REST API Security Creating a Basic REST API

Guadalupe Ortiz Bellot

Computer Science and Engineering Department



- 1. Creation of a Tomcat Server in Eclipse
- 2. Web dynamic project creation
- 3. Jersey library inclusion
- 4. Package and class creation
- 5. web.xml file creation
- 6. Service deployment in the service
- 7. Testing it from Postman

- **1.** Creation of a Tomcat Server in Eclipse
- 2. Dynamic web project creation
- 3. Jersey library inclusion
- 4. Package and class creation
- 5. web.xml file creation
- 6. Service deployment in the service
- 7. Testing it from Postman

1. Creation of a Tomcat Server in Eclipse (i)

- REMINDER: it is recommended to create a folder for the workspace, for instance inside the root folder created for the software installation (Development)
- We open Eclipse and select the workspace where we will store the created projects.
- We create a Tomcat server instance: File→ New→ Other→ Server→ Server



1. Creation of a Tomcat Server in Eclipse (ii)

- Select Tomcat 9 and click Next
- We select the folder where we have installed the Tomcat and the JRE we desire to use.
- For the JRE we have to select the installed JDK. Usually, by default, it is JRE selected. If so, follow the steps in the following slide.



1. Creation of a Tomcat Server in Eclipse (iii)

How to select the JDK: Click on *Installed JRE*

- OPTION 1. If the JDK is not in the list, follow the steps below:
 - Click on the Installed JRE→ Add→ Standard
 VM →Next
 - In the pop-up screen, select the folder where the JDK is installed. Then click on Finish
 - Now follow OPTION 2 Steps



1. Creation of a Tomcat Server in Eclipse (iv)

- OPTION 2. If the JDK is in the list:
 - Click on Installed JRE→ Select the JDK and click OK.
 - It may happen that after clicking OK in the following Windows the JRE is yet selected. You have to open the deployable list, select the JDK and Finish.

Installed IPEs			Tomcat Server		
Installed JRES	Installed JREs				
	Add, remove or edit JRE definitions. I newly created Java projects.	By default, the checked JRE is added to			
	Installed JREs:			Name:	
	Name	Location	Add	Apache Tomcat v9.0	
	Jidk1.8.0_101 (default)	C:\Program Files\Java\jdk1.8 C:\Program Files\Java\jdk1.8.0	Edit	C:\Development\apache-tomcat-9.0.0.M10	Browse
	🕅 🛋 jre1.8.0_45	C:\Program Files\Java\jre1.8.0_45	Duplicate		Download and Ins
			Remove	JRE:	
			Search	jdk1.8.0_101	 Installed JREs
	٠	,	Apply		
?		ОК	Cancel		

1. Creation of a Tomcat Server in Eclipse (v)

Start the Apache Tomcat server:

• In the down part of Eclipse, we have to click on the Server tab and there the Tomcat should be (Stopped):

Markers Properties & Servers X Markers Data Source Explorer Snippets Console
Tomcat v8.0 Server at localhost [Stopped, Synchronized]

In order to launch the server we right-click on it and select Start. We wait until the start is finished, once finished we will see that the server status is Started, Synchronized:

□ 莎 🜔 🖉 🔳 🕮 🔻 🗖

😰 Markers 🗇 Properties 👫 Servers 💥 🎬 Data Source Explorer 🚡 Snippets 📮 Console 🕞 🏇 🕥 🔊 🔳 弾 <table-cell> 🖓

• NOTE: To see the servers view in Eclipse, Window \rightarrow Show View \rightarrow Server \rightarrow Servers.

- 1. Creation of a Tomcat Server in Eclipse
- 2. Web dynamic project creation
- 3. Jersey library inclusion
- 4. Package and class creation
- 5. web.xml file creation
- 6. Service deployment in the service
- 7. Testing it from Postman

2. Dynamic Web Project Creation

- We create a Dynamic Web Project named HelloWorld:
 - New \rightarrow Dynamic Web Project
 - Fill in Project name and select the Tomcat recently created as the Target runtime and → Finish
- VERY IMPORTANT: Respect upper-case and lower-case conventions for Java, specially for the first letter

File	Edit Navigate	Search	Project	Run	Wind	wol	Help	
	New		AI	t+Shift	+N ►		JPA Project	
	Open File					17	Enterprise Application Project	
	Close			Ctrl	+W	7	Dynamic Web Project	
	Close All		Ctr	l+Shift	+W	8	EJB Project	
	-			~		1	Connector Project	
	Save			Ctr	1+5	R	Application Client Project	
	Save As					3	Static Web Project	
	Save All		Ct	rl+Shit	t+S	1	Maven Project	
	Revert					1	Project	
	Move					R	Servlet	
-0	Rename				F2		Session Bean (EJB 3.x)	
8	Refresh				F5		Message-Driven Bean (EJB 3.x)	
	Convert Line Del	imiters To			•		Web Service	
Ð	Print			Ctr	I+P	2	Folder	
	Switch Workspac	e			×		File	
	Restart					2	Example	
2	Import						Other	Ctrl+N
21	Export							
	Properties			Alt+Er	nter			
	Exit							

