gerrit

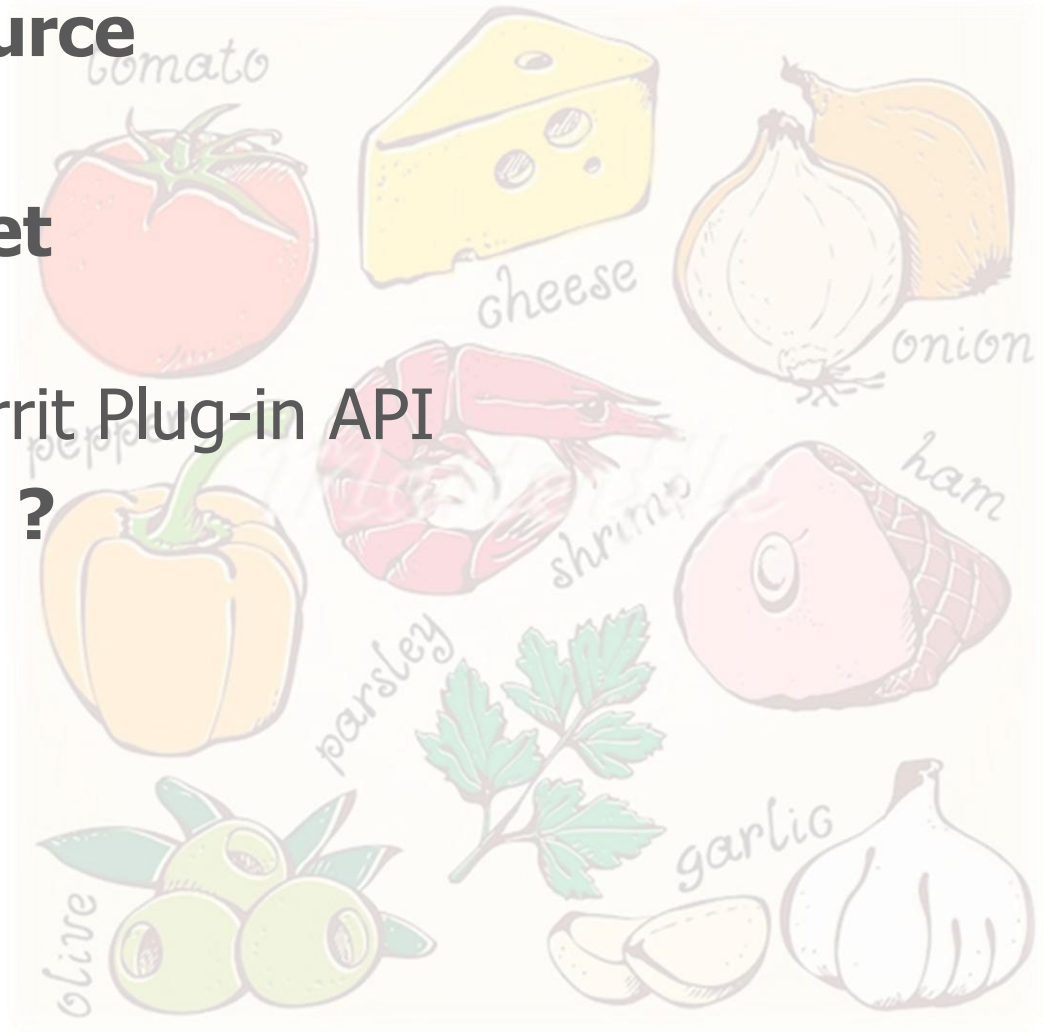
Gerrit Code Review

How to cook a Plugin

Base ingredients

GerritForge™

1. **Gerrit code-review source**
   - ✓ Ver. 2.5 or later
2. **Maven 3.x and Internet connectivity**
   - ✓ Dependencies and Gerrit Plug-in API
3. **Focus for your plug-in ?**
   - ✓ HTTP/HTML ?
   - ✓ SSH command ?
   - ✓ Init step ?
   - ✓ Gerrit GUI ?
   - ✓ Gerrit internals ?



