

Apache Felix Web Console

Carsten Ziegeler | chiegeler@apache.org

ApacheCon NA 2014



- RnD Team at Adobe Research Switzerland
- Member of the Apache Software Foundation
 - Apache Felix and Apache Sling (PMC and committer)
 - And other Apache projects
- OSGi Core Platform and Enterprise Expert Groups
- Member of the OSGi Board
- Book / article author, technical reviewer, conference speaker

Problem Description

- Manage an OSGi based application
- Remotely
- Easy of use
- Extensibility
- Offline support

Various Solutions

- JMX
- Text based
 - Apache Felix Shell, Apache Felix Gogo Shell
 - Knopflerfish Shell
 - Equinox Console
 - OSGi RFC 147 (Command Line Interface)
- GUI based
 - Knopflerfish Desktop (Swing)
 - **Apache Felix Web Console**

Apache Felix Web Console

- Rich set of core functionality
 - Bundles
 - Services
 - Configuration Admin
 - System Information
- JQuery based UI
- Extensible
- Pluggable authentication
- Still light-weight

Installation

- Dependencies
 - OSGi Http Service
 - Commons IO 1.4 (*)
 - Commons File Upload 1.2 (*)
 - org.json (.e.g. Apache Geronimo Bundles: json-20090211) (*)
- Two flavors: bundle with and without above marked dependencies
- Additional optional dependencies
- Additional plugins

Demo

Extension Points

- Branding
- Translations
- Security Provider
- Plugins
- Inventory Printer (Configuration Printer)

- Service Interface: BrandingPlugin
- Default Branding: DefaultBrandingPlugin
- Branding Fragment
 - /META-INF/webconsole.properties
webconsole.brand.name = Apache Sling Web Console
webconsole.product.name = Apache Sling
webconsole.product.url = <http://sling.apache.org>
webconsole.product.image = /res/sling/logo.png
webconsole.favicon = /res/sling/favicon.ico
- Sample
 - Sling Web Console Branding Plugin

Translations

- Out-of-the-box
 - Bulgarian, English, German, Russian
- Extensible
 - Fragment to the Web Console
 - Based on Standard Bundle Localization
 - OSGI-INF/l10n/bundle*.properties

Web Console Security

- Secure access to the web console!
 - Consider blocking internet access
- Configure authentication

Security Provider

- Service interface: WebConsoleSecurityProvider
 - HTTP Basic Authentication Based
 - Validates Username / Password
 - Access Control (hook only)
- Service Interface: WebConsoleSecurityProvider2
 - Extends WebConsoleSecurityProvider
 - Flexible Authentication (implementing HttpContext.handleSecurity)

