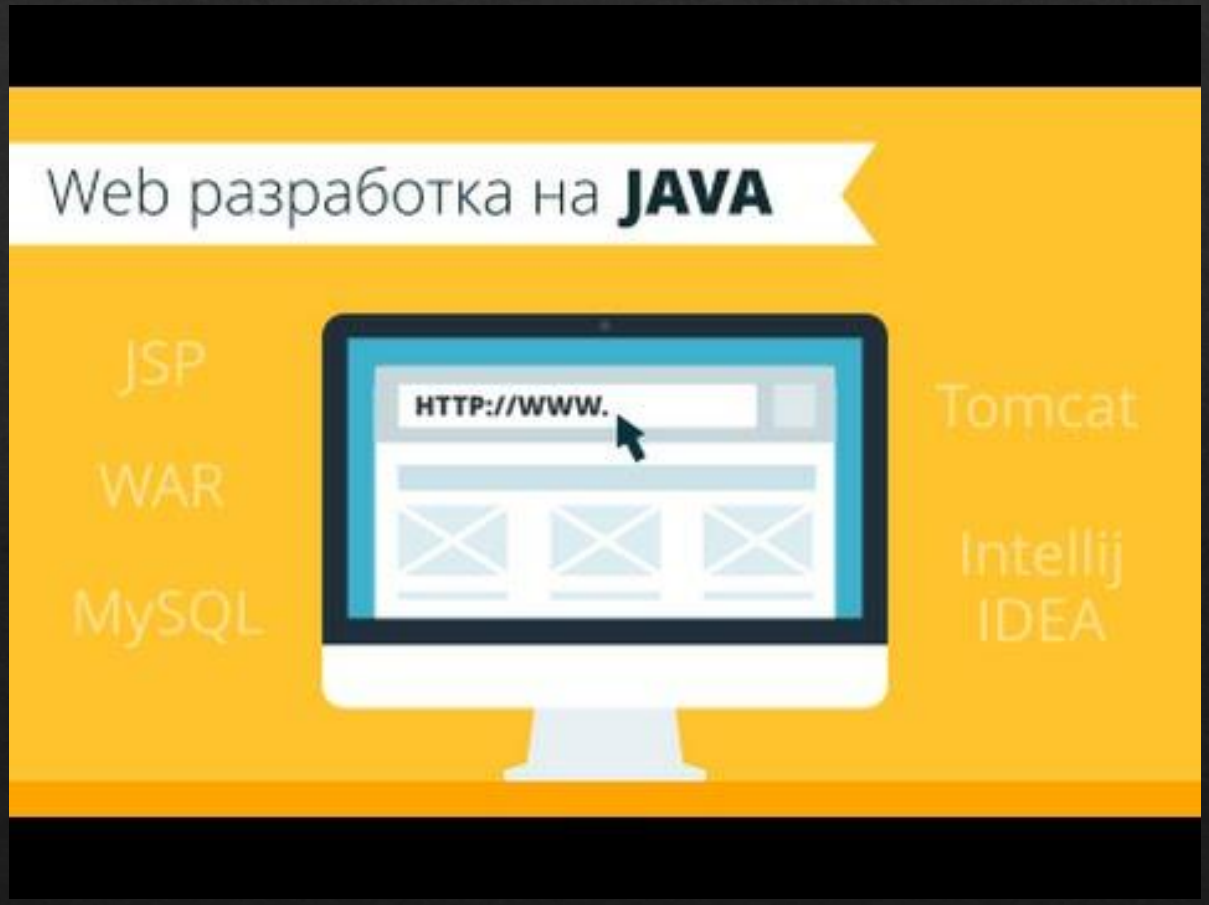
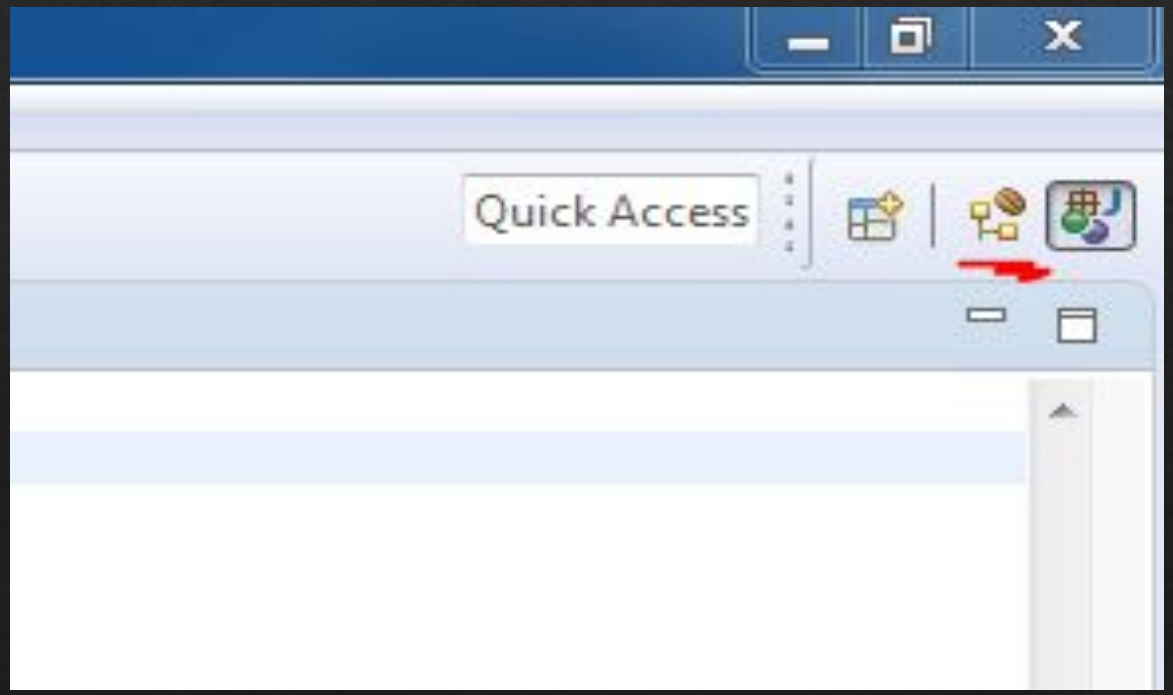