Step-1: get Gerrit 2.5 source

GerritForge™

```
$ git clone -b v2.5 https://gerrit-review.googlesource.com/gerrit
Cloning into 'gerrit'...
remote: Counting objects: 5622, done
remote: Finding sources: 100% (3294/3294)
remote: Getting sizes: 100% (2224/2224)
remote: Compressing objects: 100% (2224/2224)
remote: Total 73349 (delta 39481), reused 72619 (delta 39381)
Receiving objects: 100% (73349/73349), 14.30 MiB | 348 KiB/s, done.
Resolving deltas: 100% (39939/39939), done.
Note: checking out 'b465b6d753559781bcc44adf1077b09d820d09a5'.
```

Needed for getting Gerrit plug-in APIs archetypes and sources (NOT yet on Maven public repos)

Step-2: build Gerrit plug-in archetype

GerritForge™

```
$ cd gerrit/gerrit-plugin-archetype && mvn install && cd ../..
[INFO] ------------------------------------------------------------------------
[INFO] Building Gerrit Code Review - Plugin Archetype 2.5-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[...lots of Maven rubbish ...]
[INFO] Installing /Users/lucamilanesio/GerritUserSummit/gerrit/gerrit-
plugin-archetype/target/gerrit-plugin-archetype-2.5-SNAPSHOT.jar to
/Users/lucamilanesio/.m2/repository/com/google/gerrit/gerrit-plugin-
archetype/2.5-SNAPSHOT/gerrit-plugin-archetype-2.5-SNAPSHOT.jar
[INFO] Installing /Users/lucamilanesio/GerritUserSummit/gerrit/gerrit-
plugin-archetype/pom.xml to
/Users/lucamilanesio/.m2/repository/com/google/gerrit/gerrit-plugin-
archetype/2.5-SNAPSHOT/gerrit-plugin-archetype-2.5-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```

Step-3: create your plug-in project

GerritForge™

```
$ mvn archetype:generate -DarchetypeGroupId=com.google.gerrit \
    -DarchetypeArtifactId=gerrit-plugin-archetype \
    -DarchetypeVersion=2.5-SNAPSHOT \
    -DgerritApiVersion=2.5 \
    -DgroupId=com.gerritforge -Dversion=1.0-SNAPSHOT \
    -DImplementation-Url=http://gerritforge.com \
    -DImplementation-Vendor=GerritForge \
    -DartifactId=plugin10mins -DpluginName=plugin10mins
[...]
Confirm properties configuration:
[...]
gerritApiType: plugin
gerritApiVersion: 2.5-SNAPSHOT
pluginName: plugin10mins
 Y: :
[INFO] ------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------
```

Step-4: build and install plugin

GerritForge™

```
$ cd plugin10mins && mvn package && cp target/plugin10mins*jar
$GERRIT_SITE/plugins/plugin10mins.jar
[INFO] ------------------------------------------------------------------------
[INFO] Building plugin10mins 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[...lots of Maven rubbish ...]
[INFO] Building jar:
/Users/lucamilanesio/GerritUserSummit/plugin10mins/target/plugin10mins-
1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```