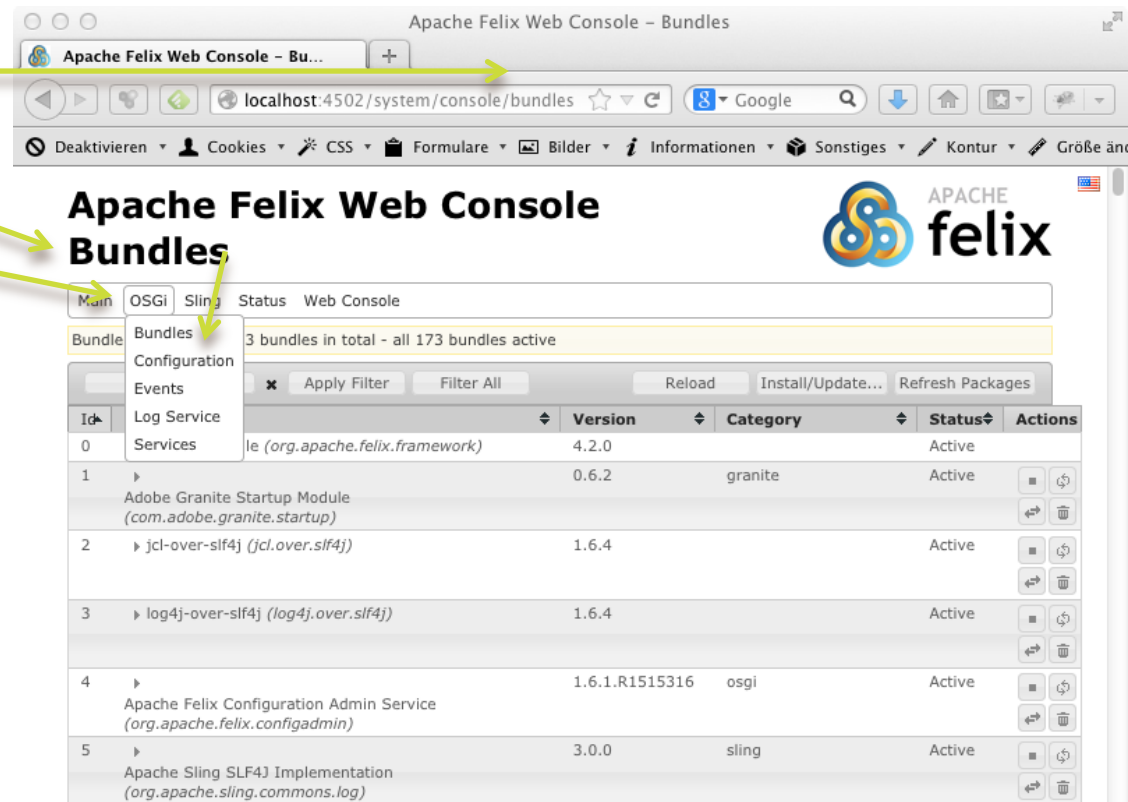
- Implementations
 - Default: Single configurable User
 - Apache Karaf: JAAS based authentication
 - Apache Sling
 - Authenticates against JCR Repository
 - Supports Sling Authentication Setup

Security Provider

- WebConsoleSecurityProvider, WebConsoleSecurityProvider2
- Web Console uses them for authentication
- Methods for authorization not used atm
- Authorization is on the road map for next version!

Plugins

- javax.servlet.Servlet service
- Optional extend AbstractWebConsolePlugin or SimpleWebConsolePlugin
- Service properties
 - felix.webconsole.label
 - felix.webconsole.title
 - felix.webconsole.category



Sample Plugin #1

```
@Component
@Service(value=HttpServlet.class)
@Properties({
    @Property(name="felix.webconsole.label", value="sample"),
    @Property(name="felix.webconsole.title",
        value="Sample Plugin"),
    @Property(name="felix.webconsole.category",
        value="My App")
})
public class SamplePlugin extends HttpServlet {

    @Override
    public void service(ServletRequest rq, ServletResponse rs)
        throws ServletException, IOException {
        rs.getWriter().println("Hello World...");
    }
}
```

