

EXHIBIT 20

Doc. 656-At. 20

Leader Technologies, Inc. v. Facebook, Inc.



topics

- ▶ Main Page
- ▶ Core Components
- ▶ Platform Policies
- ▶ Facebook Connect
- ▶ Internationalization
- ▶ Mobile
- ▶ Client Libraries
- ▶ Support Resources
- ▶ Developer Roadmap

get involved

- ▶ Contribute
- ▶ Developer site
- ▶ Platform FAQ

reference

- ▶ RESTful API
- ▶ JavaScript API
- ▶ FQL
- ▶ XFBML
- ▶ FBML
- ▶ FBJS

wiki

- ▶ Random page
- ▶ Recent changes

search

toolbox

- ▶ What links here
- ▶ Related changes
- ▶ Upload file
- ▶ Special pages
- ▶ Printable version
- ▶ Permanent link

Stream.publish

Description

This method publishes a post into the stream on the Wall of a friend or a Facebook Page, group, or event connected to the current session or specified user (but not to an application profile page). By default, this call publishes to the current session user's Wall, but if you specify a user ID, Facebook Page ID, group ID, or event ID as the `target_id`, then the post appears on the Wall of the target, and not the user posting the item.

The post also appears in the streams (News Feeds) of any user connected to both the actor and the target of the post.

Before your application can publish to the stream using this method, the user or Page must grant your application the `publish_stream` [extended permission](#). If the user previously granted your application the permission to publish short stories into the News Feed automatically, then you don't need to prompt for this permission in order to call this method.

Note: To give users control over what gets published to their streams, you should use [Feed forms](#) (rendered with `Facebook.streamPublish` or `FB.Connect.streamPublish`) instead of `stream.publish`. This method is intended to be used in cases where [Feed forms](#) are not available or do not make sense in the natural workflow. For more information, read about the [publish_stream permission](#).

You can give your users the opportunity to add their own message to the post.

To provide rich content like MP3 audio, Flash, or an image, you can supply a predefined JSON-encoded object called `attachment`. Facebook formats the attachment into the post. The attachment is described in [Attachment \(Streams\)](#).

Note: The examples below are all server-side versions of the call in PHP. To see the JavaScript equivalents, check out [FB.Connect.streamPublish](#).

Contents [hide]

- 1 Description
- 2 Parameters
- 3 Example Requests
- 4 Response
- 5 Error Codes
- 6 Notes
- 7 See Also

Parameters

Required	Name	Type	Description
optional	<code>session_key</code>	<code>string</code>	The session key of the logged in user, or the session key provided when the user granted your application the <code>offline_access</code> extended permission . The session key is automatically included by our PHP

Plaintiff's Trial Exhibit
PTX-911
Case No. 08-CV-00862

		client. This is only required if you don't specify a <code>uid</code> , or if a desktop application calls <code>stream.publish</code> .
<code>format</code>	<code>string</code>	The desired response format, which can be either <code>XML</code> or <code>JSON</code> . (Default value is <code>XML</code> .)
<code>callback</code>	<code>string</code>	Name of a function to call. This is primarily to enable cross-domain JavaScript requests using the <code><script></code> tag, also known as JSONP, and works with both the XML and JSON formats. The function will be called with the response passed as the parameter.
<code>message</code>	<code>string</code>	The message the user enters for the post at the time of publication. If the message is a status update (that is, you're not including an attachment), it can contain up to 420 characters. Otherwise, if the post contains an attachment, the message can contain up to 10,000 characters.
<code>attachment</code>	<code>object</code>	A JSON-encoded object containing the text of the post, relevant links, a media type (image, mp3, flash), as well as any other key/value pairs you may want to add. See Attachment (Streams) for more details. Note: If you want to use this call to update a user's status, don't pass an <code>attachment</code> ; the content of the <code>message</code> parameter will become the user's new status and will appear at the top of the user's profile.
<code>action_links</code>	<code>array</code>	A JSON-encoded array of action link objects, containing the link text and a hyperlink.
<code>target_id</code>	<code>string</code>	The ID of the user, Page, group, or event where you are publishing the content. If you specify a <code>target_id</code> , the post appears on the Wall of the target profile, Page, group, or event, not on the Wall of the user who published the post. This mimics the action of posting on a friend's Wall on Facebook itself. Note: If you specify a Page ID as the <code>uid</code> , you cannot specify a <code>target_id</code> . Pages cannot write on other users' Walls. Note: You cannot publish to an application profile page's Wall.
<code>uid</code>	<code>string</code>	The user ID or Page ID of the user or Page publishing the post. If this parameter is not specified, then it defaults to the session user. If you specified a <code>session_key</code> , and that session user is a Page admin, then you can specify a Page ID here to publish to one Page for which the session user is an admin. Note: If you specify a Page ID as the <code>uid</code> , you cannot specify a <code>target_id</code> . Pages cannot write on other users' Walls.
<code>privacy</code>	<code>object</code>	A JSON-encoded object that defines the privacy setting for a post, video, or album. It contains the following fields. <ul style="list-style-type: none"> ▸ <code>value</code> (string): The privacy value for the object, specify one of <code>EVERYONE</code>, <code>CUSTOM</code>, <code>ALL_FRIENDS</code>, <code>NETWORKS_FRIENDS</code>, <code>FRIENDS_OF_FRIENDS</code>. ▸ <code>friends</code> (string): For <code>CUSTOM</code> settings, this indicates which users can see the object. Can be one of <code>EVERYONE</code>, <code>NETWORKS_FRIENDS</code> (when the object can be seen by networks and friends), <code>FRIENDS_OF_FRIENDS</code>, <code>ALL_FRIENDS</code>,

