

**Author: Hong Luu**

**This document will show how to get a Maven Restful Web Service template that can produce JSON and XML. In addition, it solves the issue where unwanted null values appear in the output.**

**Assumption is you already have the following installed already:**

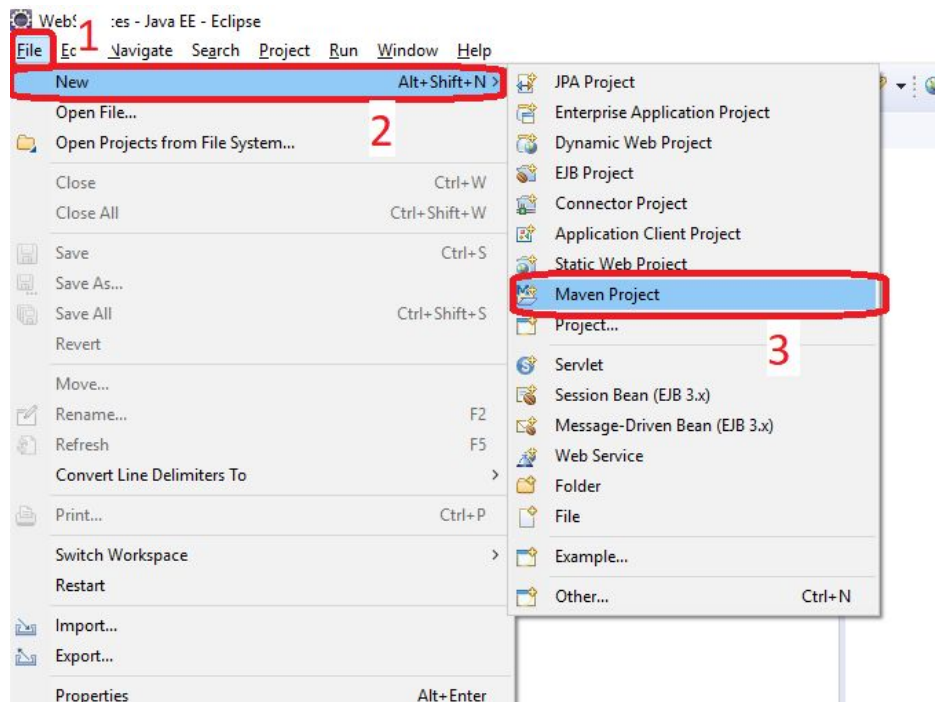
Eclipse JavaEE (I'm using Neon.2 Release)

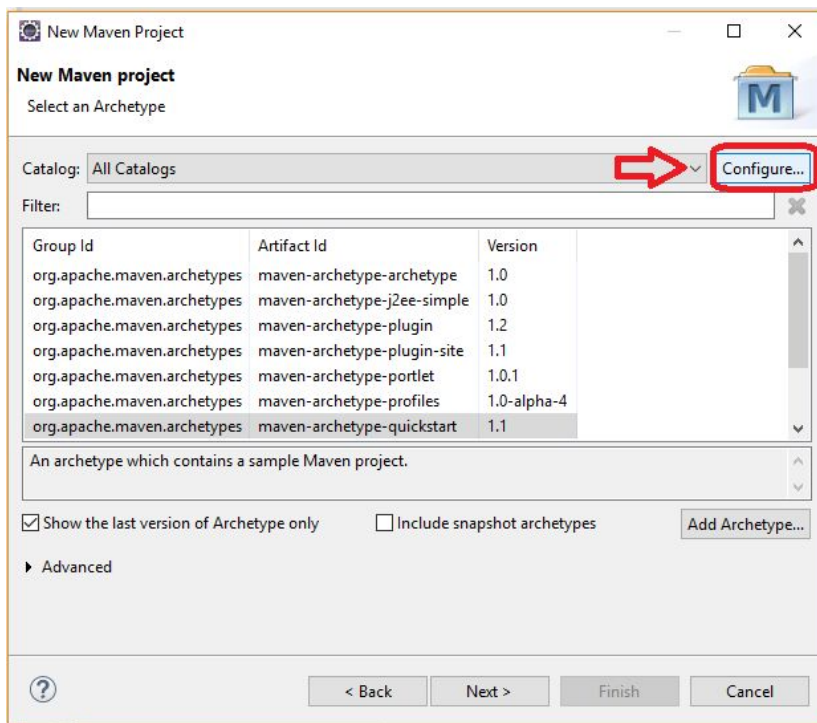
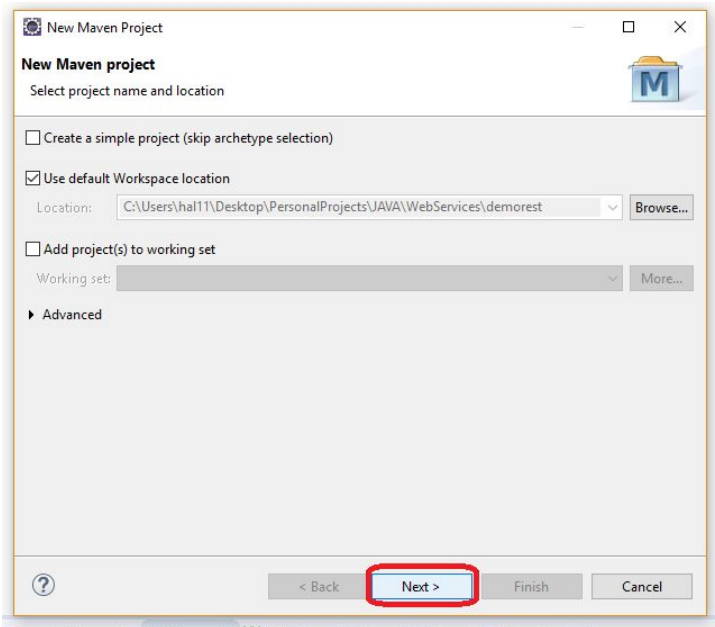
Java 1.7 (or above)

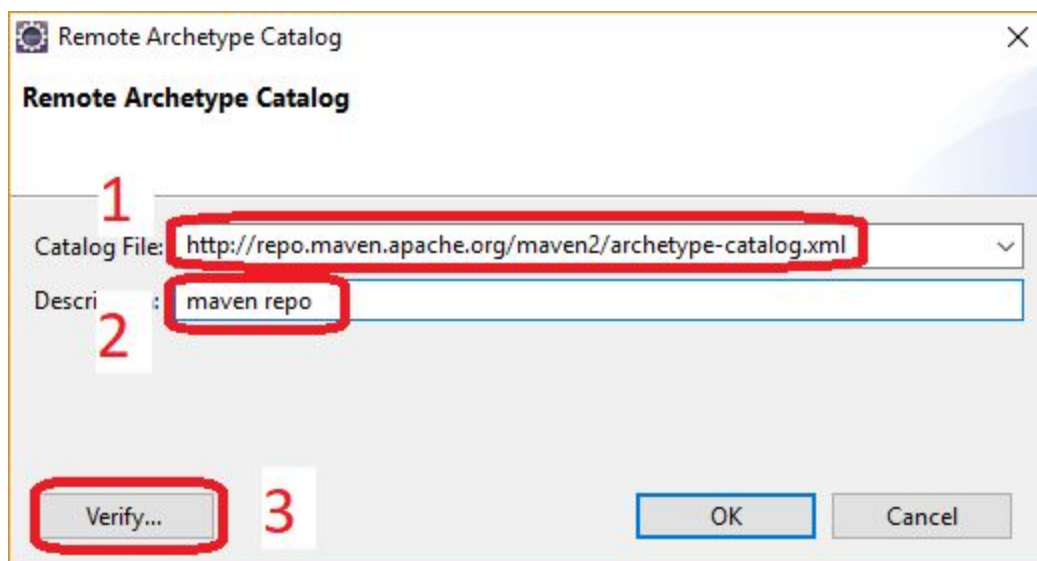
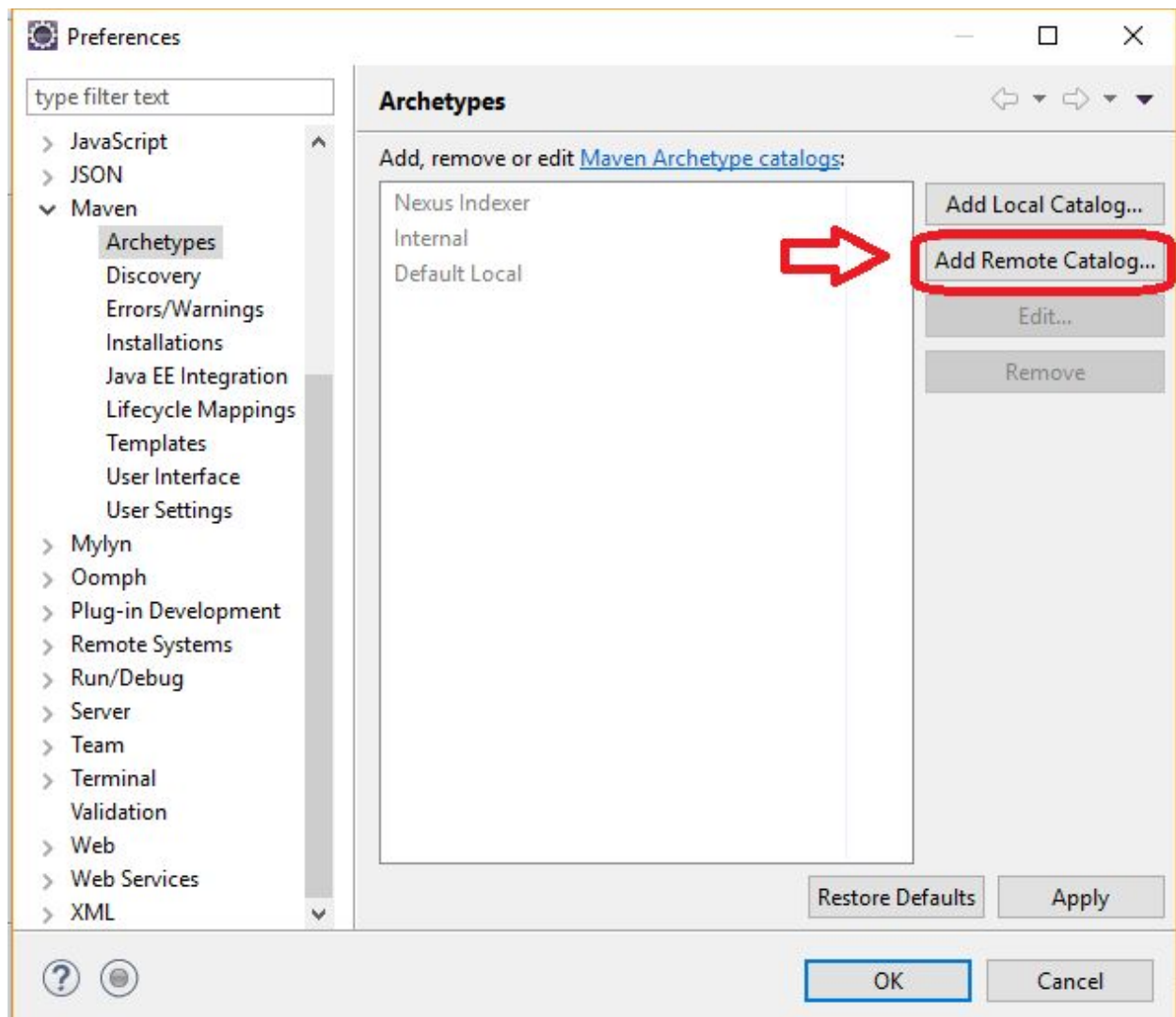
Apache Tomcat server (8.0x or above)