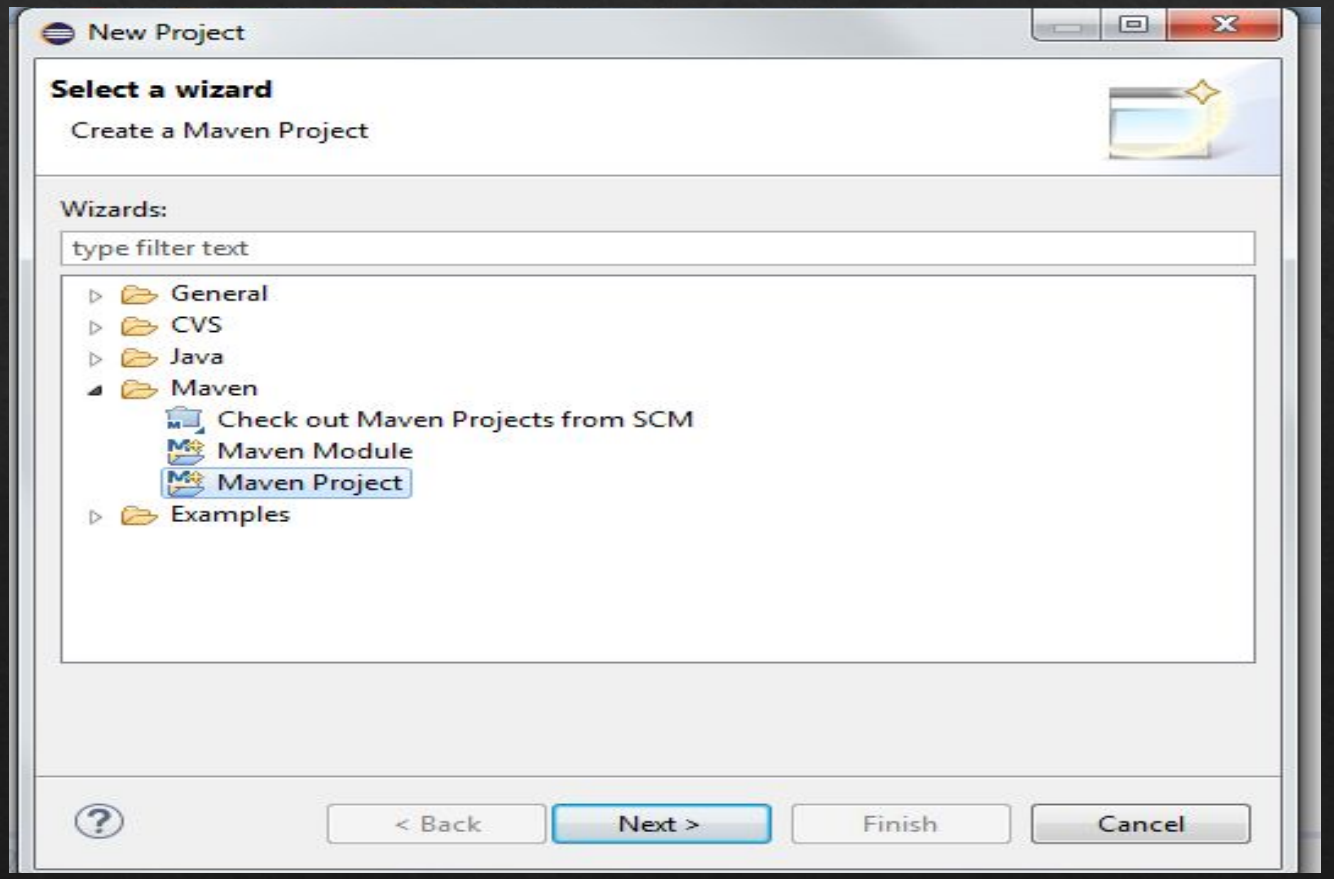
WEB java