Step-5: Ready to Go ☺ *… starter is served*

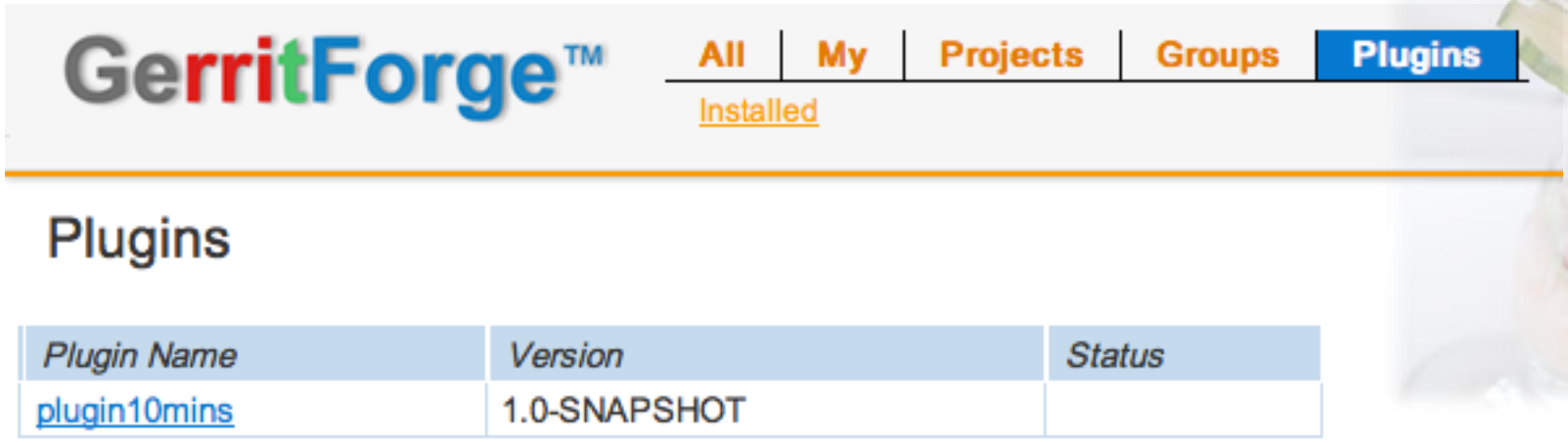
GerritForge™

Gerrit plugins hot-deploy

- No need to restart: plugins are automatically scanned and loaded

```
$ tail -1 $GERRIT_SITE/logs/error_log
[2012-11-09 12:05:33,838] INFO
com.google.gerrit.server.plugins.PluginLoader : Loaded plugin plugin10mins
```

http://<<GerritURL>>/#/admin/plugins/



First course: let's add some initial code

GerritForge™

First goal: adding HTTP/HTML features

- New Server-side feature: show Gerrit server date/time

Gerrit HTTP plug-ins have the ability to:

- Define new HTTP Servlets
- Map Servlets to plugin URLs paths automatically
- Define static resources mapped over HTTP
  (i.e. CSS, images, other)

Step-1: Disable Guice modules in MF

GerritForge™

Plug-in archetype has created Guice Modules

➔We will use auto-registration: comment out modules in pom.xml

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>2.4</version>
      <configuration>
        <archive>
          <manifestEntries>
            <!--
            <Gerrit-Module>com.gerritforge.Module</Gerrit-Module>
             <Gerrit-SshModule>com.gerritforge.SshModule</Gerrit-SshModule>
             <Gerrit-HttpModule>com.gerritforge.HttpModule</Gerrit-HttpModule>
             -->
```

Step-2: Define the ServerTime servlet

GerritForge™

```
package com.gerritforge;

import java.io.IOException;

@Export("/servertime")
@Singleton
public class ServerTime extends HttpServlet {
  private static final long serialVersionUID = -8809144980777379332L;

  @Override
  protected void doGet(HttpServletRequest req, HttpServletResponse resp)
      throws ServletException, IOException {
    PrintWriter out = resp.getWriter();
    try {
      out.write(String.format(
          "<html>" +
          "<head>" +
          "<link rel=\"stylesheet\" type=\"text/css\" href=\"static/bootstrap.css\">" +
          "</head>" +
          "<body>" +
          "<h1>Server current time is: </h1>" +
          "<h2>%s</h2>" +
          "</body>", new Date()));
    } finally {
      out.close();
    }
  }
}
```

Automatically register this Servlet to <gerrit-url>/plugins/<plugin-name>/servertime

Plug-ins inherit Gerrit class loader: Servlet-API is covered by PlugIn API

Step-3: add static resources

GerritForge™

We use bootstrap.css in our generated HTML page

➔/static folder is automatically managed by Gerrit and mapped to <<gerrit-url>>/plugins/<plugin-name>/static