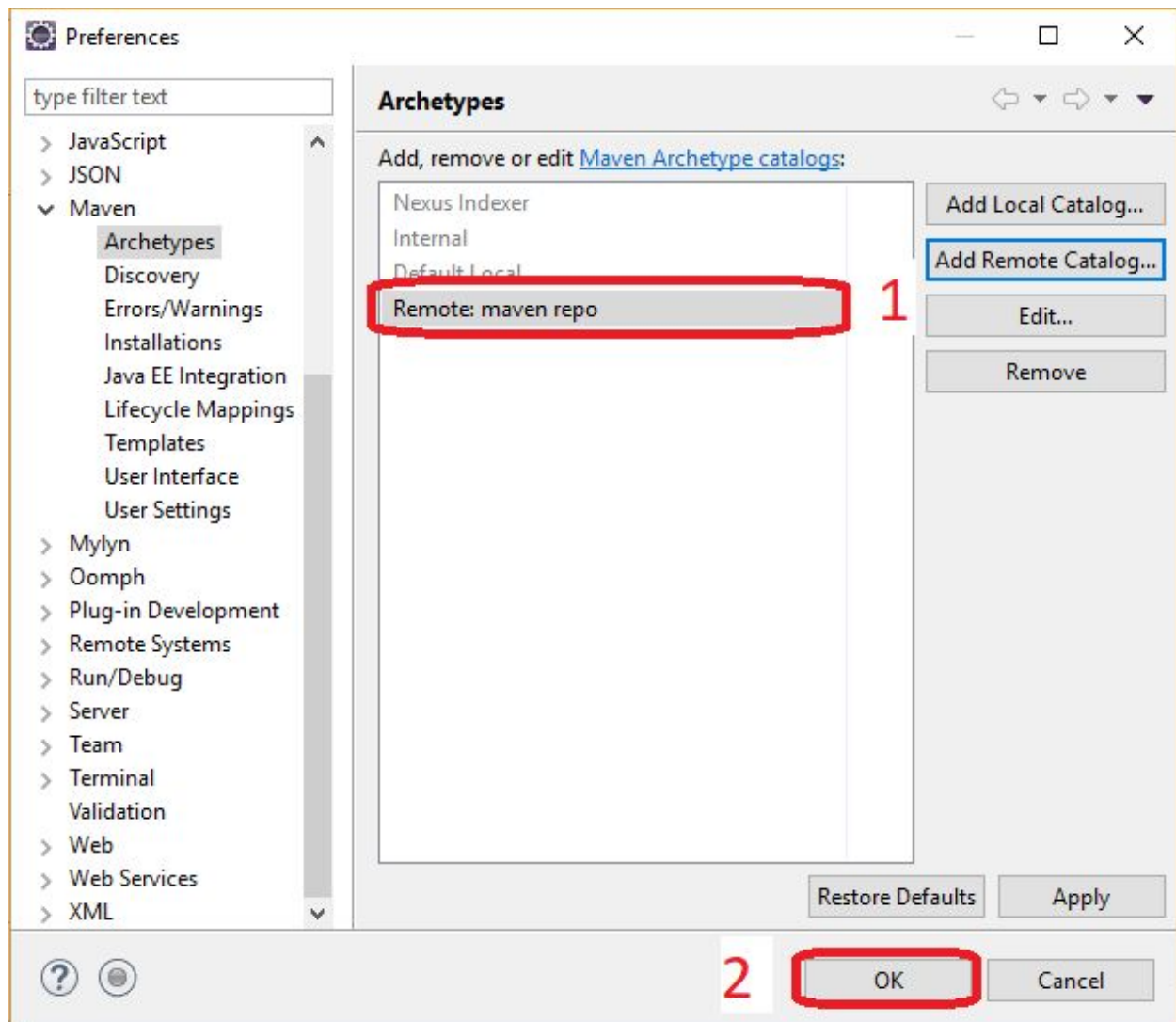
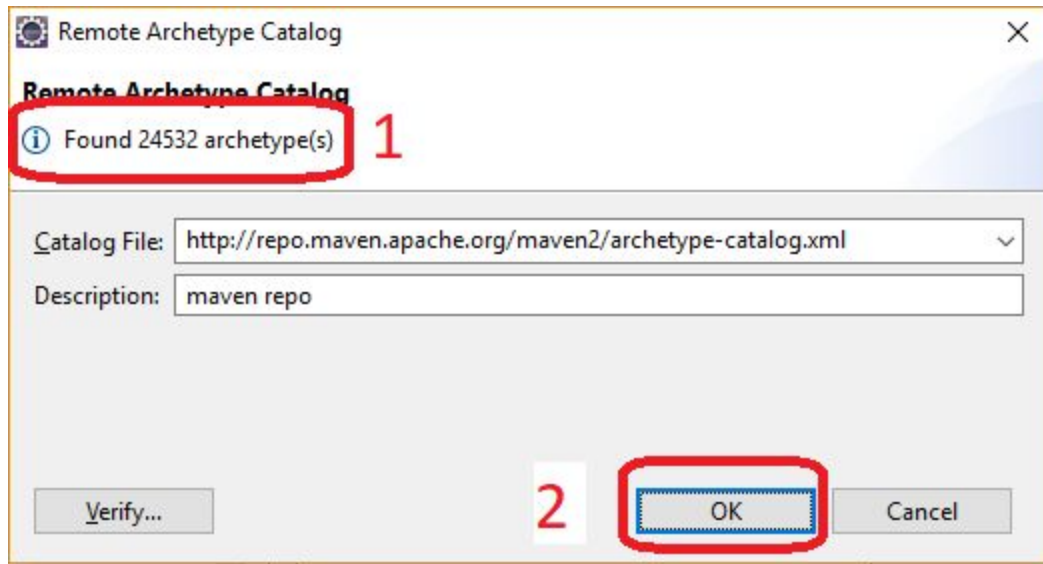
For a more thorough tutorial, here is a good resource:

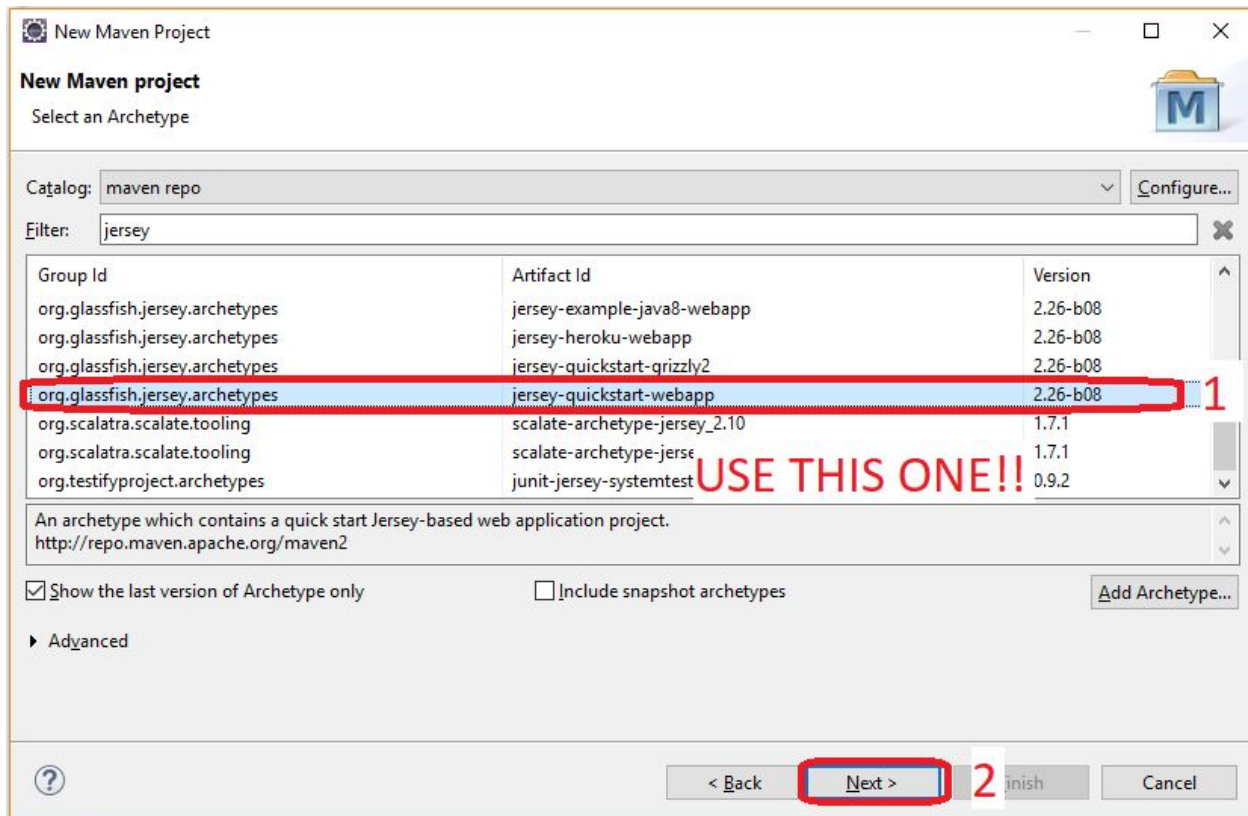
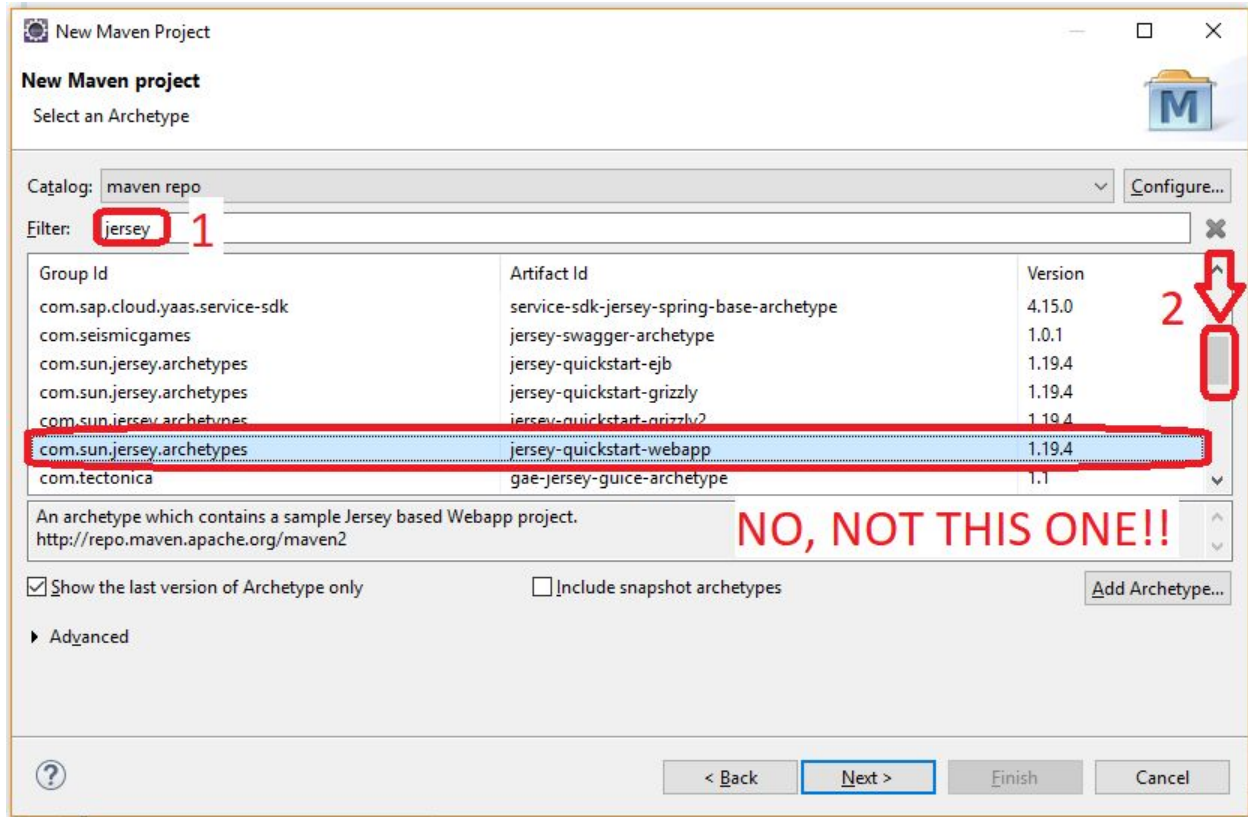
<https://www.youtube.com/watch?v=DWPN45AJA3k&list=PLsyeobzWxl7ps4Z1C4VMtvZEx5-PgyoYI&index=2>











**New Maven Project**

Specify Archetype parameters

Group Id: **com.java.mystuff** <sup>1</sup>

Artifact Id: **mavenExample** <sup>2</sup>

Version: 0.0.1-SNAPSHOT

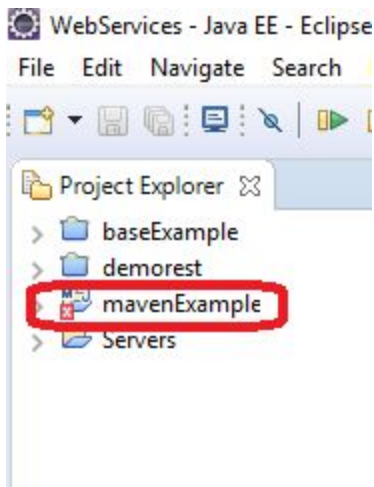
Package: com.java.mystuff.mavenExample

Properties available from archetype:

Name	Value

► Advanced

**Finish** <sup>3</sup>





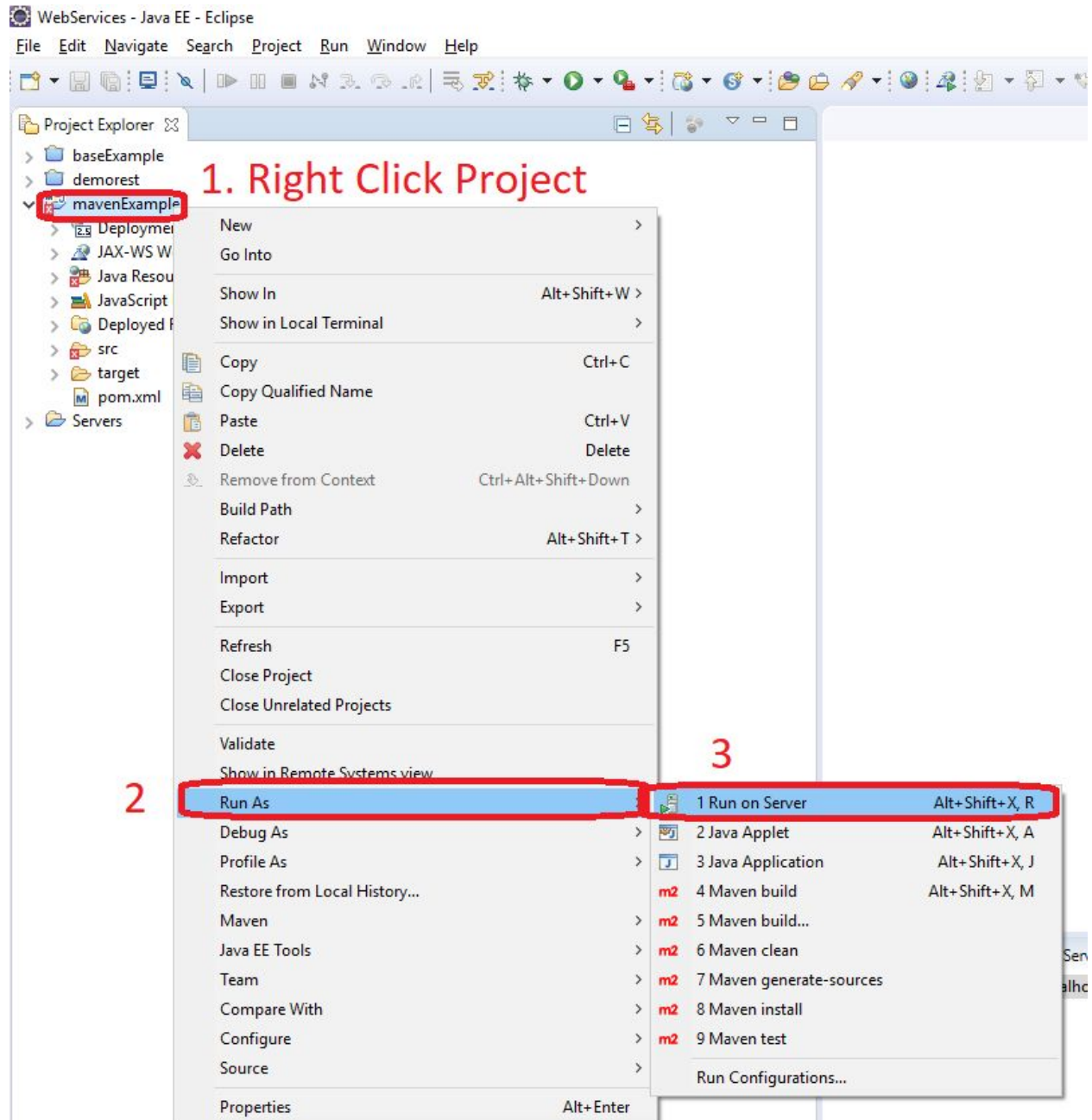
WebServices - Java EE - Eclipse

File Edit Navigate Search Project Run Window Help

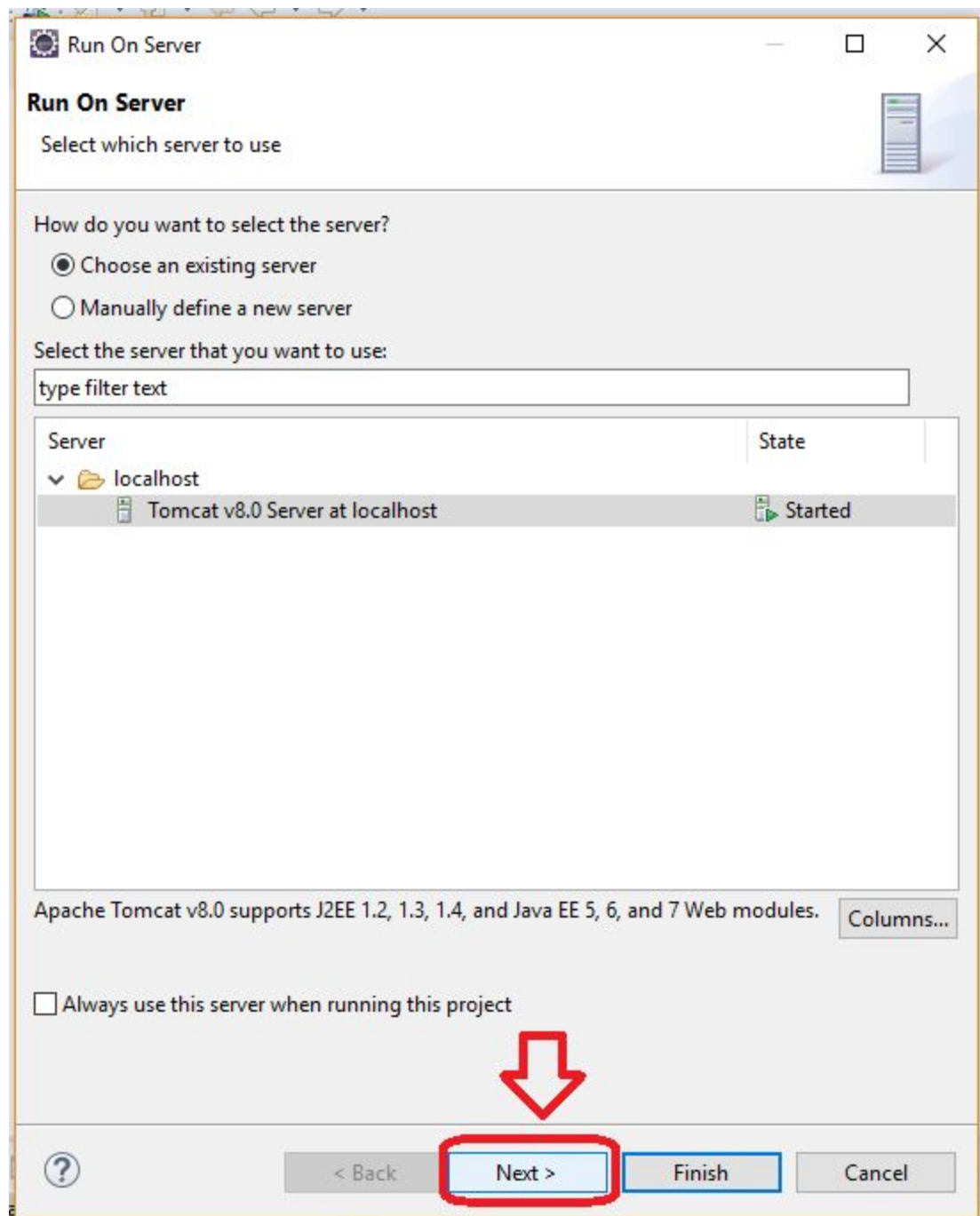


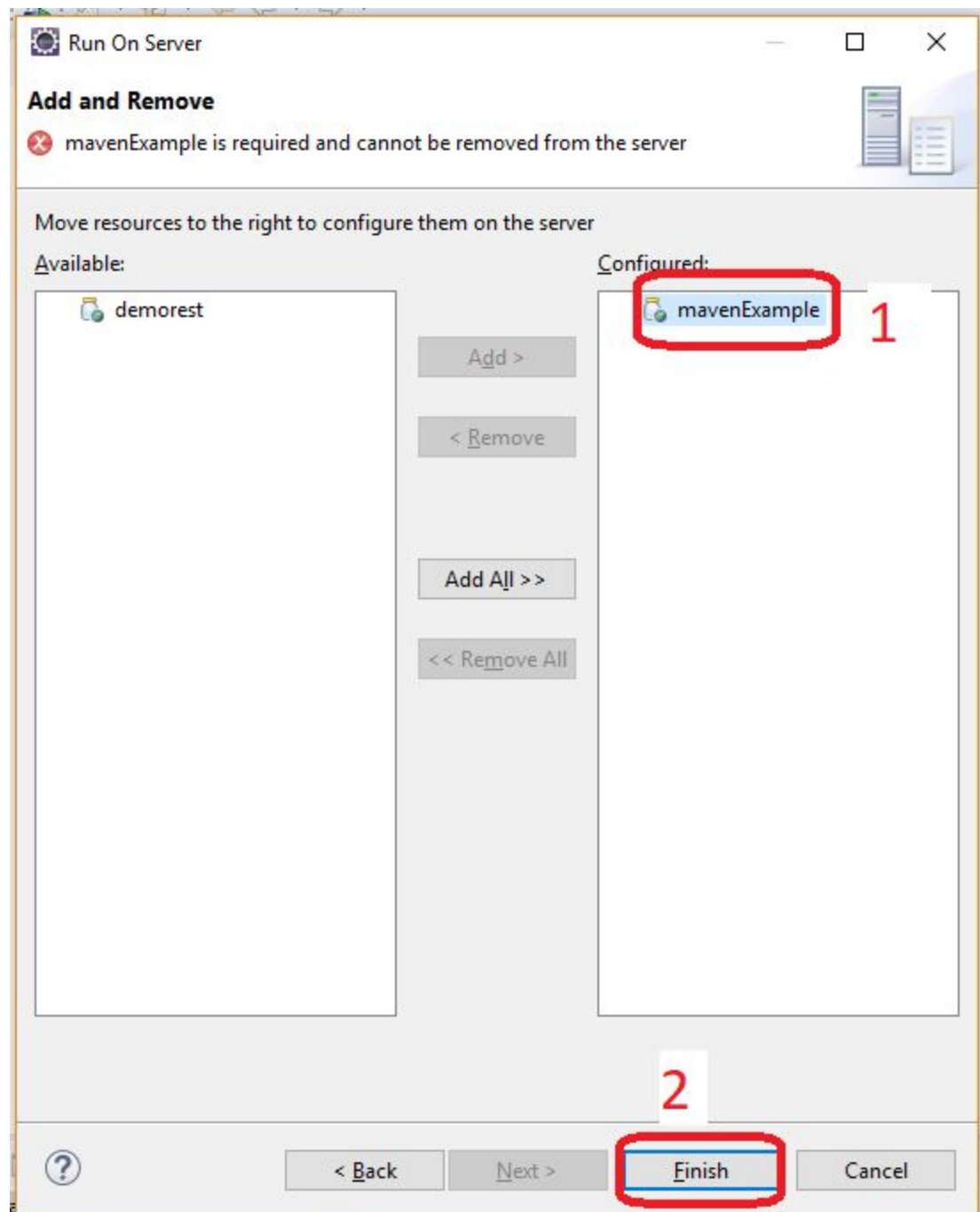
Project Explorer

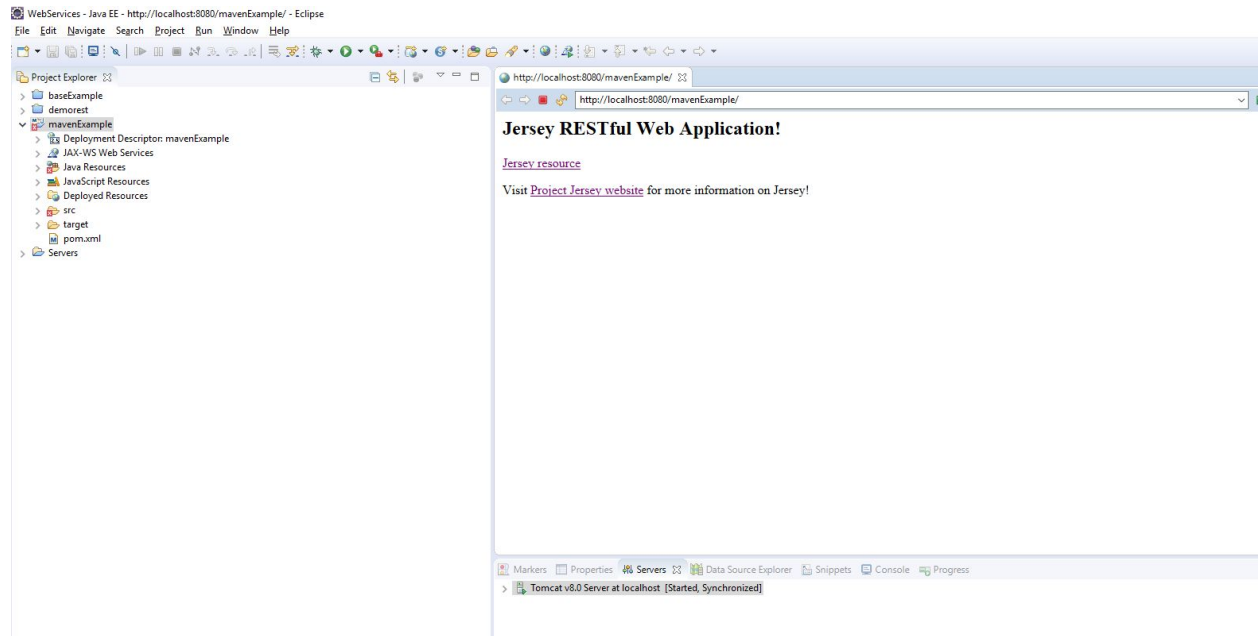
- > baseExample
- > demorest
- ▼ mavenExample
  - > Deployment Descriptor: mavenExample
  - > JAX-WS Web Services
  - ▼ Java Resources
    - ▼ src/main/java
      - ▼ com.java.mystuff.mavenExample
        - > MyResource.java
      - src/main/resources
    - ▼ Libraries
      - > JRE System Library [JavaSE-1.7]
      - > Maven Dependencies
    - > JavaScript Resources
    - > Deployed Resources
  - ▼ src
    - ▼ main
      - ▼ java
        - ▼ com
          - ▼ java
            - ▼ mystuff
              - ▼ mavenExample
                - MyResource.java
        - resources
      - ▼ webapp
        - ▼ WEB-INF
          - web.xml
        - index.jsp
    - ▼ target
      - ▼ m2e-wtp
        - ▼ web-resources
          - ▼ META-INF
            - ▼ maven
              - ▼ com.java.mystuff
                - ▼ mavenExample
                  - pom.properties
                  - pom.xml
          - MANIFEST.MF
    - ▼ pom.xml
  - > Servers