Web разработка на JAVA
JSP
WAR
MySQL
HTTP://WWW.
Tomcat
Intellij
IDEA

В Eclipse нужно выбрать с Java на Java EE (серверная платформа) в правом верхнем углу. (проверить, чтобы так же было)

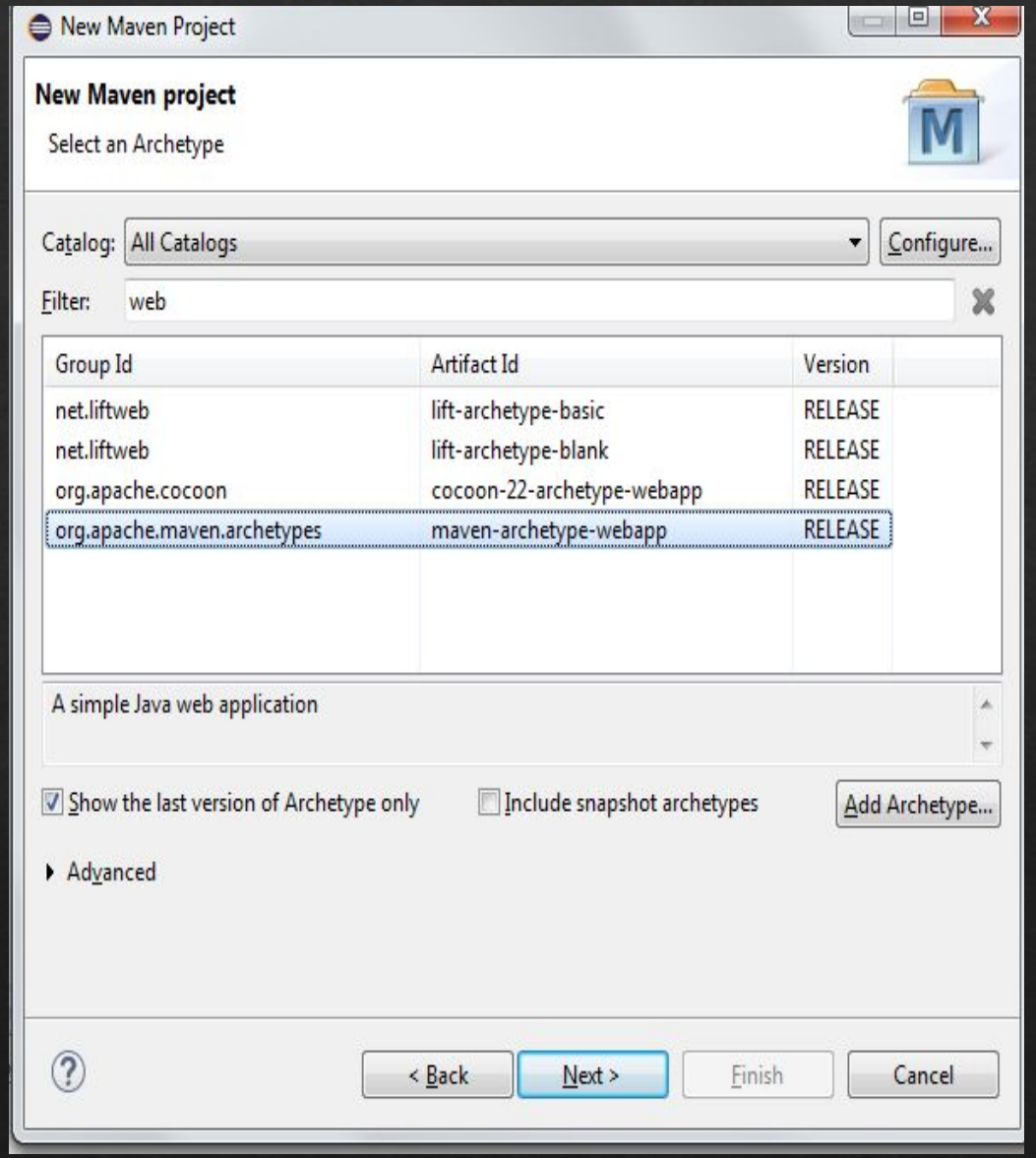


Теперь нужно создать новый Maven проект. File -> new -> other… -> maven -> Maven Project

