This page last changed on Dec 14, 2015 by jack.van.ooststroom.

JavaScript Integration

This is the lowest-level client-side integration, and assumes only a basic HTML page with an embedded JavaScript implementation that initializes and interacts with the ICEpush client.

Initialization and Page Inclusion

The ICEpush Javascript Bridge must be included in the page.

```
<html>
  <head>
    <script type="text/javascript" src="code.icepush"></script>
  </head>
  <body>
    ...
  </body>
</html>
```

The script tag will cause the browser to request ./code.icepush, which, by convention, is mapped to the ICEpush Servlet in your web.xml file.

```
<servlet>
    <servlet-name>icepush</servlet-name>
    <servlet-class>org.icepush.servlet.ICEpushServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>icepush</servlet-name>
    <url-pattern>*.icepush</url-pattern>
</servlet-mapping>
```

The ICEpush Servlet will serve the javascript code when this url is requested, and the ICEpush Bridge will then be loaded and initialized in the client browser.

Usage

Once the ICEpush bridge is loaded and initialized in the client browser, it is possible for JavaScript application logic to register for inclusion in a group, register notification callbacks, get current notifications, and initiate notifications as follows:

```
<script type="text/javascript">
  var pushId = ice.push.createPushId();
  ice.push.addGroupMember("rooms", ice.push.createPushId());
  ice.push.register([pushId], window.refreshChatRoomsList);
</script>
```

ICEpush Javascript API

`ice.push.createPushId(retries);`

(Until 4.0)

Creates a unique push id that can be used for group registration. Multiple push ids can be created for a single page in an application.

`ice.push.createPushId(retries, callback);`

(As of 4.1)

Creates a unique push id that can be used for group registration. This function **replaced** the previous `ice.push.createPushId(retries)` function and requires a callback function which is used to pass the newly created unique push id. Multiple push ids can be created for a single page in an application.

`ice.push.addGroupMember(groupName, pushId);`

Adds a previously created push id to a group.

`ice.push.register(pushIds, callback);`

Registers a callback function to a list of push ids. When a push notification occurs for a group that has a registered push id, the ICEpush Bridge will call the registered callback listener function in the browser.

`ice.push.deregister(pushId);`

Unregisters a previously created push id with any groups and callback listener functions.

`ice.push.getCurrentNotifications();`

Can be called in a callback function to find the current push ids which are being notified.

`ice.push.notify(groupName, options);`

(As of 3.4)

Trigger a notification event for the specified *groupName*. The optional *options* parameter can be a JS object with properties that are read as name-value pairs.
For example:

```
ice.push.notify(groupName, {subject:'hello', detail:'how are you'});
```

`ice.push.removeGroupMember(groupName, pushId);`

Removes a push id from a previously joined group.

Payload Processing

Payload processing is entirely application-specific. A `callback-function` is registered against a `pushId` and application logic within that `callback-function` must initiate payload retrieval from the server using an `XMLHttpRequest` and processing the payload in an application-specific way.