Sample Plugin #2

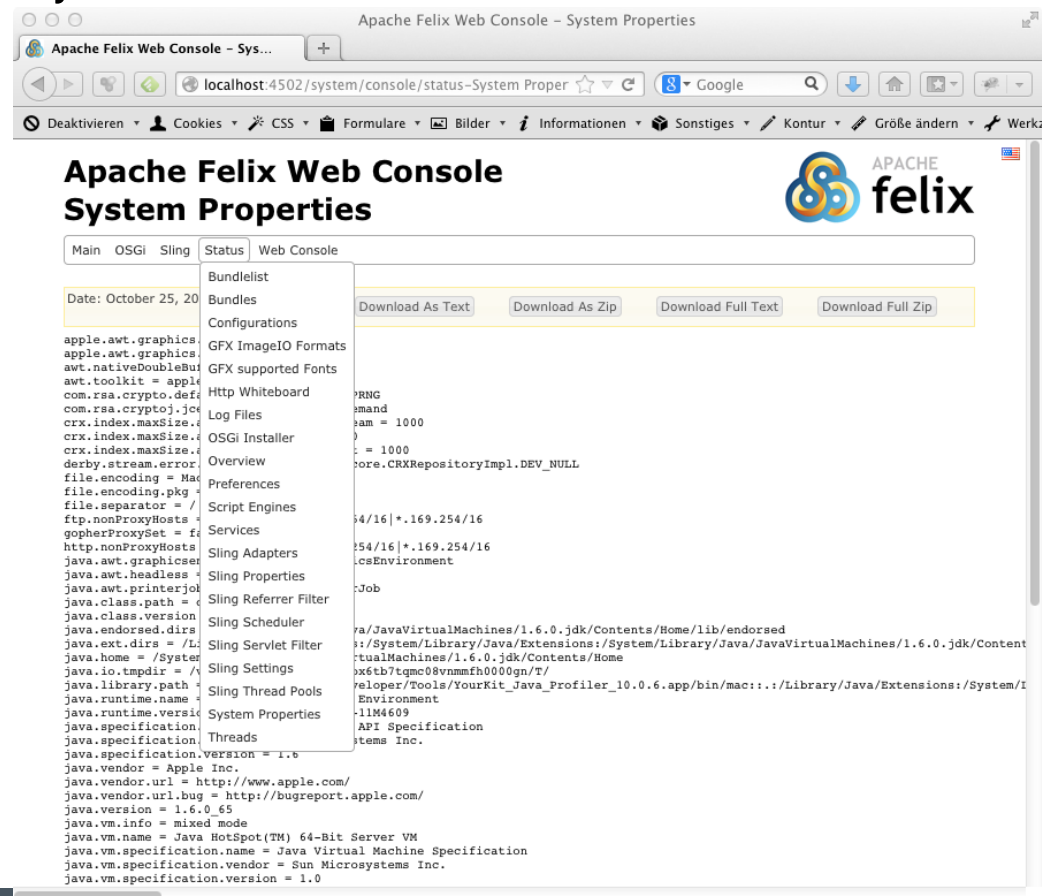
```
public class Sample2 extends SimpleWebConsolePlugin {  
    public Sample2(BundleContext ctx) {  
        super("sample2", "Sample Plugin #2", null);  
        register(ctx);  
    }  
  
    @Override  
    protected void renderContent(  
        HttpServletRequest req,  
        HttpServletResponse res)  
        throws IOException {  
        res.getWriter().println("Hi There ...")  
    }  
}
```

Plugins

- Rendering: `service()`, `doGet()` or `renderContent()`
- Interaction: `service()`, `doPost()`
- UI: jQuery
- Add own js, css as resources
- Simple templating support

Apache Felix Inventory

- OSGi service providing status information
 - Bundle list, services, system properties etc.
- Different formats: text, html, json
- Optional attachments
- Direct support in web console



Inventory Printer Sample

```
@Component
@Service(value={InventoryPrinter.class})
@Properties({
    @Property(name=InventoryPrinter.NAME, value="slingjobs"),
    @Property(name=InventoryPrinter.TITLE, value="Sling Jobs"),
    @Property(name=InventoryPrinter.FORMAT,
        value={"TEXT", "JSON"})
})
public class InventoryPlugin implements InventoryPrinter {

    public void print(final PrintWriter pw,
        final Format format, final boolean isZip) {
        // print something
    }
}
```

Attachments: ZipAttachmentProvider

Offline Usage

- Status zip contains
 - Output from all inventory printers
 - Output from special web console plugins
 - Attachments
 - JSON files
- Automated analysis
 - Bundle list and status
 - Configurations
 - Log files

Demo

- Today: Simple REST API
 - Bundles
 - Config Admin
 - <http://felix.apache.org/documentation/subprojects/apache-felix-web-console/web-console-restful-api.html>

- RFC 182 - REST Interface for OSGi
- Definition of REST API for
 - Framework status
 - Framework operations
- Bundle states
- Bundle operations
- Service states
- Service operations

- RFC 182 - REST Interface for OSGi
- Extension mechanism
 - Config Admin?
- Java client
- Javascript client

Apache Felix Web Console

- Flexible and extensible administration console
 - Web based
- Many plugins available
- Offline support
- Planned major task
 - Authorization
 - REST support