WebServices - Java EE - Eclipse

File Edit Navigate Search Project Run Window Help

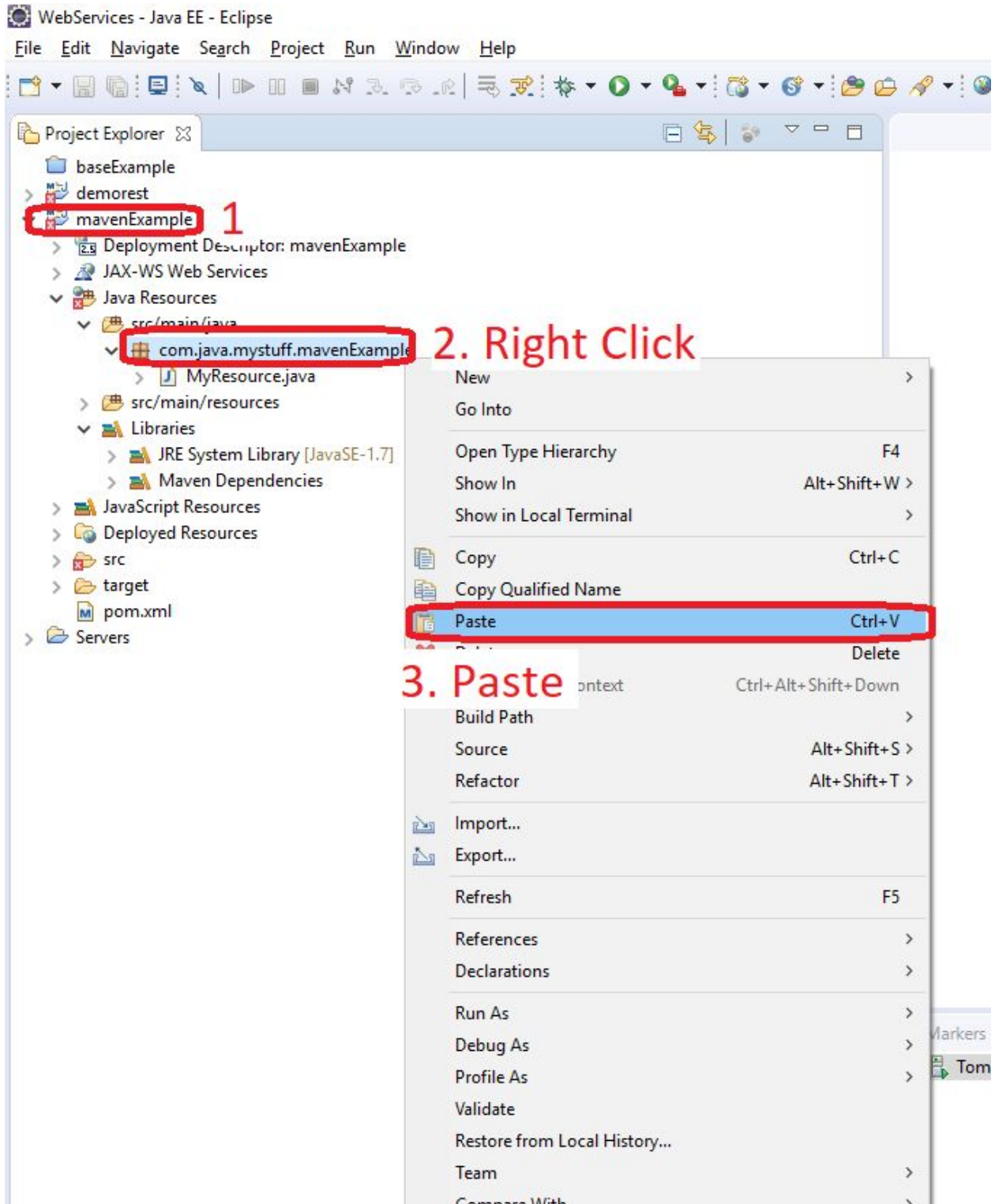


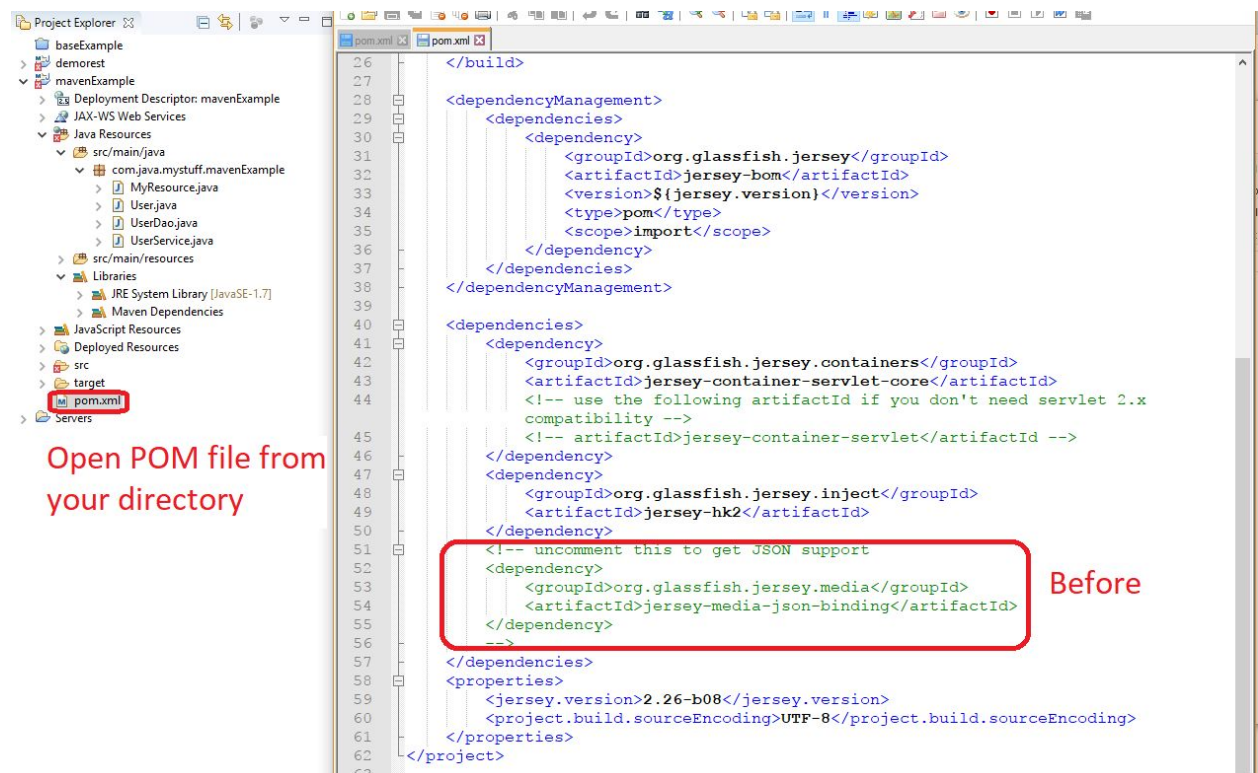
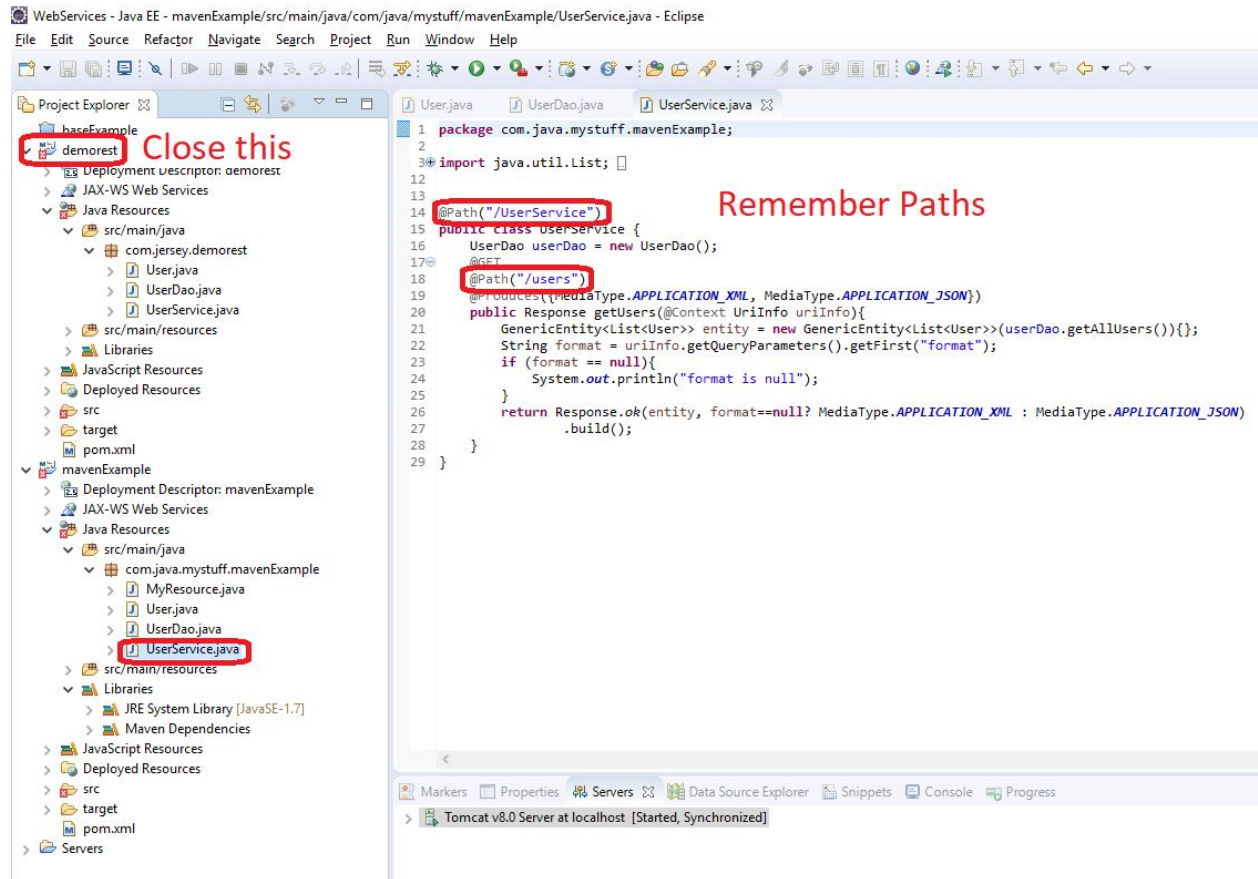
Project Explorer

- baseExample
  - ✓ demorest 1
    - > Deployment Descriptor: demorest
    - > JAX-WS Web Services
    - ✓ Java Resources
      - ✓ src/main/java
        - ✓ com.jersey.demorest
          - > User.java
          - > UserDao.java
          - > UserService.java
        - > src/main/resources
        - > Libraries
        - > JavaScript Resources
        - > Deployed Resources
        - > src
        - > target
        - > pom.xml
      - > mavenExample
      - > Servers