0	New Dynamic Web Project	- 🗆 🗙
Dynamic Web Project Create a standalone Dynamic Web	p project or add it to a new or existing Enterprise Appl	lication.
Project name: HelloWorld Project location Use default location		
Location: C:\Development\won	kspace\Helloworld	Browse
Target runtime Apache Tomcat v8.0	Ŷ	New Runtime
Dynamic web module version		
3.1		~
Configuration		
Default Configuration for Apach	e Tomcat v8.0 🗸 🗸	Modify
A good starting point for workin installed to add new functionalit	g with Apache Tomcat v8.0 runtime. Additional facet y to the project.	ts can later be
EAR membership		
EAR project name: EAR	V	New Project
Working sets		
Working sets:	V	Select
?	< Back Next > Finish	Cancel

- 1. Creation of a Tomcat Server in Eclipse
- 2. Dynamic Web project creation
- 3. Jersey library inclusion
- 4. Package and class creation
- 5. web.xml file creation
- 6. Service deployment in the service
- 7. Testing it from Postman

3. Jersey Library Inclusion

- We copy Jersey libraries in WebContent/WEB-INF/lib (we should have unzipped them previously)
- If they are not shown inside the folder after pasting them, you should refresh the project (Right click on the project → Refresh)



- 1. Creation of a Tomcat Server in Eclipse
- 2. Dynamic Web project creation
- 3. Jersey library inclusion
- 4. Package and class creation
- 5. web.xml file creation
- 6. Service deployment in the service
- 7. Testing it from Postman

4. Package and Class Creation (i)

 We create a package inside the JavaResources/src name packageName.

• VERY IMPORTANT: Respect upper-case and lower-case conventions for Java, specially for the first letter

0	Java EE - Eclipse	New Java Package	- - ×
File Edit Navigate Search Project [™] ▼ □ □ □ ▼ ▼ ▼ [™] □ □ □ ▼ ▼ ▼ ▼ [™] □ □ □ ↓ ▼ ▼ ▼	Run Window Help IP II ■ M R R P IR 表 R (> ▼ -> ▼ [Java Package Create a new Java package.	Ť
Project Ex X Project Ex X HelloWorld Deployment Descr A JAX-WS Web Servio Marka Resources		Creates folders corresponding to packages. Source folder: HelloWorld/src Name: nombrePaquete	Browse
▷ Alan ▷ bu ▷ bu ▷ bu ○ Wu Compare With	 Project Class Interface Package Source Folder Filter Listener Servlet 		
	Example Other Ctrl+N	? Finish	Cancel

4. Package and Class Creation (ii)

- We create the class Hello and we introduce the code in the following slide.
- VERY IMPORTANT: Respect upper-case and lower-case conventions for Java, specially for the first letter

0		Java El	E - Eclipse			0	New Java Class
File Edit Navigate ♪ ▼ 📄 🕼 🎰 ೨ 🕞 🔗 ▼ 🌚 2	e S	earch Project Run Window	Help 과 유 교 등 🧝 🕯 Quin	Ş ▼ ck Ac		Java Class Create a new Java	class.
Project Ex X	69	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	•		An outline is not availa	Source folder: Package:	HelloWorld/src nombrePaquete
j		Go Into Open Type Hierarchy Show In	F4 Alt+Shift+W ►	@ 6	Annotation Class Enum	Enclosing type:	
▷ ➡ Librari ▷ ➡ JavaScript ▷ ➢ build ▷ ➢ WebCont		Copy Copy Qualified Name	Ctrl+C	0 8 8 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Interface Package	Modifiers:	public Opackage Oprivate abstract final static
b Servers	10 X	Paste Delete Remove from Context	Ctrl+V Delete Ctrl+Alt+Shift+Down	© • • ●	Film File	Superclass: Interfaces:	java.lang.Object
		Source Refactor	Alt+Shift+S ► Alt+Shift+T ►	*8	Listener Servlet		
		Import Export			Other Ctrl+N	Which method stu	bs would you like to create? public static void main(String[] args) Constructors from superclass
	8	Refresh Profile As Debug As Run As Validate Team	+5 + + +	Re	e Explo Snippets 🖻	Do you want to add	☑ Inherited abstract methods d comments? (Configure templates and de ☐ Generate comments
<		Compare With Restore from Local History	,		-	?	[
H nombrePaquete -		Properties	Alt+Enter			-	

_ 🗆

Browse...

Browse...

Browse..