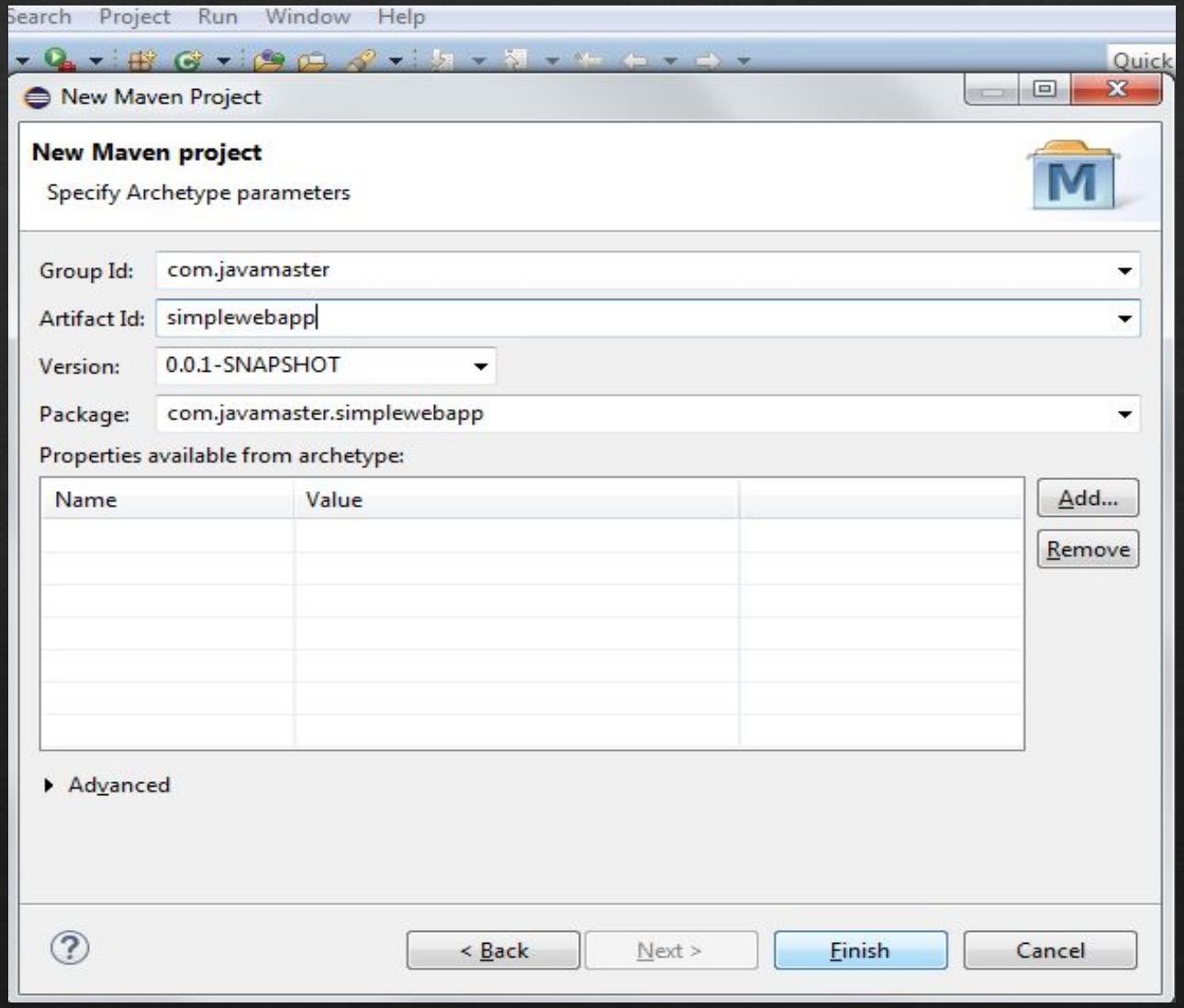


Нажимаем Next и далее нас перебрасывает в окно выбора так называемого архетипа (archetype). Мы используем Maven как инструмент сборки потому, он практичен, лёгок в усвоение. Нам не нужно больше искать дополнительные jar библиотеки в Интернете (как с mp3 или jdbc), качать их и подключать. Maven позволяет нам подключать дополнительные библиотеки в специальном файле настроек, который называется pom.xml.

При выборе архетипа нужно выбрать maven-archetype-webapp:



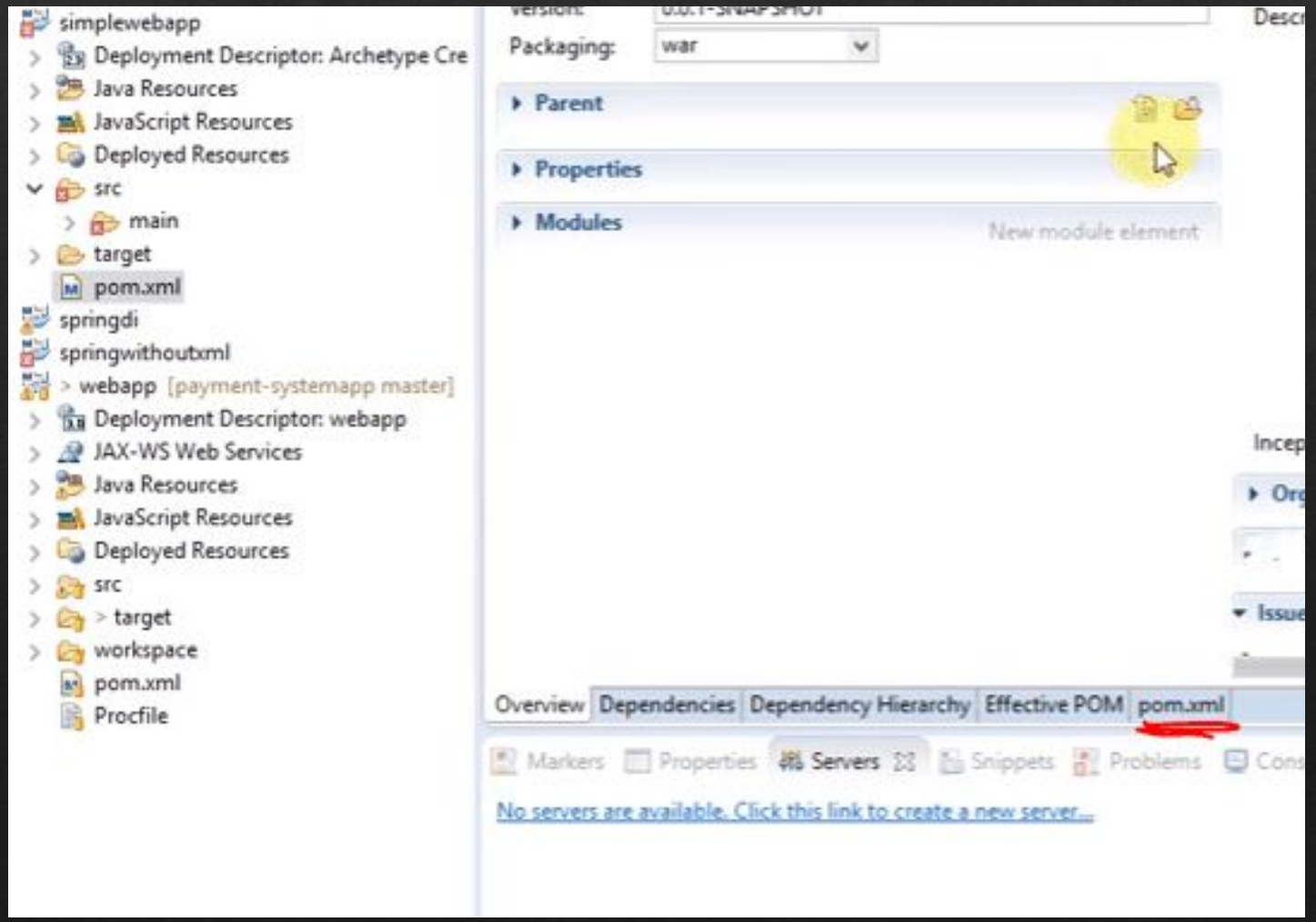
Далее появится окно ввода Group id, Artifact id. В строку Group id введите com.javamaster, а в строку Artifact id - например, simplewebapp.



Нажимаем Finish и видим, что в панели проектов нам создало новый проект с названием нашего Artifact id. В проекте много файлов и папок по умолчанию.

Далее (если присутствует красный крестик, теоретически приложение должно запускаться, но если по каким-то причинам ошибка, то тогда) –

Откройте файл pom.xml, перейдите на вкладку кода:

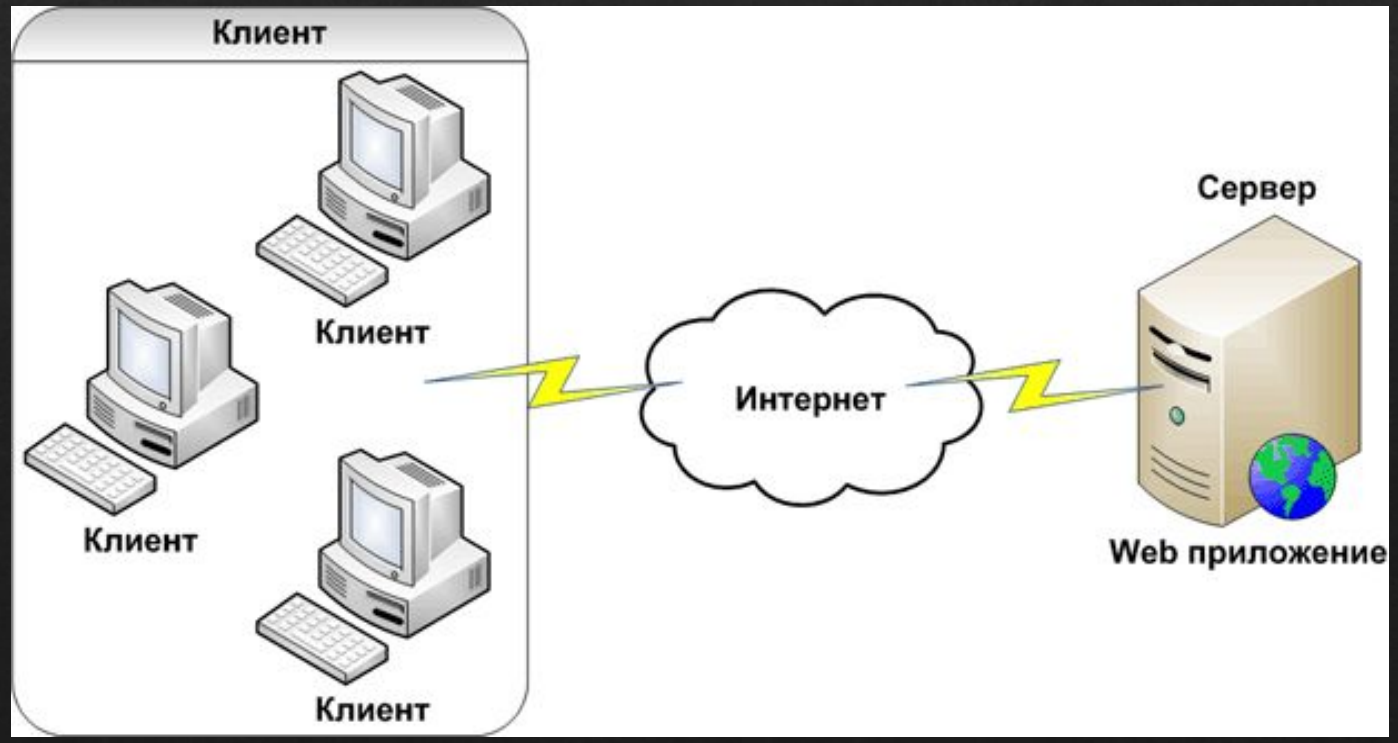


Далее в раздел dependencies (зависимости) нужно добавить (ниже основного dependency. То есть должно получиться dependency … dependency и dependency… dependency):

```
<dependency>
        <groupId>javax</groupId>
        <artifactId>javaee-web-api</artifactId>
        <version>6.0</version>
        <scope>provided</scope>
</dependency>
```

Этим действием мы добавили новую библиотеку к нашему проекту. Ничего не нужно скачивать: добавление нескольких строчек в pom.xml укажет программе автоматически загрузить данные библиотеки. Сохраните файл и проблема с ошибкой проекта должна исчезнуть.

Теперь необходимо рассмотреть, как работает веб и в частности, веб приложения.



На рисунке представлена схема работы интернет приложений: клиент, то есть Ваш браузер, посылает запрос по определенному адресу и порту;

через Интернет запрос доходит до Вашего сервера, на котором находится приложение;

обработанный запрос возвращается клиенту в виде ответа;

браузер интерпретирует ответ с сервера в понятный для человека вид картинок или текст.