```
$ mkdir src/main/resources/static/

$ curl http://gitblit.com/bootstrap/css/bootstrap.css \
  -o src/main/resources/static/bootstrap.css
 % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 99896  100 99896    0     0  49384      0  0:00:02  0:00:02 --:--:--
98033
```

Step-4: Rebuild and install

GerritForge™

```
$ cd mvn package && cp target/plugin10mins*jar
$GERRIT_SITE/plugins/plugin10mins.jar
[...lots of Maven rubbish ...]
[INFO] Building jar:
/Users/lucamilanesio/GerritUserSummit/plugin10mins/target/plugin10mins-1.0-
SNAPSHOT.jar
[INFO] ------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------
```

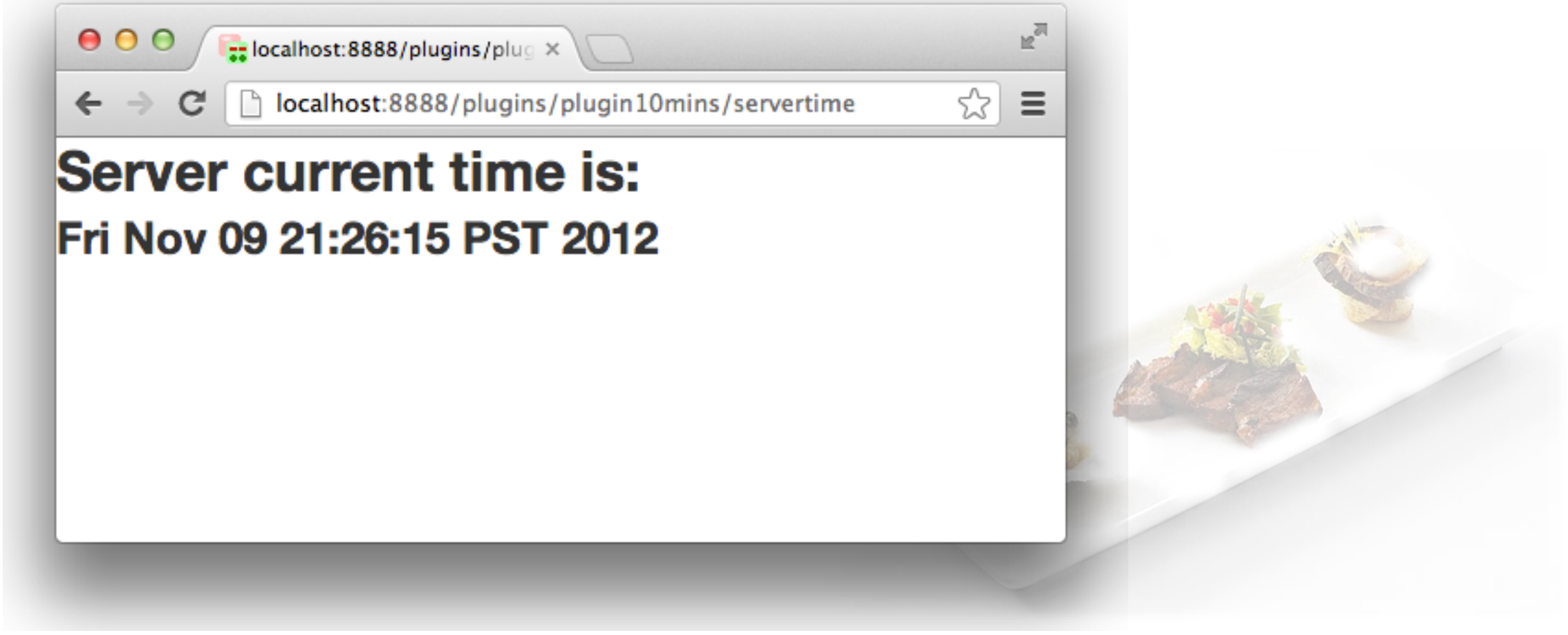
Gerrit automatically scans for changes in $GERRIT_SITE/plugins: wait for your plugin getting reloaded