Add... Remove

Cancel

protected

ault value <u>here</u>)

Finish

4. Package and Class Creation (iii) Use the class name you chose @Path("/hello") package packageName; Use the package public class Hello { name you chose import javax.ws.rs.GET; QGET import javax.ws.rs.Path; @Produces(MediaType.TEXT PLAIN) import javax.ws.rs.Produces; public String sayPlainTextHello() {return "Hello import javax.ws.rs.core.MediaType; Plain"; } NOTE: Do not forget to save the file after pasting or modifying the code

- 1. Creation of a Tomcat Server in Eclipse
- 2. Dynamic Web project creation
- 3. Jersey library inclusion
- 4. Package and class creation
- 5. web.xml file creation
- 6. Service deployment in the service
- 7. Testing it from Postman

5. web.xml File Creation (i)

- We create web.xml file in WebContent/WEB-INF
- The XML editor opens automatically. Click on "Source" in the tab below to see it in text mode.
- We copy the code in the following slide and we save it.

٢	Java EE - HelloWorld/src/nombrePaquete/Helllo.java - Eclipse 💦 – 🗖	0
File Edit Source	Refactor Navigate Search Project Run Window Help	File Create a new f
 Project Ex S Project Ex S HelloWorld Browney Browney<th>Image: Construction of the second second</th><th>Enter or select t HelloWorld/W C C C A C C C C C C A C C C C C C A C C C C C C C A C C C C C C C A C C C C C C C C A C C C C C C C C C A C C C C C C C C C C C A C C C C C C C C C C C C C C C C C C C</th>	Image: Construction of the second	Enter or select t HelloWorld/W C C C A C C C C C C A C C C C C C A C C C C C C C A C C C C C C C A C C C C C C C C A C C C C C C C C C A C C C C C C C C C C C A C C C C C C C C C C C C C C C C C C C
< 📂 WEB-INF - Helk	Restore from Local History Source Properties Alt+Enter	?

2	New File		×	X we	hxml 🕾 🛙 🖬
ile					
Create a new file resource.				1	xml vers</td
				20	<web-app></web-app>
Enter or select the parent folder:				3	
HelloWorld/WebContent/WEB-INF				4⊝	<pre><servlet></servlet></pre>
☆ ⇔				5	<servlet< td=""></servlet<>
4 🐸 HelloWorld				6	<servlet< td=""></servlet<>
.settings				7⊝	<i< td=""></i<>
Src				8	
			1.5	9⊝	
			1.55	10	
RemoteSystemsTempFiles			193	11	</td
Servers				12	
				130	<servlet-m< td=""></servlet-m<>
			1.05	14	<servle< td=""></servle<>
				15	<url-pa< td=""></url-pa<>
ile name: web.xml				16	
Advanced >>				17	
				18	
					1
?	Fir	nish Can	cel	Decia	Source
<u> </u>				Desig	Source
					J

5. web.xml File Creation (ii)

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/webapp_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

5. web.xml File Creation (iii)

<servlet>

```
<servlet-name>My REST service </servlet-name>
```

<servlet-class>org.glassfish.jersey.servlet.ServletContainer</servletclass>

<init-param>

<param-name>jersey.config.server.provider.packages</param-name>

<param-value>

```
packageName, com.fasterxml.jackson.jaxrs.json</param-value>
```

</init-param> Replace by </servlet> your package name

5. web.xml File Creation (iv)

<servlet-mapping>

<servlet-name>My REST service </servlet-name>

<url-pattern>/demo/*</url-pattern>

</servlet-mapping>

</web-app>

NOTE: depending on the operating system the quotation marks are copied wrongly. Revise them when you get an error.

NOTE: You can create first the class and then the web.xml or vice-versa.

Optional

- 1. Creation of a Tomcat Server in Eclipse
- 2. Dynamic Web project creation
- 3. Jersey library inclusion
- 4. Package and class creation
- 5. web.xml file creation
- 6. Service API deployment
- 7. Testing it from Postman

6. Service API Deployment

- We deploy the service in the Tomcat server previously created: Right click on the project → Run as → Run on Server → We select the server created → Finish
- We will be asked to restart the server \rightarrow we click on OK
- If everything worked fine in the deployment, the service will appear as synchronized.
- Otherwise, we will have to click in the console to see the error.
- Do not worry if you get a 404 error in the browser.
- NOTE: To see the console window, problems or error log: Window→ Show View→ General→ *corresponding option*



Http://localhost:8080/HelloWorld/

Estado HTTP 404 - /HelloWorld/

type Informe de estado

mensaje <mark>/HelloWorld</mark>/

descripción El recurso requerido no está disponible.

Apache Tomcat/8.0.14

- 1. Creation of a Tomcat Server in Eclipse
- 2. Dynamic Web project creation
- 3. Jersey library inclusion
- 4. Package and class creation
- 5. web.xml file creation
- 6. Service deployment in the service
- 7. Testing it from Postman

7. Testing it from Postman



Support Bibliography and References

- Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON. By: Masoud Kalali; Bhakti Mehta. Publisher: Packt Publishing Pub. Date: October 15, 2013. Print ISBN-13: 978-1-78217-812-5
- REST with Java (JAX-RS) using Jersey Tutorial Lars Voguel <u>https://www.vogella.com/tutorials/REST/article.html</u>
- Postman API Client Postman Inc. <u>https://www.postman.com/product/api-client/</u>