RESTFul Services In Java Using Jersey

Building Your First Jersey Application

Bryan Hansen
twitter: bh5k
http://www.linkedin.com/in/hansenbryan

pluralsight
hardcore developer training

Prerequisites

Java

Maven

maven

Spring STS

springsource
TOOLSUITE

Tomcat

Apache Tomcat

What We Are Building

Service Features

Service Features

- Retrieve all activities that a user has done

Service Features

- Retrieve all activities that a user has done
- Retrieve a specific activity

Service Features

- Retrieve all activities that a user has done
- Retrieve a specific activity
- Create a new activity

Service Features

- Retrieve all activities that a user has done
- Retrieve a specific activity
- Create a new activity
- Update an activity

Service Features

- Retrieve all activities that a user has done
- Retrieve a specific activity
- Create a new activity
- Update an activity
- Delete an activity

Tomcat

Tomcat

- RESTFul Services are ran out of a basic war file

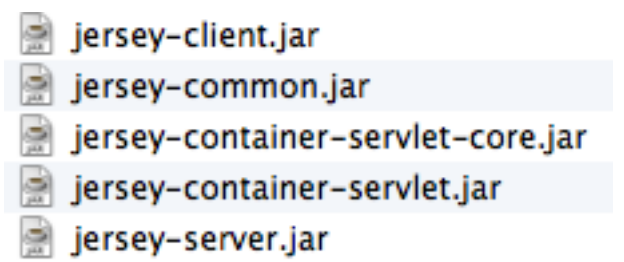
Tomcat

- RESTFul Services are ran out of a basic war file
- Tomcat is a very lightweight web container

Tomcat

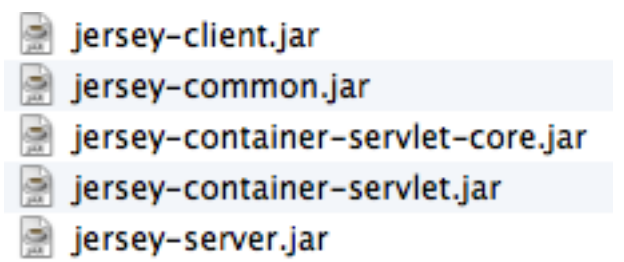
- RESTFul Services are ran out of a basic war file
- Tomcat is a very lightweight web container
- http://tomcat.apache.org/download-70.cgi

Downloading Jersey



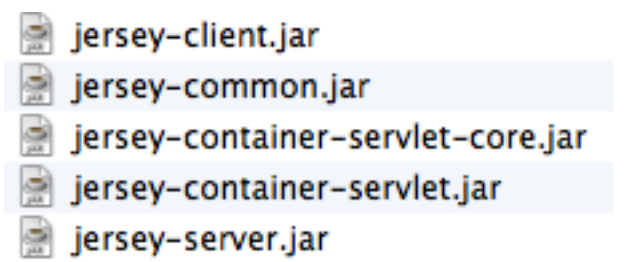
Downloading Jersey

- **As recommended on Jersey's website, we are going to use Maven**



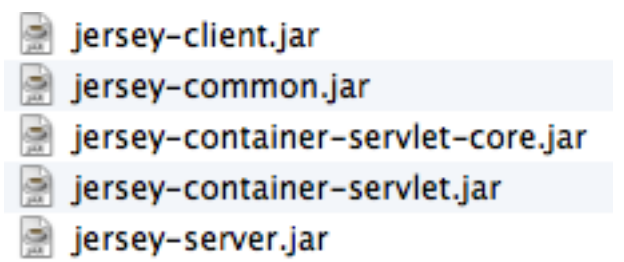
Downloading Jersey

- **As recommended on Jersey's website, we are going to use Maven**
  - http://pluralsight.com/training/Courses/TableOfContents/maven-fundamentals



Downloading Jersey

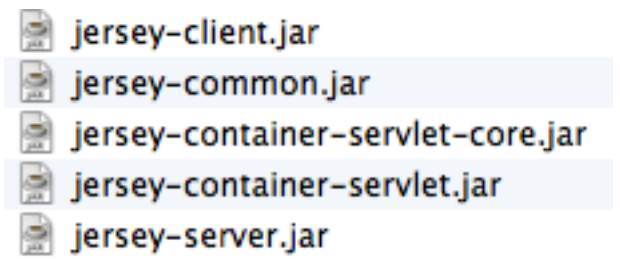
- **As recommended on Jersey's website, we are going to use Maven**
  - http://pluralsight.com/training/Courses/TableOfContents/maven-fundamentals
- **Jersey has an Archetype to help start us off:**