```
$ tail -f -1 $GERRIT_SITE/logs/error_log
[2012-11-09 15:49:09,082] INFO
com.google.gerrit.server.plugins.PluginLoader : Reloading plugin
plugin10mins
```

Step-5: First course is on the table ☺

GerritForge™

`http://<<GerritURL>>/plugins/plugin10mins/servertime`



Main course now: get injected by Gerrit

GerritForge™

Let's make it useful: extend Gerrit functionality

- Get injected the same objects available in Gerrit *(well, almost)*
- Access Gerrit DB and JGit backend
- Display my commit from a repository

Our HTTP plug-in will have now the ability to:

- Current user identity
- Check if user has access that the repository
- Open the repository and get latest commit

Step-1: New Servlet "LastCommit"

GerritForge™

Define a new Servlet mapped to /lastcommit path

```
package com.gerritforge;

import java.io.IOException;

@Export("/lastcommit")
@Singleton
public class LastCommit extends HttpServlet {
  private static final long serialVersionUID = 20941779937039872671;

  @Override
  protected void doGet(HttpServletRequest req, HttpServletResponse resp)
      throws ServletException, IOException {
    PrintWriter out = resp.getWriter();
    try {
      out.print(String
          .format("<html>"
              + "<head>"
              + "<link rel=\"stylesheet\" type=\"text/css\" href=\"static/bootstrap.css\">"
              + "</head>" + "<body>" + "I'm alive" + "</body></html>"));
    } finally {
      out.close();
    }
  }
}
```

Step-2: Gerrit injected objects

GerritForge™

We need access to accounts, project access and Git repositories
➔ let's get them injected in the constructor

```
private final GitRepositoryManager repoManager;
private final ProjectControl.Factory projControlFactory;
private final AccountCache accounts;

@Inject
public LastCommit(final ProjectControl.Factory projControlFactory,
    final GitRepositoryManager repoManager, final AccountCache accounts) {
  this.projControlFactory = projControlFactory;
  this.repoManager = repoManager;
  this.accounts = accounts;
}
```

Step-3: Get project and check access

GerritForge™

Get project name in 'p' Servlet URL parameter; Gerrit ProjectControl provides access control info

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
  PrintWriter out = resp.getWriter();
  try {
    NameKey projNameKey = NameKey.parse(req.getParameter("p"));
    ProjectControl projControl = projControlFactory.controlFor(projNameKey);
    if (!projControl.isVisible()) {
      resp.sendError(HttpURLConnection.HTTP_FORBIDDEN, "Access to project "
          + projNameKey + " is DENIED");
    }
    out.print(String
        .format("<html>"
            + "<head>"
            + "<link rel=\"stylesheet\" type=\"text/css\" href=\"static/bootstrap.css\">"
            + "</head>" + "<body>" + "I'm alive" + "</body></html>"));
  } catch (NoSuchProjectException e) {
    resp.sendError(HttpURLConnection.HTTP_NOT_FOUND);
  } finally {
    out.close();
  }
}
```

Step-4: Get my current identity

GerritForge™

We get the current Gerrit user and display his accounts details (user full name and preferred e-mail)

```
Account currentUser =
    accounts.getByUsername(projControl.getCurrentUser().getUserName())
        .getAccount();

out.print(String.format(
    "<html>"
        + "<head>"
        + "<link rel=\"stylesheet\" type=\"text/css\" href=\"static/bootstrap.css\">"
        + "</head>" + "<body>" + "<h2>Current user</h2>"
        + "<h3>%s &lt;%s&gt;</h3>", currentUser.getFullName(),
    currentUser.getPreferredEmail()));

out.print("</body></html>");
```