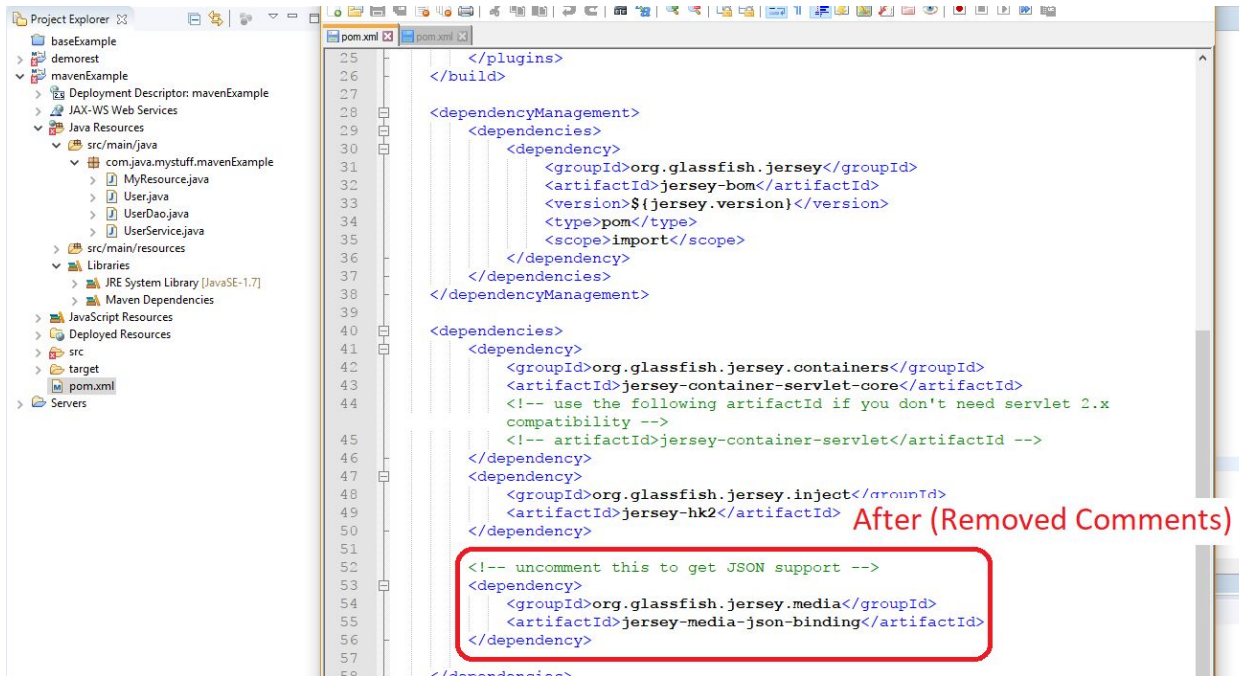
Getting files from  
other project

2. Copy



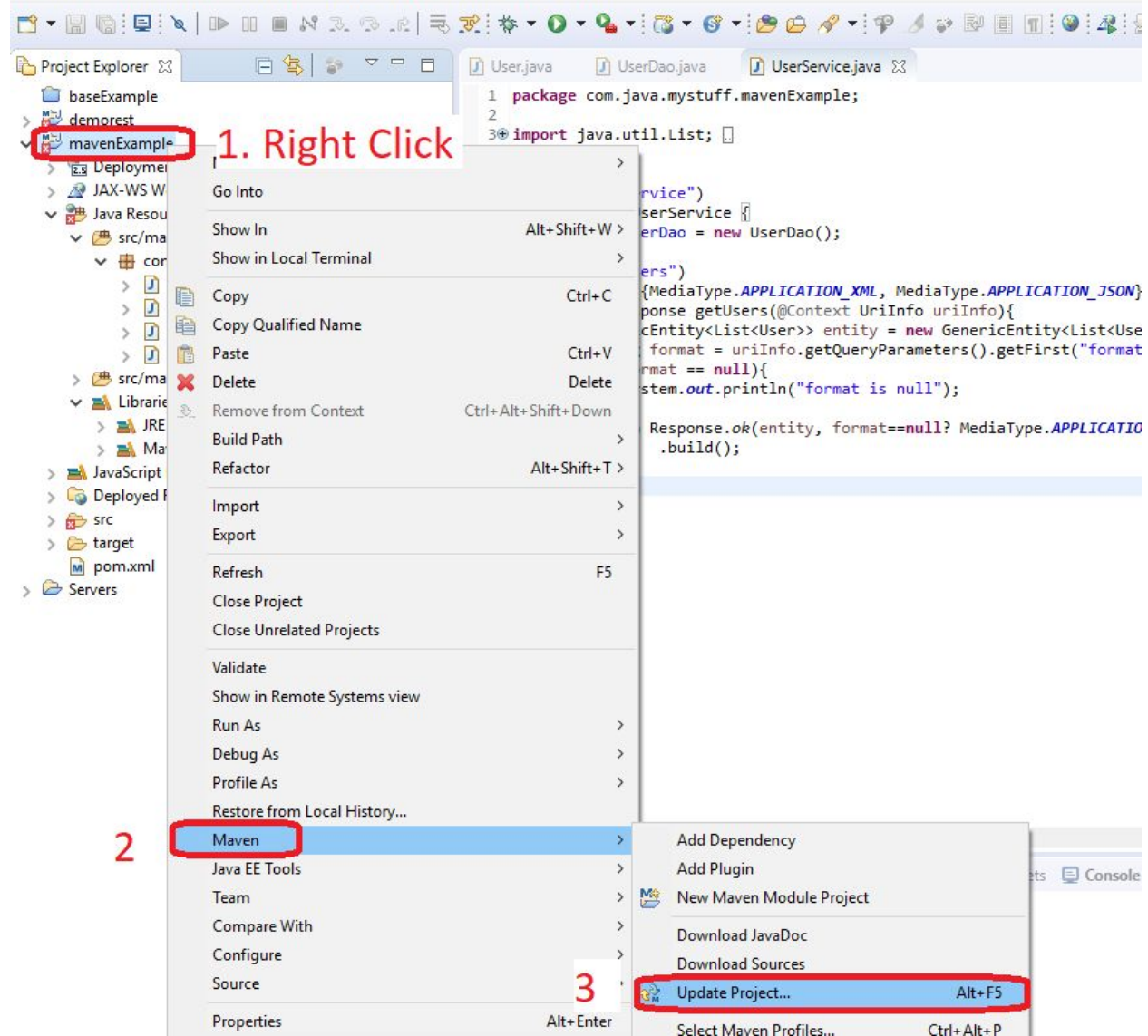


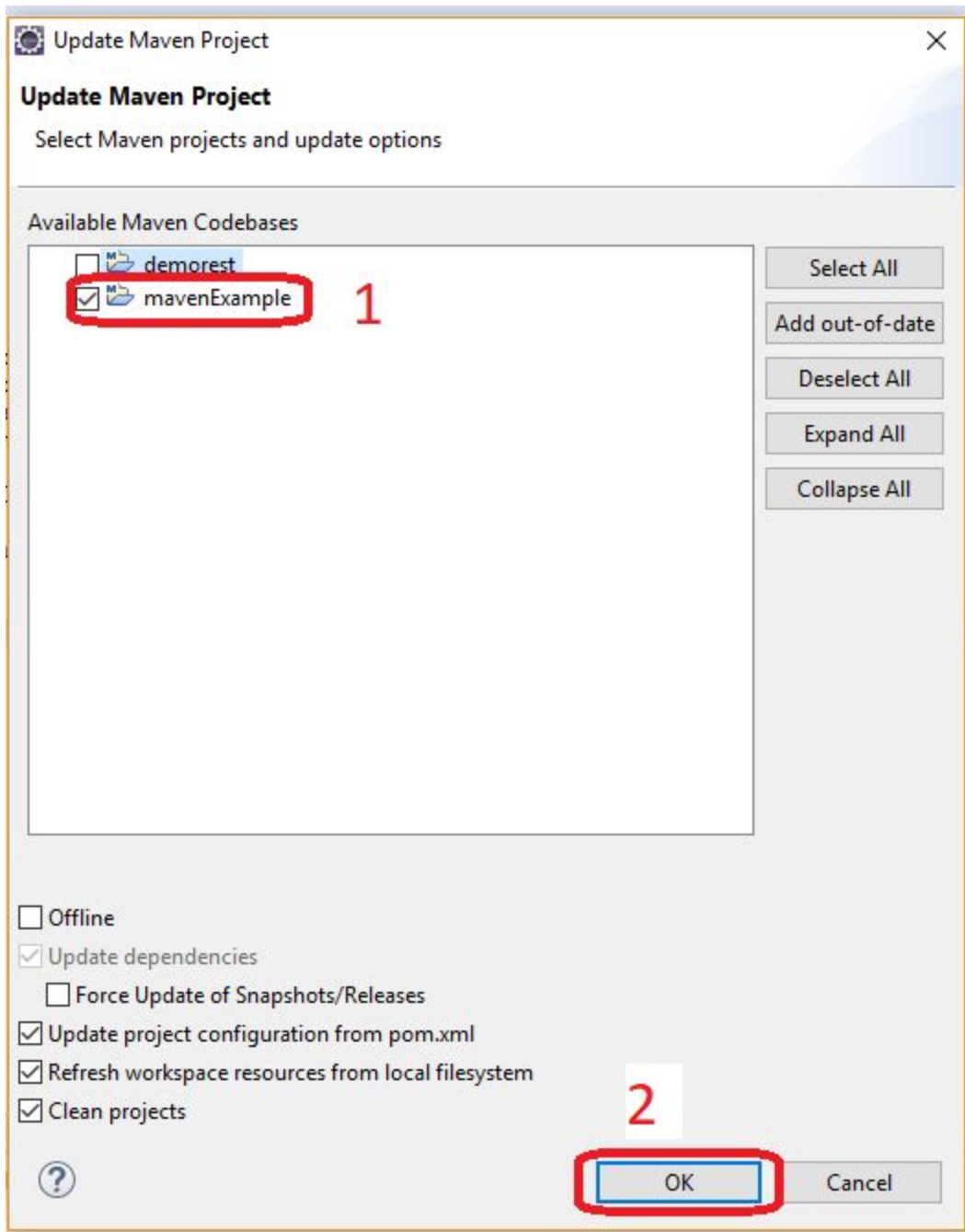




WebServices - Java EE - mavenExample/src/main/java/com/java/mystuff/mavenExample/UserService.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help