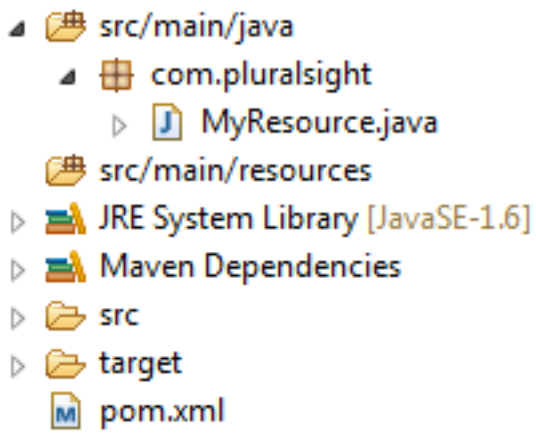
Downloading Jersey

- **As recommended on Jersey's website, we are going to use Maven**
  - http://pluralsight.com/training/Courses/TableOfContents/maven-fundamentals
- **Jersey has an Archetype to help start us off:**

```
mvn archetype:generate -DarchetypeGroupId=org.glassfish.jersey.archetypes \
    -DarchetypeArtifactId=jersey-quickstart-webapp -DarchetypeVersion=2.2
```

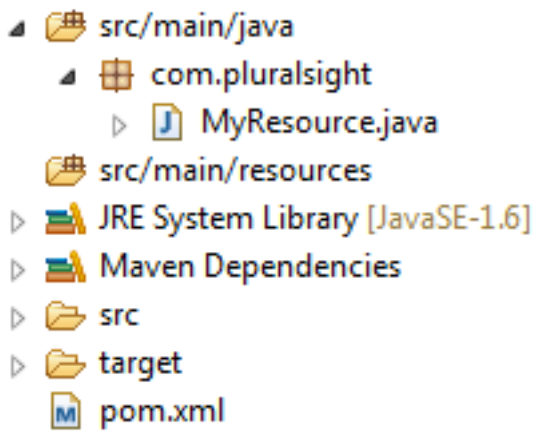


Project Layout



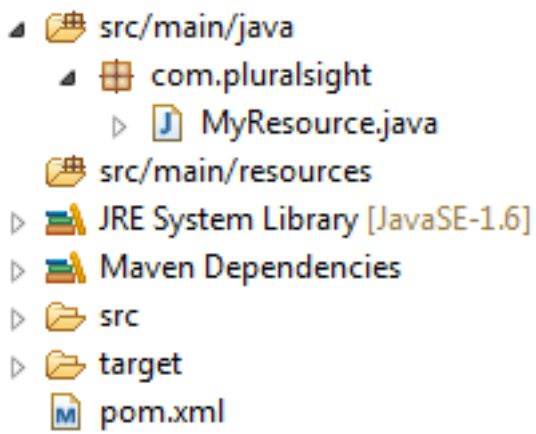
Project Layout

- **Standard Maven structure**



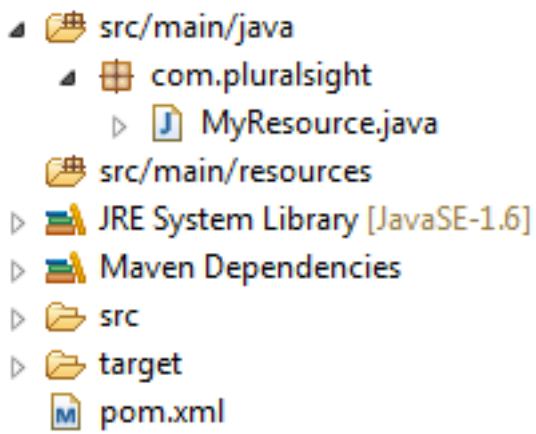
Project Layout

- **Standard Maven structure**
  - src/main/java



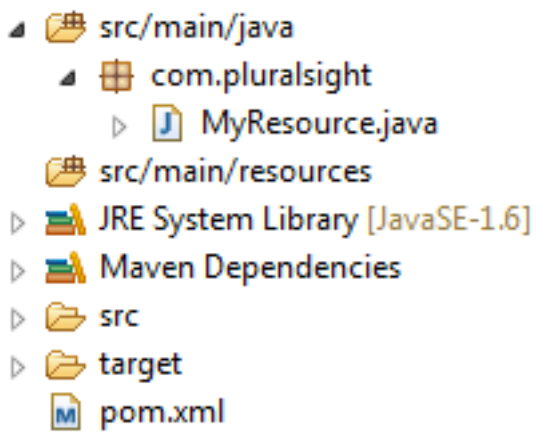
Project Layout

- **Standard Maven structure**
  - src/main/java
  - src/main/resources



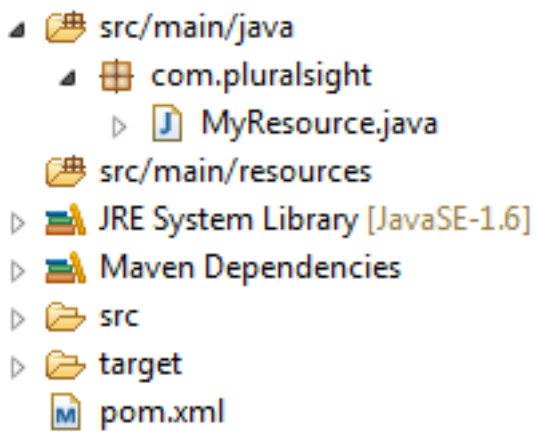
Project Layout

- **Standard Maven structure**
  - src/main/java
  - src/main/resources
  - src/main/webapp



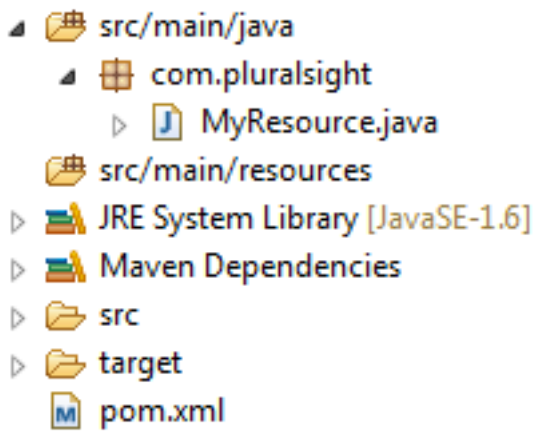
Project Layout

- **Standard Maven structure**
  - src/main/java
  - src/main/resources
  - src/main/webapp
- **Libs are pulled in through Maven in the dependencies section**



Project Layout

- **Standard Maven structure**
  - src/main/java
  - src/main/resources
  - src/main/webapp
- **Libs are pulled in through Maven in the dependencies section**
- **web.xml is already configured with Jersey servlet**



web.xml

web.xml

- src/main/webapp/WEB-INF/web.xml

web.xml

- **src/main/webapp/WEB-INF/web.xml**

```
<servlet>
    <servlet-name>Jersey Web Application</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
    <init-param>
        <param-name>jersey.config.server.provider.packages</param-name>
        <param-value>com.pluralsight</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
```

web.xml

- **src/main/webapp/WEB-INF/web.xml**

```
<servlet>
    <servlet-name>Jersey Web Application</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
    <init-param>
        <param-name>jersey.config.server.provider.packages</param-name>
        <param-value>com.pluralsight</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
```

web.xml