Step-5: Get latest commit from repo

GerritForge™

Using Gerrit's own repo manager, we get our last commit in the repo

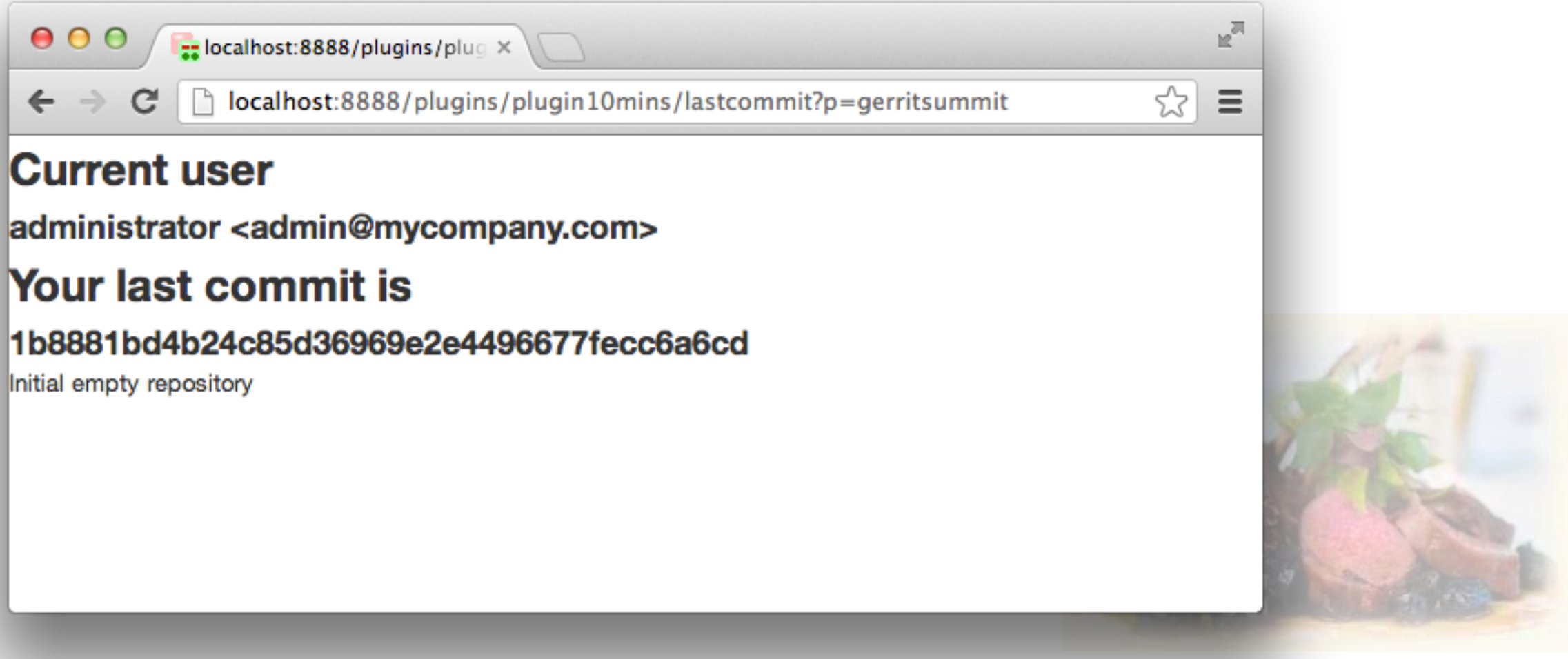
```
private RevCommit getLastCommit(NameKey projNameKey, Account account)
    throws RepositoryNotFoundException, IOException {
  Repository repo = repoManager.openRepository(projNameKey);
  RevWalk revWalk = new RevWalk(repo);
  String userFullName = account.getFullName();
  String userEMail = account.getPreferredEmail();

  try {
    revWalk.sort(RevSort.COMMIT_TIME_DESC);
    revWalk.markStart(revWalk
        .parseCommit(repo.getRef("master").getObjectId()));
    for (RevCommit revCommit : revWalk) {
      PersonIdent author = revCommit.getAuthorIdent();
      if (author.getName().equalsIgnoreCase(userFullName)
          && author.getEmailAddress().equalsIgnoreCase(userEMail)) {
        return revCommit;
      }
    }
    return null;
  } finally {
    revWalk.release();
    repo.close();
  }
}
```