The screenshot shows the Eclipse IDE interface. In the Project Explorer on the left, the 'mavenExample' project is selected and highlighted with a red box, with a red '1' and the text '1. Right Click' next to it. A context menu is open over this project, listing various actions like 'New', 'Go Into', 'Copy', 'Paste', etc. The 'Run As' option is highlighted with a red box and a red '2'. A submenu is open for 'Run As', showing options like '1 Run on Server', '2 Java Applet', '3 Java Application', etc. The '1 Run on Server' option is highlighted with a red box and a red '3'. The main editor on the right shows the code for 'UserService.java'.

```

1 package com.java.mystuff.mavenExample;
2
3 import java.util.List;

...

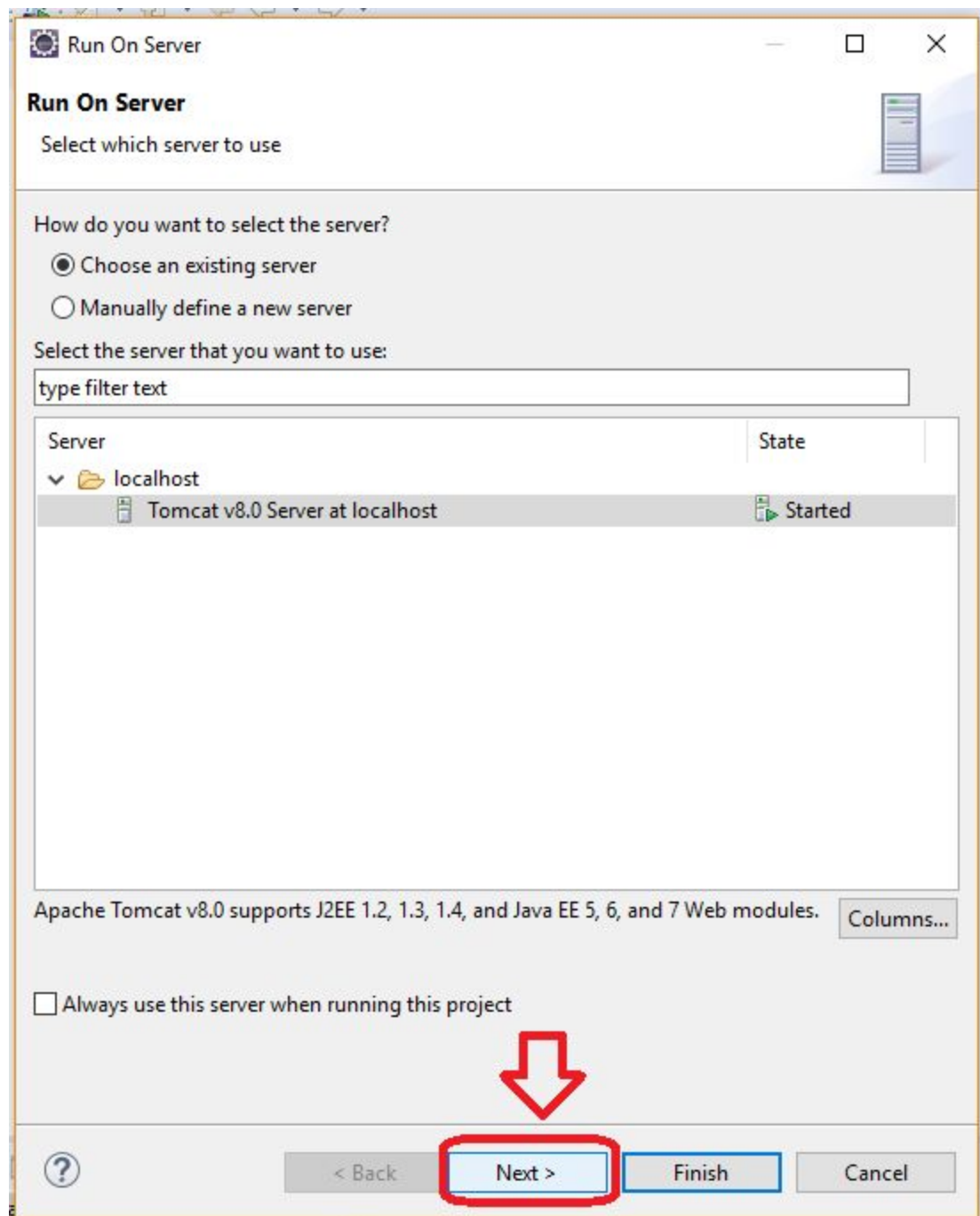
service")
UserService {
    UserDao = new UserDao();

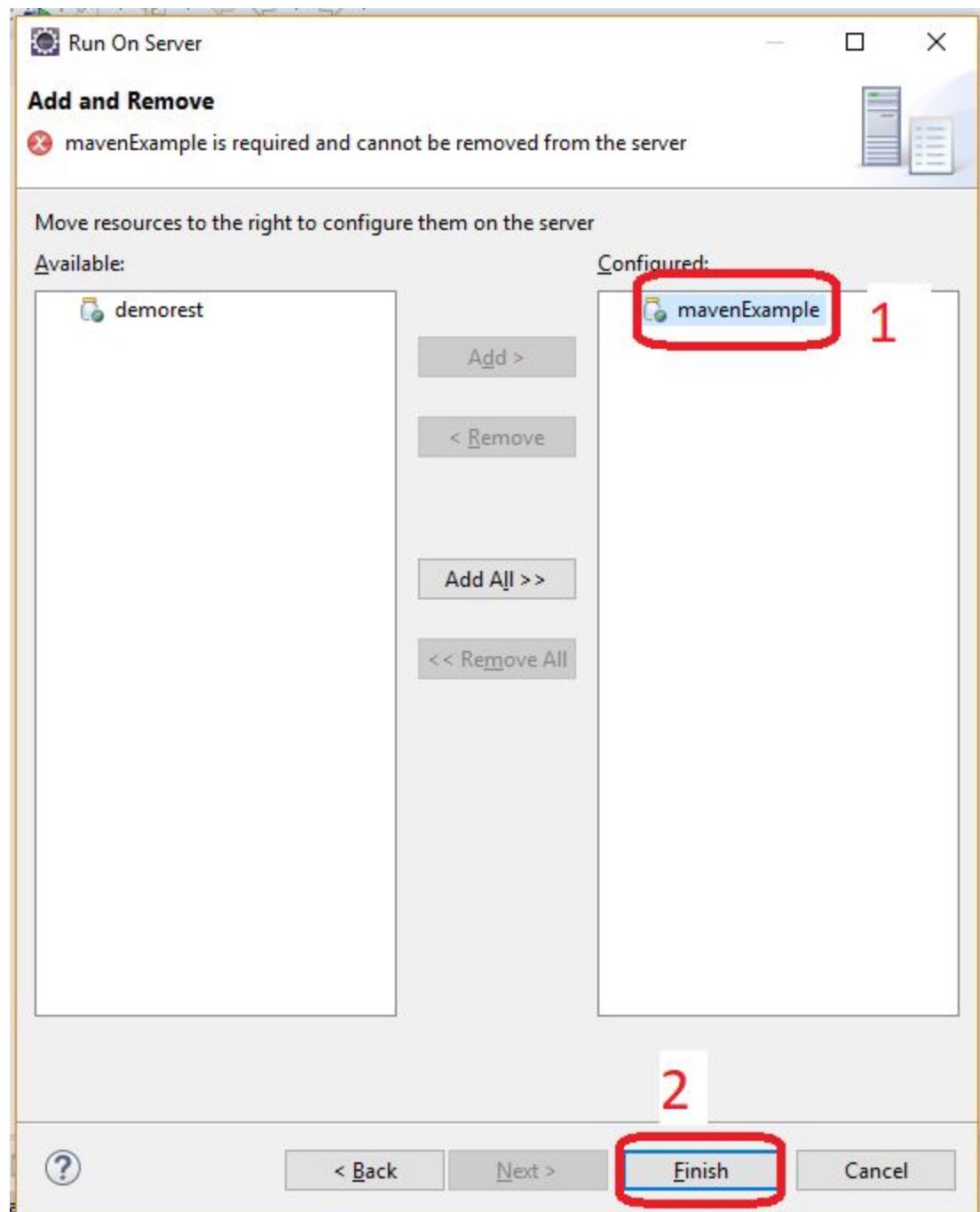
...

sers")
({MediaType.APPLICATION_XML, MediaType.APPLICATION
sponse getUsers(@Context UriInfo uriInfo){
    GenericEntity<List<User>> entity = new GenericEntity
    g format = uriInfo.getQueryParameters().getFirs
    rmat == null){
        system.out.println("format is null");

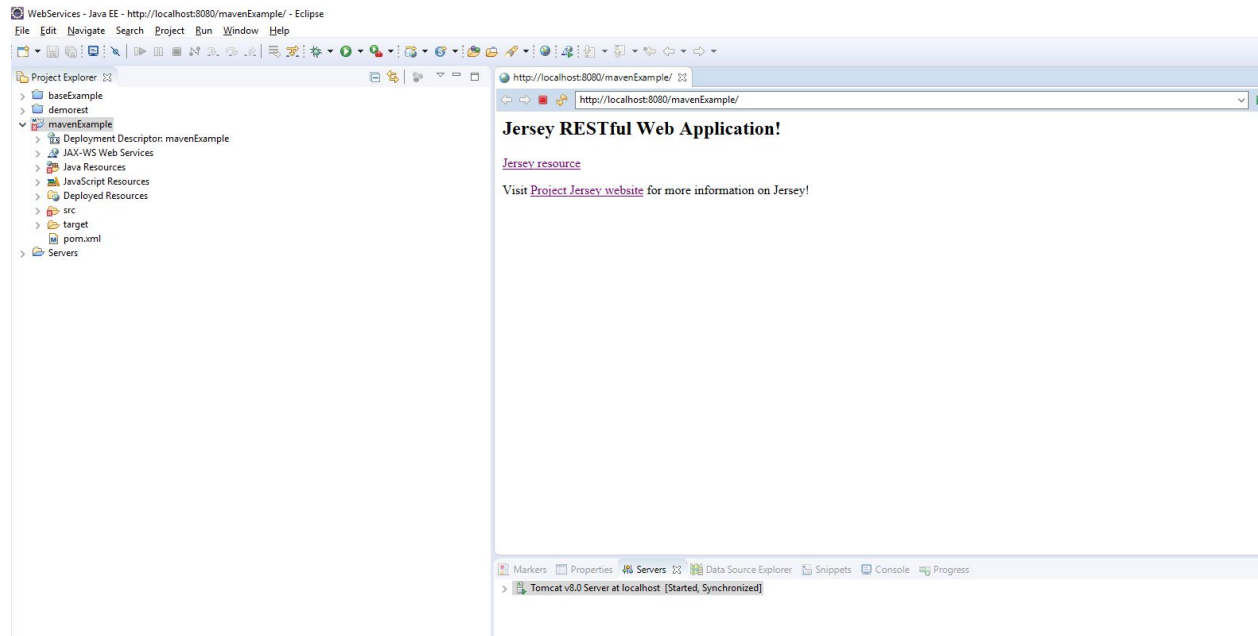
...

n Response.ok(entity, format==null? MediaType.A
        .build();
    
```









```
User.java UserDao.java UserService.java http://localhost:8080
1 package com.java.mystuff.mavenExample;
2
3 import java.util.ArrayList;
4
5
6 public class UserDao {
7     public List<User> getAllUsers(){
8
9         List<User> userList = null;
10        User user = new User(1, "Name01", "Teacher01");
11        User user2 = new User(2, null, "Teacher02");
12        userList = new ArrayList<User>();
13        userList.add(user);
14        userList.add(user2);
15        return userList;
16    }
17 }
```

NULL

Builder Team Library IN SYNC hluu114@...

http://localhost:8080/web-s http://localhost:8080/ x + No Environment

GET http://localhost:8080/mavenExample/webapi/UserService/users Params Send

Authorization Headers Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (4) Tests Status: 200 OK

Pretty Raw Preview XML

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <users>
3   <user>
4     <id>1</id>
5     <name>Name01</name>
6     <profession>Teacher01</profession>
7   </user>
8   <user>
9     <id>2</id>
10    <profession>Teacher02</profession>
11  </user>
12 </users>
```

XML output without NULLS

Builder Team Library IN SYNC hluu114@...

http://localhost:8080/web-s http://localhost:8080/ x + No Environment

GET http://localhost:8080/mavenExample/webapi/UserService/users?format=json Params Send

Authorization Headers Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (4) Tests Status: 200 OK

Pretty Raw Preview JSON

```
1 [
2   {
3     "id": 1,
4     "name": "Name01",
5     "profession": "Teacher01"
6   },
7   {
8     "id": 2,
9     "profession": "Teacher02"
10  }
11 ]
```

JSON output without NULLS