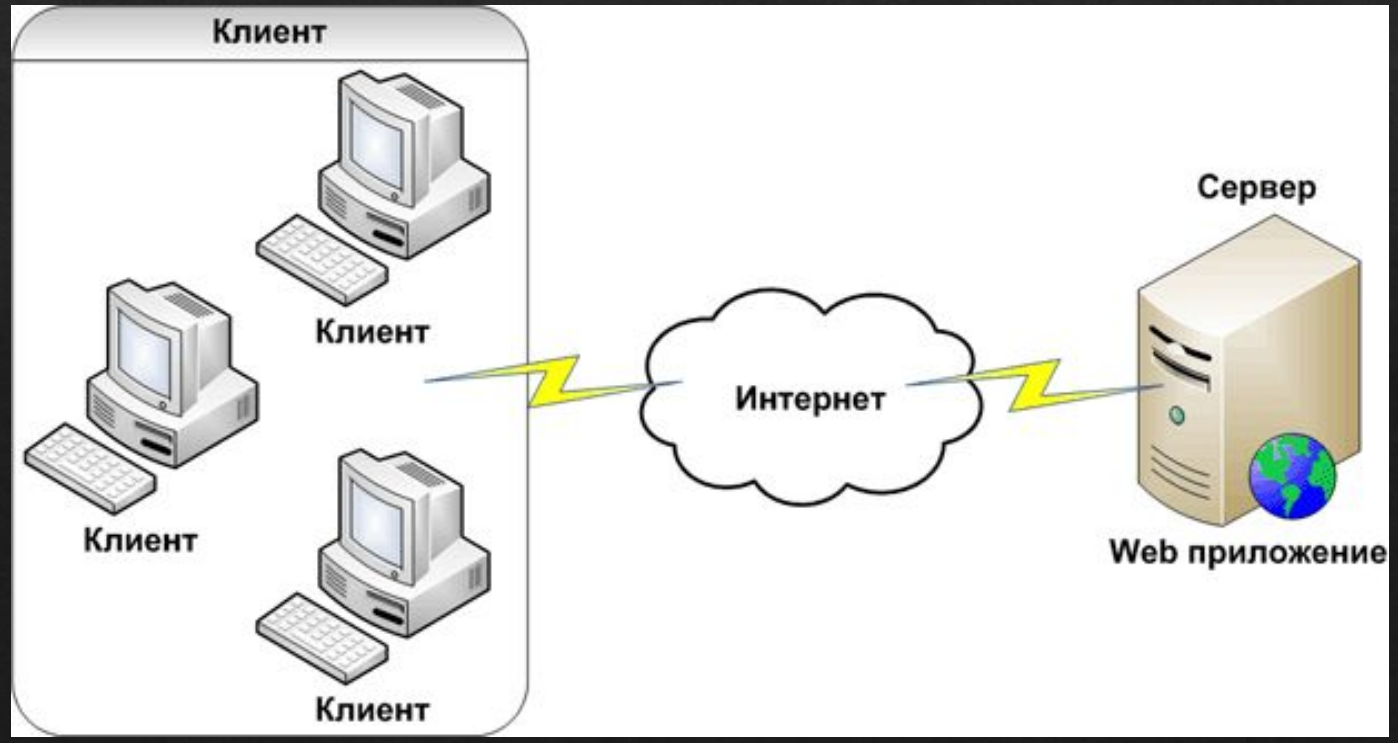
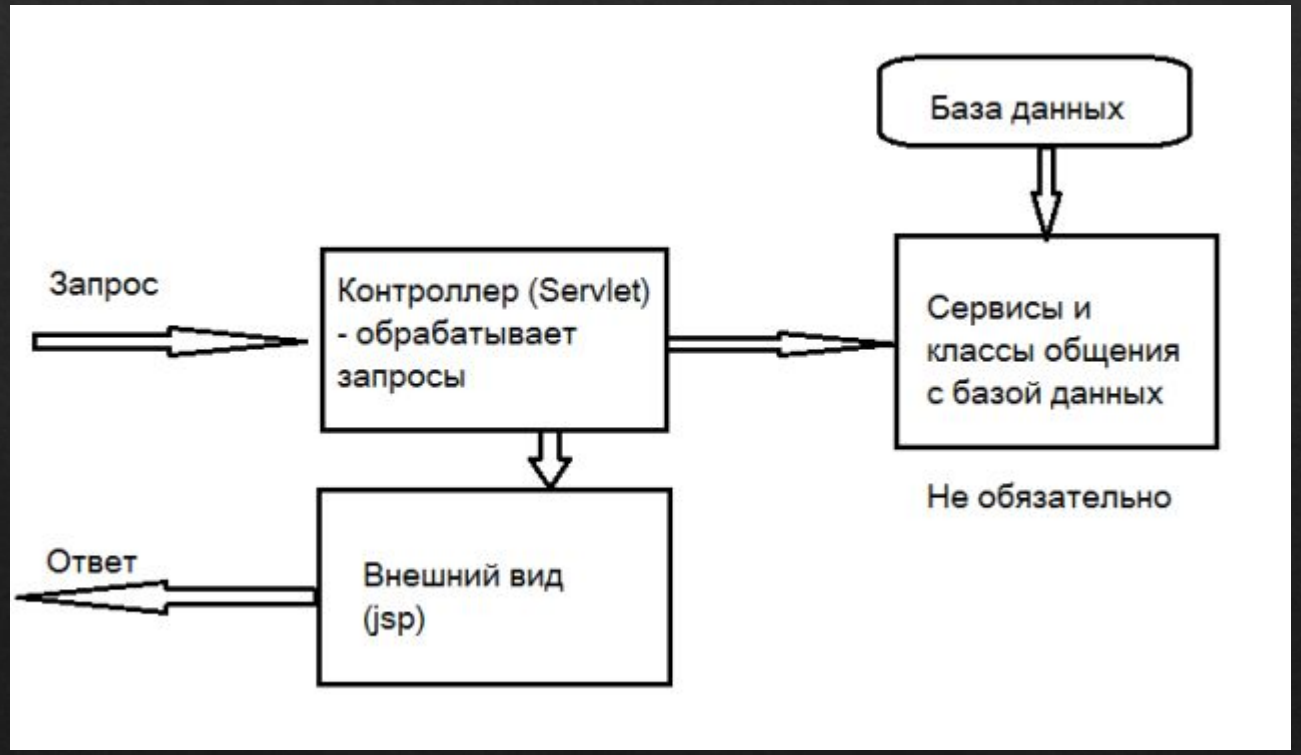