Main course is served ☺

GerritForge™

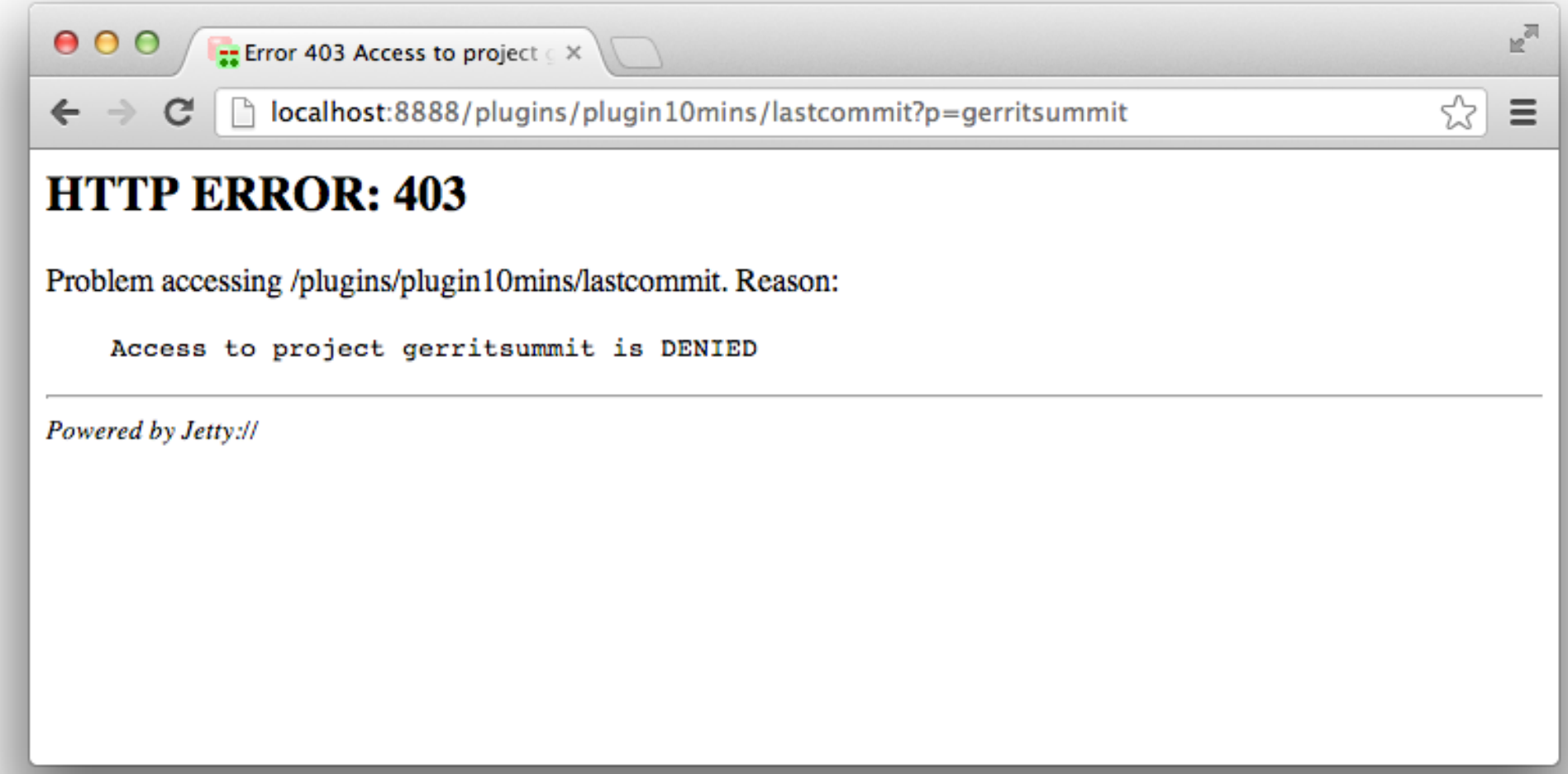
`http://<<GerritURL>>/plugins/plugin10mins/lastcommit?p=gerritsummit`



