Java Servlet 3.0

Rajiv Mordani
Spec Lead

Overview

JCP

- Java Servlet 3.0 API – JSR 315
- 20 members
  - Good mix of representation from major Java EE vendors, web container developers and web framework authors

Overview

- Main areas of focus
  - Ease of Development
  - Extensibility
  - Asynchronous support
  - Security
- Final release available now as part of Java EE 6 and GlassFish v3

Ease of Development

Overview

- Enhance API using new language features in introduced since J2SE 5.0
  - Annotations for declarative style of programming
  - Generics for type safety in API without breaking backwards compatibility
- Better defaults and convention over configuration

Ease of Development

Annotations

- Annotations to declare Servlets, Filters, Listeners and Security constraints
  - `@WebServlet` – Defines a Servlet
  - `@WebFilter` – Defines a Filter
  - `@WebListener` – Defines a listener
  - `@WebInitParam` – Defines an init param
  - `@ServletSecurity` – security constraints
  - `@MultipartConfig` – file upload
- Use web.xml to override values

Ease of Development

Use of annotations

- `@WebServlet` - For defining a Servlet
- MUST specify url mapping
- All other attributes optional with reasonable defaults
  - For example, the default name of a Servlet is the fully qualified class name
- Must still extend **`HttpServlet`** to derive method contracts for doGet, doPost and other methods

Ease of Development

Servlet 2.5 example

```
/* Code in Java Class */

package com.foo;
public class MyServlet extends
HttpServlet {
public void
doGet(HttpServletRequest
req,HttpServletResponse res)

{

…

}



…

}
```

```
<!--Deployment descriptor web.xml
   -->

<web-app>

   <servlet>
       <servlet-name>MyServlet

       </servlet-name>
      <servlet-class>
        com.foo.MyServlet
      </servlet-class>
   </servlet>
   <servlet-mapping>
      <servlet-name>MyServlet
      </servlet-name>
      <url-pattern>/myApp/*
      </url-pattern>
   </servlet-mapping>
   ...
</web-app>
```

Ease of Development

Servlet 3.0 example

```
package com.foo;
@WebServlet(name="MyServlet", urlPattern="/myApp/*")
public class MyServlet extends HttpServlet {
   public void doGet(HttpServletRequest req,
                     HttpServletResponse res)
   {
      ...
   }
```

Extensibility

Adding frameworks

- Enable use of frameworks without boiler plate configuration in deployment descriptors
  - Put the burden on framework developer
- Dynamic registration of Servlets and Filters using programmatic configruation APIs
- Modularize **`web.xml`** to allow frameworks to be self-contained
- Use of annotations enables extensibility as well

Dynamic Registration of Servlets and Filters

Register

- Performed during **`ServletContext`** initialization
- **`ServletContext.add[Servlet|Filter]`**
- Overloaded versions take [Servlet|Filter] name and
  - Fully qualified class [Servlet|Filter] class name
  - **`Class<? extends [Servlet|Filter]>`** OR
  - [Servlet|Filter] instance

Dynamic Registration of Servlets and Filters

Create and Register

- Use returned **`Registration`** handle to configure all aspects of [Servlet|Filter]
- **`ServletContext.create[Servlet|Filter]`**
  - Takes **`Class<? extends [Servlet|Filter]>`**
  - Container responsible for instantiating [Servlet|Filter]
  - Returned [Servlet|Filter] may be customized before it is registered via the **`ServletContext.add[Servlet|Filter]`** method

Dynamic Registration of Servlets and Filters

Lookup

- **`ServletContext.find[Servlet|Filter]Registration`**
  - Takes [Servlet|Filter] name as argument
  - Returned **`Registration`** handle provides subset of configuration methods
  - May be used to configure [Servlet|Filter]
  - Conflicts returned as **`java.util.Set`**

Dynamic Registration of Servlets and Filters

Register example

```
ServletRegistration.Dynamic dynamic =

servletContext.addServlet("DynamicServlet","com.mycom.MyServlet");
dynamic.addMapping("/dynamicServlet");
dynamic.setAsyncSupported(true);
```

Dynamic Registration of Servlets and Filters

Lookup example

```
ServletRegistration declared =

servletContext.getServletRegistration("DeclaredServlet");
declared.addMapping("/declaredServlet");
declared.setInitParameter("param",
"value");
```

Extensibility

`web-fragment.xml`

- **`web-fragment.xml`** – partial web.xml
- Bundled in framework jar file in **`META-INF`** directory
- Container discovers and assembles the effective descriptor
- Almost identical to **`web.xml`** – a few ordering related elements are different
- Only JAR files in **`WEB-INF/lib`** are considered

