Result:

```
[callback]({
  user: {
    oid: '152966031785675',
    displayName: 'JoeUser',
    url: ' http://joe.user.yourcommunity.com'
  },
  embeddedTopic: {
    oid: '53915197861562788',
    externalID: 'EmbeddedTopic?1',
    replies: 1
  }
});
```

The root level user node is representative of the currently logged-in user. If this is not present, it means the user is not logged into the application.

Instant Topic Replies

This JsonP request returns information about the topic's replies, as well as the logged-in user.

It will return the following information:

- User OID of the logged-in user
- User Display Name of the user
- The URL of the user's profile page
- Topic OID of the embedded topic
- externalID of the embedded topic
- The number of replies
- The pagination used for returned results
- The OID for the reply
- The body of the reply
- A copy of the reply body for editing purposes
- The date of the reply
- User OID of the author of the reply
- User Display Name of the author of the reply
- The URL of the profile page for the author of the reply

The request takes this form:

<http://community.yoursite.com/embeddedTopicReplies>

Parameter Name	Required?	Description
externalID	Required if embeddedTopic is not provided	The alphanumeric identifier you provided when creating your Instant Topic. Either the externalID or embeddedTopic parameter is required for the Delete Reply Embedded Form.
embeddedTopic	Required if externalID is not provided	This parameter would take the value of the EmbeddedTopicOID provided in the SUCCESS response of the Create Instant Topic Request
callback	Required	This parameter is provided automatically by the jquery jsonp ajax request framework.
page	Optional	This is the page number of the topic from which you want to start returning results. Default is one.
resultsPerPage	Optional	Number of replies to include per page. Default is twenty.
forceLastPage	Optional	Returns the last page of results. Default is false.

Sample Results:

```
[callback]({
  user: {
    oid: '152966031785675',
    displayName: 'JoeUser',
    url: ' http://joe.user.yourcommunity.com'
  },
  embeddedTopic: {
    oid: '53915197861562788',
    externalID: 'EmbeddedTopic?1',
    replies: 1,
    pagination: {
      page: 1,
      maxPage: 1,
      resultsPerPage: 20,
      results: 1
    }
  },
  replies: [{
    oid: '54055943639847194',
    body: 'First Reply Message Body',
```

```
bodyForEdit: 'First Reply Message Body',
datePosted: new Date(1248191644000),
editable: true,
user: {
  oid: '152966031785675',
  displayName: 'JoeUser', url: ' http://joe.user.yourcommunity.com'
}
}]
});
```

The root level user node is representative of the currently logged-in user. If not present then the user is not logged into the application. The editable property on the replies[X] element refers to whether or not the current user can edit/delete that reply.

Appendix: The Result Url

The Result URL is used to redirect the user after a post method is made to a customer domain that should have jsript access to the embedded page. Parameters will be added to the url to give context. The root parameter for this is "type," which defines what kind of action was taken, or more importantly, the status.

These are the types, along with the parameters that are present for each:

type=new-reply

externalID=Embedded Topics External ID

topicPostCount=Number of replies for the embedded topic after the reply was added

type=edit-reply

externalID=Embedded Topics External ID

type=delete-reply

externalID=Embedded Topics External ID

type=error

message= the error message which prevented the desired action

optional:

externalID= if an embedded topic exists for this request, then its ID will be included