- **src/main/webapp/WEB-INF/web.xml**

```
<servlet>
    <servlet-name>Jersey Web Application</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
    <init-param>
        <param-name>jersey.config.server.provider.packages</param-name>
        <param-value>com.pluralsight</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>



<servlet-mapping>
    <servlet-name>Jersey Web Application</servlet-name>
    <url-pattern>/webapi/*</url-pattern>
</servlet-mapping>
```

web.xml

- **src/main/webapp/WEB-INF/web.xml**

```
<servlet>
    <servlet-name>Jersey Web Application</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
    <init-param>
        <param-name>jersey.config.server.provider.packages</param-name>
        <param-value>com.pluralsight</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
```

```
<servlet-mapping>
    <servlet-name>Jersey Web Application</servlet-name>
    <url-pattern>/webapi/*</url-pattern>
</servlet-mapping>
```

Creating a Service

Creating a Service

- **The most basic service needs three things:**

Creating a Service

- **The most basic service needs three things:**
  - @Path - the path to the service

Creating a Service

- **The most basic service needs three things:**
  - @Path - the path to the service
  - @GET - the request method

Creating a Service

- **The most basic service needs three things:**
  - @Path - the path to the service
  - @GET - the request method
  - @Produces - the response type from the service

Creating a Service

- **The most basic service needs three things:**
  - @Path - the path to the service
  - @GET - the request method
  - @Produces - the response type from the service

```
@Path("myresource")
public class MyResource {

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String getIt() {
        return "Got it!";
    }
}
```

Summary

Summary

- **Prerequisites**

Summary

- Prerequisites
- What our app is

Summary

- Prerequisites
- What our app is
- Installed Tomcat

Summary

- Prerequisites
- What our app is
- Installed Tomcat
- Built our first service

Summary

- Prerequisites
- What our app is
- Installed Tomcat
- Built our first service
- Ran our basic service