SOME_FRIENDS, SELF, or NO_FRIENDS (when the object can be seen by a network only).

- networks (string): For CUSTOM settings, specify a comma-separated list of network IDs that can see the object, or 1 for all of a user's networks.
- allow (string): When friends is set to SOME_FRIENDS, specify a comma-separated list of user IDs and friend list IDs that *can* see the post.
- deny (string): When friends is set to SOME_FRIENDS, specify a comma-separated list of user IDs and friend list IDs that *cannot* see the post.

Only the user can specify the privacy settings for the post. You can create an interface that lets the user specify the privacy setting. For CUSTOM settings, use [friends.get](#) and [friends.getLists](#) to get the user's friends and friend lists to populate the interface, then pass along the selections to the privacy object.

Privacy Policy: Any non-default privacy setting must be intentionally chosen by the user. You may not set a custom privacy setting unless the user has proactively specified that they want this non-default setting.

Example Requests

Setting a User's Status

```
$message = 'in ur tubez';
$facebook->api_client->stream_publish($message);
```

Publishing a Post Containing an Image, Action Link, and Custom Metadata

```
$message = 'Check out this cute pic.';
$attachment = array(
    'name' => 'i\'m bursting with joy',
    'href' => 'http://icanhascheezburger.com/2009/04/22/funny-pictures-bursting-with-joy/',
    'caption' => '{*actor*} rated the lolcat 5 stars',
    'description' => 'a funny looking cat',
    'properties' => array('category' => array(
        'text' => 'humor',
        'href' => 'http://www.icanhascheezburger.com/category/humor',
        'ratings' => '5 stars'),
    'media' => array(array('type' => 'image',
        'src' => 'http://icanhascheezburger.files.wordpress.com/2009/03/funny-pictures-your-cat-is-bursting-with-joy1.jpg',
        'href' => 'http://icanhascheezburger.com/2009/04/22/funny-pictures-bursting-with-joy/')),
    'latitude' => '41.4', //Let's add some custom metadata in the form of key/value pairs
    'longitude' => '2.19');
$action_links = array(
    array('text' => 'Recaption this',
        'href' => 'http://mine.icanhascheezburger.com/default.aspx?tiid=1192742&recap=1#step2'));
$attachment = json_encode($attachment);
$action_links = json_encode($action_links);
$facebook->api_client->stream_publish($message, $attachment, $action_links);
```