Does it really work with Gerrit security ?

GerritForge™

Logging out from Gerrit and coming back to the plug-in page …



There's much more ...

GerritForge™

Gerrit plug-ins are much more than just Servlet ☺

- Provide your own SSH commands
- Listen to Gerrit events for audit-trail purposes
- Implement Gerrit group backends

... and much more coming over in Gerrit 2.6 !!!!

Plugins are just "dynamic" Guice Modules

- Gerrit load and bind plug-ins modules
- No modules specified ➔ self-discovery (the one we used)

Step-1: New SSH command

GerritForge™

Create a new SshCommand exported as "servertime" command

```
package com.gerritforge;

import java.util.Date;

import com.google.gerrit.extensions.annotations.Export;
import com.google.gerrit.sshd.SshCommand;

@Export("servertime")
public class SshServerTime extends SshCommand {

  @Override
  protected void run() throws UnloggedFailure, Failure, Exception {
    stdout.println("Current date/time: " + new Date());
  }
}
```

Automatically assigned to remote command: <plugin-name> servertime

Step-2: Gerrit injected objects

GerritForge™

Similarly to HTTP plug-in, we can inject Gerrit objects
➔ get current connected user identity

```
package com.gerritforge;

import java.util.Date;

@Export("servertime")
public class SshServerTime extends SshCommand {

  private Account account;

  @Inject
  public SshServerTime(final CurrentUser currentUser,
      final AccountCache accountCache) {
    this.account =
        accountCache.getByUsername(currentUser.getUserName()).getAccount();
  }

  @Override
  protected void run() throws UnloggedFailure, Failure, Exception {
    stdout.println("Current date/time: " + new Date());
    stdout.println("Current user: " + account.getFullName() + " <"
        + account.getPreferredEmail() + ">");
  }
}
```

Et voilà le dessert !

GerritForge™

```
$ mvn package && cp target/plugin10mins*jar
$GERRIT_SITE/plugins/plugin10mins.jar
[...lots of Maven rubbish ...]
[INFO] Building jar:
/Users/lucamilanesio/GerritUserSummit/plugin10mins/target/plugin10mins-
1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------
```

... wait for hot-deploy ... and your new plugin command is there !

```
$ ssh -p 29418 administrator@localhost plugin10mins servertime
Current date/time: Fri Nov 09 21:08:41 PST 2012
Current user: administrator <admin@mycompany.com>
```

Still hungry ?

GerritForge™

Making a group system plugin *(Colby Ranger, Google)*

Trombone room – Saturday Nov 10th - 11:30 AM

What's coming in Gerrit 2.6 *(Shawn Pearce, 2.6 Lead)*

Tambourine room – Sunday Nov 11th – 9:45 AM

JavaScript plugins *(Dariusz Łuksza, CollabNet)*

Trombone room – Sunday Nov 11th – 1:30 PM

You can see these slides again at: **www.slideshare.net/lucamilanesio**