

ICEpush : Grails Integration

This page last changed on Dec 15, 2010 by [steve.maryka](#).

Grails

[Grails](#) is a leading productive development framework based on JAVA famous technologies Spring, Hibernate and Groovy. Grails brings a public plugin repository, allowing developers to simply retrieve framework dependencies when they want.

You can get ICEpush capabilities in your application with the following [Grails plugin](#), just type :

```
grails install-plugin icepush
```

Without more configuration you will have access to :

1. [Dynamically injected methods](#) in grails artefacts which provide a simple access to ICEpush API
2. A set of [GSP tags](#) that can be included in the page to register pushIDs or manage regions of dynamic content.
3. [ICEfaces 2 plugin push](#) capabilities and collaborative application 'scaffolding'

GSP Tags

<icep:bridge>

Includes ICEpush library in the current page.

```
<html>
<head>
...
<icep:bridge />
</head>
...
</html>
```

<icep:pushId>




Creates and outputs a pushId.

```
<g:set var='pushId_1' value="{icep:pushId}" />
```

<icep:register>

The `register` tag creates a unique pushID if not specified and registers a JavaScript callback with it. It optionally declares a group that the pushID belongs to.








```
<icep:register callback="myScript" group="myGroup" />
```


Attribute	Required	Description
callback		A JavaScript callback function capable of handling notifications to the assigned pushID.
group		A group name that the pushID will be registered with. If the group is not supplied it defaults to the pushID itself, resulting in a unique group of 1 for the pushID.
pushid		A specified pushId to use instead of creating a new one

<icep:region>

The `region` tag creates a unique pushID if not specified and associates it with a specific included page. The dynamic page content is included initially when the page loads, and is reloaded using an `ajax GET` when notifications occur to the pushID. The group name is included as a parameter to the `GET` request, so it is available to the included page. It optionally declares a `<div>` id for the updated region, a group that the pushID belongs to, and a notifier bean responsible for triggering notifications.

```
<icep:region group="chatGroup" controller="chat" action="messages"/>
```







Attribute	Required	Description
wrapperid		ID of the <code><div></code> that will form the region and contain the dynamic content. If defined, it must be unique to the page. If not defined, it defaults to the unique pushID.
group		A group name that the pushID will be registered with. If the group is not supplied it defaults to the pushID itself, resulting in a unique group of 1 for the pushID.
action		the name of the action to use in the include
controller		the name of the controller to use in the include
id		the id to use in the include
params		a map containing request parameters
evalJS		Determines whether any embedded scripts in the content that is fetched after a Push notification are evaluated

		(default true). Evaluating embedded scripts does carry some client-side overhead that the developer should be aware of. If the region tag is fetching areas of large content that is known not to have any embedded scripts (eg. a large amount of table data), then performance may be improved by setting <code>evalJS</code> to false. Please note, that if it is required to nest region tags, the outer tags will have to have <code>evalJS=true</code> (default) for the rendered JavaScript of the inner region tags to be evaluated, which is required for the proper functioning of the region tag.
<code>pushid</code>		A specified <code>pushId</code> to use instead of creating a new one

<icep:page>

The `page` tag causes a full page refresh to occur to the named group. Like `region` tag, it targets a set of Grails request attributes like 'action' or 'controller'.

```
<icep:region group="breakingNews" controller="news" action="alert"/>
```

Attribute	Required	Description
<code>group</code>		A group name that the <code>pushID</code> will be registered with. If the group is not supplied it defaults to the <code>pushID</code> itself, resulting in a unique group of 1 for the <code>pushID</code> .
<code>action</code>		the name of the action to use in the include
<code>controller</code>		the name of the controller to use in the include
<code>id</code>		the id to use in the include
<code>params</code>		a map containing request parameters
<code>pushid</code>		A specified <code>pushId</code> to use instead of creating a new one

<icep:push>

The `push` tag causes a push notification to occur to the named group, on when the page loads.

```
<icep:push group="myGroup"/>
```

Attribute	Required	Description
group		A group name to receive push notifications.

Notification

For notifications to occur, the notification business logic must have a valid `PushContext` to use. While in a normal or ajax-based request/response cycle the `PushContext` is readily available using the `ServletContext`. In Grails controllers, services or beans you can access it by typing `pushContext` :

```
def action = {  
  if(pushContext)  
    ...  
}
```

If, however, notification triggers are to occur outside a request/response cycle, the notifying thread of execution still needs a valid `PushContext` to use.

Dynamic Methods and Properties

`pushContext`

Access to the actual `PushContext` instance.

```
def _actualPushContext = pushContext
```

`pushId`

Return a new `PushId` on each call, to be used inside a request/response cycle.

```
def myId = pushId
```

`addToPushGroup(String group, String pushId?)`

Add a `pushId` to specified group. If `pushId` is not specified, a new one is created and returned.

```
//One parameter syntax, returns a new pushId  
def newId = addToPushGroup('chat')  
//Two parameters syntax, reuse a specified pushId  
addToPushGroup('news',newId)
```

`removeFromPushGroup(String group, String pushId)`

Remove the provided pushId from the specified group.

```
//retrieve a pushId from a parameter
def _pushId = params.pushId

removeFromPushGroup('chat', _pushId)
```

`push(String group)`

Notify target group.

```
serviceMethod(){
  ...
  push 'chat'
}
```

Group Naming

Group naming employees application-specific strategies. While group naming is orthogonal to JSP bean scopes, they can effectively be associated with a scope.

- Window-scoped groups are not named, and use only the unique pushID assigned during registration. These names/pushIDs will change each time the page is loaded.
- Session-scoped groups include the session ID: `group="${session.id}.myGroup"`
- Application-scoped groups have static names: `group="myGroup"`
- Deployment-scoped groups have static names beginning with a "/": `group="/myGroup"`



Deployment scope is only available in multi-application deployments with Enterprise Push Server.

ICEfaces 2 Plugin Push capabilities

You can combine ICEpush plugin and ICEfaces2 plugin to leverage outstanding collaborative applications. ICEfaces 2 plugin is not yet available, more informations will be provided when it will be released.

Credits

This plugin is contributed by Stephane MALDINI from [doc4web consulting](#) under Apache License V2.