Publishing a Post Containing Flash, Action Link, and a Target

```

$message = 'Watch this video!';
$attachment = array(
    'name' => 'ninja cat',
    'href' => 'http://www.youtube.com/watch?v=muLIPWjks_M',
    'caption' => '{*actor*} uploaded a video to www.youtube.com',
    'description' => 'a sneaky cat',
    'properties' => array('category' => array(
        'text' => 'pets',
        'href' => 'http://www.youtube.com/browse?s=mp&t=t&c=15'),
        'ratings' => '5 stars'),
    'media' => array(array('type' => 'flash',
        'swfsrc' => 'http://www.youtube.com/v/fzzjgBAaWZw&hl=en&fs=1',
        'imgsrc' => 'http://img.youtube.com/vi/muLIPWjks_M/default.jpg?
h=100&w=200&sig=__wsYqEz4uZUOvBIb8g-wljxpf3Q=',
        'width' => '100',
        'height' => '80',
        'expanded_width' => '160',
        'expanded_height' => '120')));
$action_links = array(
    array('text' => 'Upload a video',
        'href' => 'http://www.youtube.com/my_videos_upload'));
$target_id = 2342314;
$facebook->api_client->stream_publish($message, $attachment, $action_links, $target_id);

```

Publishing a Post With Image and Action Link in C#

```

attachment attach = new attachment();

attach.caption = "Caption for attachment";
attach.description = "Description for attachment";
attach.href = "http://www.joemagner.com";
attach.name = "Hopefully this works";

attachment_media attach_media = new attachment_media();
attach_media.type = attachment_media_type.image;

attachment_media_image image = new attachment_media_image();
image.type = attachment_media_type.image;
image.href = "http://icanhascheezburger.com/2009/03/30/funny-pictures-awlll-gone-cookie-now/";
image.src = "http://icanhascheezburger.files.wordpress.com/2009/03/funny-pictures-kitten-
finished-his-milk-and-wants-a-cookie.jpg";

List<attachment_media> attach_media_list = new List<attachment_media>();
attach_media_list.Add(image);

attach.media = attach_media_list;

attachment_property attach_prop = new attachment_property();
attachment_category attach_cat = new attachment_category();
attach_cat.text = "Sample";
attach_cat.href = "#";

attach_prop.category = attach_cat;
//attach_prop.ratings = "5 stars";

attach.properties = attach_prop;

/* action links */
List<action_link> actionlink = new List<action_link>();

action_link all = new action_link();
all.href = "http://www.genuineinteractive.com/";
all.text = "Genuine Interactive";

actionlink.Add(all);

// Create the service
FacebookService fbService = new FacebookService();
fbService.ApplicationKey = AppSettings.GetKeyAsString("APIKey");
fbService.Secret = AppSettings.GetKeyAsString("Secret");
fbService.IsDesktopApplication = false;

```

```

fbService.SessionKey = sessionKey;
fbService.uid = uid;

return fbService.API.stream.publish("Message goes here", attach, actionlink,
fbService.uid.ToString(), 0);

```

Publishing a Post With Image and Action Link in C# using facebook developer toolkit v2

```

public void Post(facebook.API fbAPI, string appLink)
{
string response = fbAPI.stream.publish(
    "is a good guy.",
    new attachment() {
        name = "I am a good guy !",
        href = appLink,
        caption = "{*actor*} is now a good guy",
        description = "Helping other people, I became a new good guy.",
        properties = null,
        media = new List<attachment_media>() {
            new attachment_media_image() { src =
"http://www.goodguy.com/goodGuy.png", href = appLink }
        }
    },
    new List<action_link>() {
        new action_link() { text = "Become a good guy", href = appLink }
    },
    null,
    0);
}

```

Response

This call returns a `post_id` string containing the ID of the stream item upon success. If the call fails, it returns an error code instead.

Error Codes

For a complete list of error codes, see [Error codes](#).

Code	Description
1	An unknown error occurred.
100	Invalid parameter.
102	Session key invalid or no longer valid (if it's a desktop application and the session is missing).
200	Permissions error. The application does not have permission to perform this action.
210	User not visible. The user doesn't have permission to act on that object.
340	Feed action request limit reached.

Notes

You can call this method using a [session secret](#), and not the application secret (for example, for a [Facebook Connect](#) site or [desktop application](#)).

See Also

[Using the Open Stream API](#)

Categories: [API functions](#) | [Session Secret API](#) | [Open Stream API](#) | [Session Optional API](#)

This page was last modified 21:46, 30 March 2010. This page has been accessed 198,852 times.

[Privacy policy](#)

[About](#)

[Facebook Developer Wiki](#)

[Disclaimers](#)