Extensibility

web-fragment.xml example

```
<web-fragment>
  <servlet>
    <servlet-name>welcome</servlet-name>
    <servlet-class>
      WelcomeServlet
    </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>welcome</servlet-name>
    <url-pattern>/Welcome</url-pattern>
  </servlet-mapping>
...
</web-fragment>
```

Extensibility

Ordering

- Compatible with JavaServer Faces
- Fragment identified by **`<name>`**
- **`web.xml`** may declare absolute ordering of fragments via **`<absolute-ordering>`**
- Fragments may declare ordering preferences relative to other fragments via **`<ordering>`** with nested **`<before>`** and **`<after>`**
  - Ignored if **`<absolute-ordering>`** specified

Extensibility

`ServletContainerInitializer`

- Provided by apps or the container
- Discovered using the service provider API in the Java runtime
- Expresses interest in classes via **`@HandlesTypes`**
- Container scans webapp for classes that match **`@HandlesTypes`** and passes them to the **`ServletContainerInitializer`**, along with **`ServletContext`**

Extensibility

`ServletContainerInitializer`

- **`ServletContainerInitializer`** inspects passed set of classes and may register Servlets / Filters / Listeners based on them.
- Mojarra (JSF implementation) plugged into GlassFish v3 using this mechanism

Extensibility

Resource sharing

- Static and JavaServer Pages (JSP) resources no longer confined to document root of web application
- May be placed inside **`WEB-INF/lib/[*.jar]/META-INF/resources`**
- Resources in document root take precedence over those in bundled JAR files

Asynchronous Support

Overview

- Useful for Comet, long waits
- Must declare `@WebServlet(asyncSupported=true)`
- Then Call

```
AsyncContext ctx = ServletRequest.startAsync(req, res)
```

- AsyncContext can then either:

```
dispatch(String path)
start(Runnable action)
```

- Then paired with `complete()`

Asynchronous API

ServletRequest

- **`ServletRequest.isAsyncSupported()`**
  - True if ALL [Filter | Servlet]s support async in
    - The Filter chain
    - The Request dispatch chain
- Can be configured
  - web.xml
  - Annotation
  - API

Asynchronous API

ServletRequest

- **`AsyncContext ServletRequest.startAsync`**
  - Called by Servlet | Filter
  - Response not committed on return of
    - service method
    - Filter chain

Asynchronous API

AsyncContext

- **`AsyncContext.dispatch`**
  - Called by asynchronous handler
  - Schedule async dispatch
    - DispatcherType.ASYNC
  - Response generated by [Servlet | Filter] using
    - Container thread pool
    - JSP, JSF or other frameworks

Asynchronous API

AsyncContext

- **`AsyncContext.complete`**
  - Called by asynchronous handler or container
  - Signals Response has been generated

Security

Programmatic login / logout

- Support for programmatic authentication, login and logout
  - **`HttpServletRequest.authenticate`**
  - **`HttpservletRequest.login`**
  - **`HttpServletRequest.logout`**
- Annotations for declaring http constraints for resources
  - @ServletSecurity

Servlet 3.0

Summary

- Annotations for ease of development
- Optional web.xml
- Better defaults
- Web Framework pluggability
- Asynchronous Processing
- Security enhancements

Resources

Java EE 6 and GlassFish v3

Java EE 6 Home

http://java.sun.com/javaee

Java EE 6 Downloads

http://java.sun.com/javaee/downloads/index.jsp

Upcoming Training

http://java.sun.com/javaee/support/training/

Sun GlassFish Enterprise Server v3 Home

http://www.sun.com/glassfishv3

Community Page

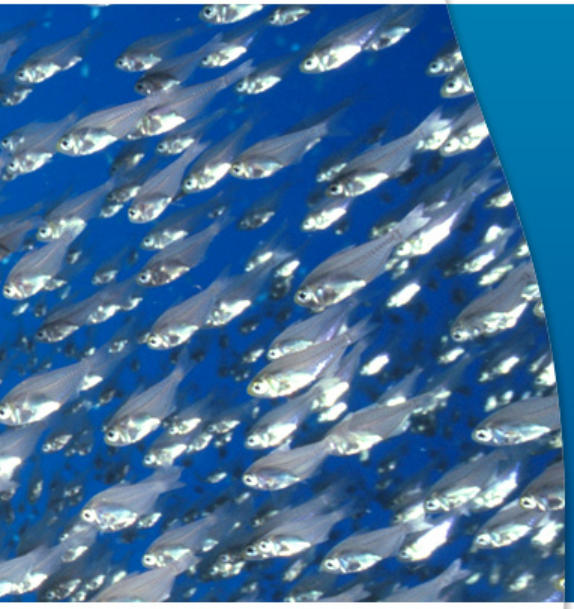
glassfish.org

White Papers/Webinars

http://www.sun.com/glassfish/resources

Java EE 6

GlassFish



Thank You