JavaScript Plugins

- Introduction
- Current status:
  - Native JS plugins
    - Generate plugin stub from Maven artifact
    - Important parts of JS plugin
    - Build and deployment
  - GWT plugins
    - Generate plugin stub from Maven artifact
    - Important parts of GWT plugin
    - Build and deployment
- To be defined:
  - Client side extension points
  - Registering server side services

Dariusz Łuksza (@dluksza)
CollabNet Engineering Office
Potsdam, Germany
What is Gerrit?
What is Gerrit?

### Search for status:open

<table>
<thead>
<tr>
<th>ID</th>
<th>Subject</th>
<th>Owner</th>
<th>Project</th>
<th>Branch</th>
<th>Updated</th>
<th>CR</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1c34c22cf</td>
<td>Make the text for &quot;Register&quot; customizable</td>
<td>Chad Horohoe</td>
<td>gerrit</td>
<td>master</td>
<td>05:29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1cb32083</td>
<td>Remove unused class member variable</td>
<td>David Pursehouse</td>
<td>gerrit</td>
<td>master</td>
<td>05:25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1806b5b3b</td>
<td>Don't advertise refs/cache-automerge during push</td>
<td>Shawn Pearce</td>
<td>gerrit</td>
<td>master</td>
<td>10. Nov</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>1be8b4898</td>
<td>Draft 2.6 release notes</td>
<td>Shawn Pearce</td>
<td>gerrit</td>
<td>master</td>
<td>10. Nov</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1634b9a67</td>
<td>Tomcat Install Instructions</td>
<td>Jason Huntley</td>
<td>gerrit</td>
<td>master</td>
<td>9. Nov</td>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>13c4abf9b</td>
<td>Document the submit_type in the Prolog cookbook.</td>
<td>Saša Živkov</td>
<td>gerrit</td>
<td>master</td>
<td>9. Nov</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19cf05af</td>
<td>Add a new projects menu that displays the recently accessed projects</td>
<td>Edwin Kempin</td>
<td>gerrit</td>
<td>master</td>
<td>9. Nov</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1a3248cb3</td>
<td>Updated GroupBackend to lookup all groups at construction. Now suggest...</td>
<td>Colby Ranger</td>
<td>plugins/google-apps-group</td>
<td>master</td>
<td>8. Nov</td>
<td></td>
<td></td>
</tr>
<tr>
<td>177606e7e</td>
<td>Implemented GroupBackend membershipsOf()</td>
<td>Colby Ranger</td>
<td>plugins/google-apps-group</td>
<td>master</td>
<td>8. Nov</td>
<td></td>
<td></td>
</tr>
<tr>
<td>179a6b5d0</td>
<td>Implemented GroupBackend get() and suggest()</td>
<td>Colby Ranger</td>
<td>plugins/google-apps-group</td>
<td>master</td>
<td>8. Nov</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.5 Plugin infrastructure
What can be done in 2.5?

Extension points:
- Stable API
- Listen for events
- Register new SSH command
- Register new HTTP servlets

Plugins:
- API is strongly coupled with Gerrit internals... because they are Gerrit internals
- Can do all the rest...
What can be done in 2.5?

Extension points:
- Stable API
- Listen for events
- Register new SSH command
- Register new HTTP servlets

Plugins:
- API is strongly coupled with Gerrit internals... because they are Gerrit internals
- Can do all the rest... except of thins that it cannot do.
What can be done in 2.5?

Extension points:
- Stable API
- Listen for events
- Register new SSH command
- Register new HTTP servlets

Plugins:
- API is strongly coupled with Gerrit internals... because they are Gerrit internals
- Can do all the rest... except of thins that it cannot do:
  - Cannot define own authentication method,
  - Cannot change Web UI
JavaScript Plugins
$ mvn archetype:generate \\n  -DarchetypeGroupId=com.google.gerrit \\n  -DarchetypeArtifactId=gerrit-plugin-js-archetype \\n  -DarchetypeVersion=2.6-SNAPSHOT \\n  -DgroupId=com.collabnet \\n  -DartifactId=gerrit-js-plugin-stub
Important parts of JS plugin

Plugin registration in Gerrit:

```java
package com.collabnet.gerrit.js;

import com.google.gerrit.extensions.annotations.Listen;
import com.google.gerrit.extensions.webui.JavaScriptPlugin;

@Listen
public class JsExtension extends JavaScriptPlugin {
    public JsExtension() {
        super("plugin-init.js");
    }
}
```
Simple build and deploy procedure:

$ mvn clean package
$ ssh -p 29418 admin@localhost gerrit plugin install - \
/path/to/plugin-project/target/plugin.jar
GWT Plugins
$ mvn archetype:generate \
  -DarchetypeGroupId=com.google.gerrit \
  -DarchetypeArtifactId=gerrit-plugin-gwt-archetype \
  -DarchetypeVersion=2.6-SNAPSHOT \
  -DgroupId=com.collabnet \
  -DartifactId=gerrit-gwt-plugin-stub
Important parts of GWT plugin

Gerrit-plugin-gwtui dependency in pom.xml:

```xml
<dependency>
  <groupId>com.google.gerrit</groupId>
  <artifactId>gerrit-plugin-gwtui</artifactId>
  <version>${Gerrit-ApiVersion}</version>
</dependency>
```
Important parts of GWT plugin

Plugin registration in Gerrit:

```java
package com.collabnet.gerrit.gwt;

import com.google.gerrit.extensions.annotations.Listen;
import com.google.gerrit.extensions.webui.GwtPlugin;

@Listen
public class GwtExtension extends JavaScriptPlugin {
    public GwtExtension() {
        super("gwt_module_name");
    }
}
```
Gerrit linker registration in gwt.xml:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<module rename-to="gwt_module_name">
    <!-- Inherit the core Web Toolkit stuff. -->
    <inherits name="com.google.gwt.user.User"/>
    <!-- Inherit Gerrit GWT linker -->
    <inherits name="com.google.gerrit.Plugin"/>
    <entry-point class="com.collabnet.gerrit.gwt.client.HelloGwt"/>
</module>
```

Entry point class must extend:
- `com.google.gerrit.client.Plugin`
Build and deployment

Same as for JS plugins, simple build and deploy procedure:

```bash
$ mvn clean package
$ ssh -p 29418 admin@localhost gerrit plugin install - \
/path/to/plugin-project/target/plugin.jar
```
Extension API
Requirements for extension API

Requirements:

Client side:
• Easy to consume
• Easily accessible from GWT and JavaScript
• Allow cross-plugin interactions/dependencies (?)

Server side:
• Automatically bind services
• Support server side dependency injection
• Use common protocol for GWT and JavaScript (?)
  ➢ Use JSON
  ➢ Or provide JavaScript API for GWT base protocol (RequestFactory?)
First draft idea of GWT client side extension point API:

```java
import com.google.gwt.user.client.ui.Widget;

public interface GerritUi {
    // we should also consider here async calls
    // and somehow provide data from UI
    public interface ExtensionListener {
        boolean performAction();
    }
    public interface Actions {
        void onChangeSubmitListener(ExtensionListener l);
        void removeOnChangeSubmitListener(ExtensionListener l);
    }
    public interface Containers {
        void appendToMainMenu(Widget widget);
        void removeFromMainMenu(Widget widget);
    }
}

Actions getActions();
Containers getContainers();
```
JavaScript Plugins

Dariusz Łuksza (@dluksza)
CollabNet Engineering Office
Potsdam, Germany