

ICEpush : Wicket Integration

This page last changed on Mar 19, 2010 by [steve.maryka](#).

Wicket Integration

Wicket integration comes in the form of a Panel. Extending `org.icepush.integration.wicket.core.PushPanel` automatically creates a group with the same name as the Panel id, allows you to make push call(s) when necessary, and update your model and GUI by implementing the `pushCallback()` method.

Enable Wicket Application for Push

1. Add `icepush.jar` and `icepush-wicket.jar` to your application.
2. In the **web.xml** of your application, add the following:

```
<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>
```

3. Nest the following in the head tag of your page:

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

Implement Push in Wicket Application

1. Create a Wicket Panel by extending `org.icepush.integration.wicket.core.PushPanel`.
2. Add push JavaScript to your panel's html file:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns:wicket>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>ExamplePushPanel</title>
  </head>
  <body>
    <wicket:panel>
      <script wicket:id="pushJavascript" type="text/javascript">
      </script>
      <!-- INSERT CONTENTS OF YOUR PANEL HERE -->
    </wicket:panel>
  </body>
</html>
```



Do not nest this script in a component that is updated via AJAX. This will result in the script being executed multiple times.

3. Add push call(s), as necessary, to your Java class:

```
leftForm.add(new AjaxButton("leftButton") {  
    protected void onSubmit(AjaxRequestTarget target, Form form) {  
        // PUSH CALL  
        push();  
        isPushMine=true;  
        target.addComponent(this.getParent());  
    }  
});
```

4. Implement the `pushCallback(AjaxRequestTarget target)` method in your panel's Java class to update your model and render the appropriate components on callback.

```
protected void pushCallback(AjaxRequestTarget target) {  
    if(isPushMine){  
        pushList.add("My Push.");  
    }else{  
        pushList.add("Pushed From Another User.");  
    }  
    isPushMine=false;  
    pushListView.modelChanged();  
    target.addComponent(leftForm);  
}
```