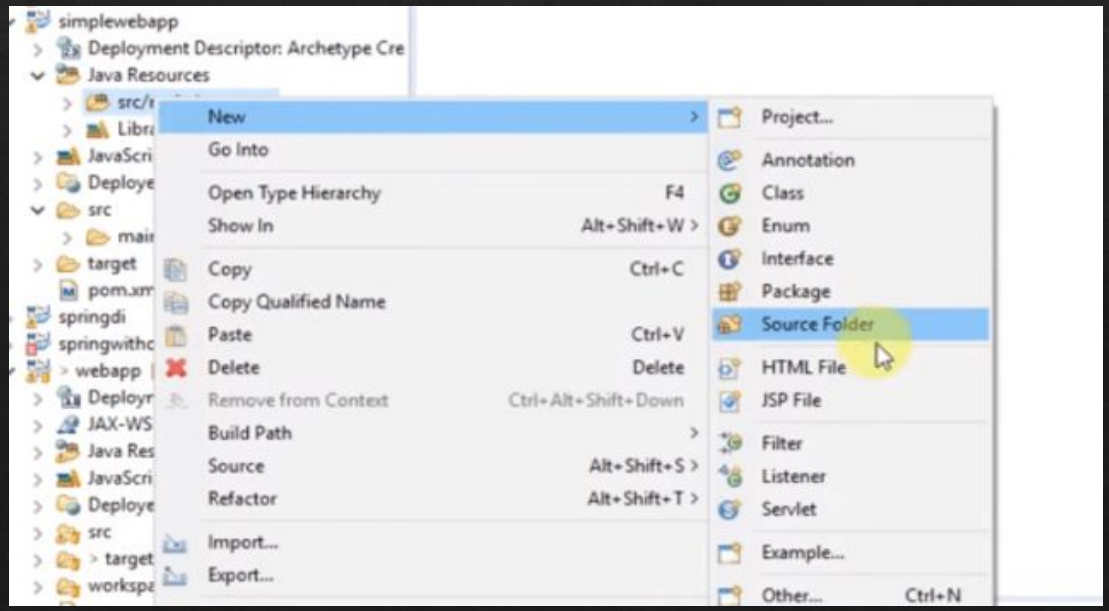
Схема веб программы:



Для распознавания запросов в Java есть такой механизм как сервлет (Servlet) — он может определить строку запроса и перенаправить его на jsp. jsp — технология, которая позволяет динамически генерировать веб страницы. По сути jsp очень похож на HTML с тем отличием, что в нём можно запускать Java код.

Сервлет — это класс, который унаследован от HttpServlet. В классе HttpServlet есть несколько методов по обработке запросов. Нас пока будут интересовать doGet и doPost, который обрабатывают соответственно гет и пост запросы.

В проекте нужно найти папку src/main/java. Если таковой нет, нужно её создать: правой кнопкой на src -> new -> source folder.



Далее нужно добавить в эту папку новый пакет (package). например - com.javamaster.controller

Теперь создаём Servlet (с кодом ниже), например - HomeServlet.

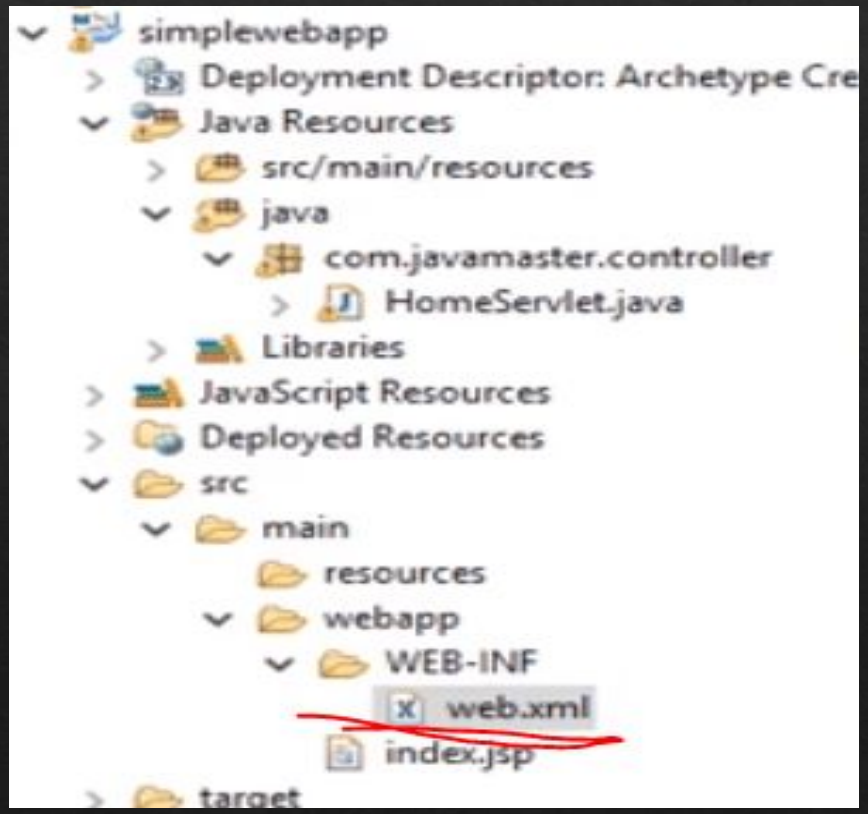
```
package com.javamaster.controller;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class HomeServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    }}
```

В коде только один метод: doGet на данном этапе этого достаточно.

Теперь нужно дать нашей программе понять, что класс HomeServlet является контроллером и что все запросы нужно отправлять на него.

Теперь нужно дать нашей программе понять, что класс HomeServlet является контроллером и что все запросы нужно отправлять на него.

Для этого у нас есть web.xml — файл настроек нашей программы. Его можно найти в папке WEB-INF.



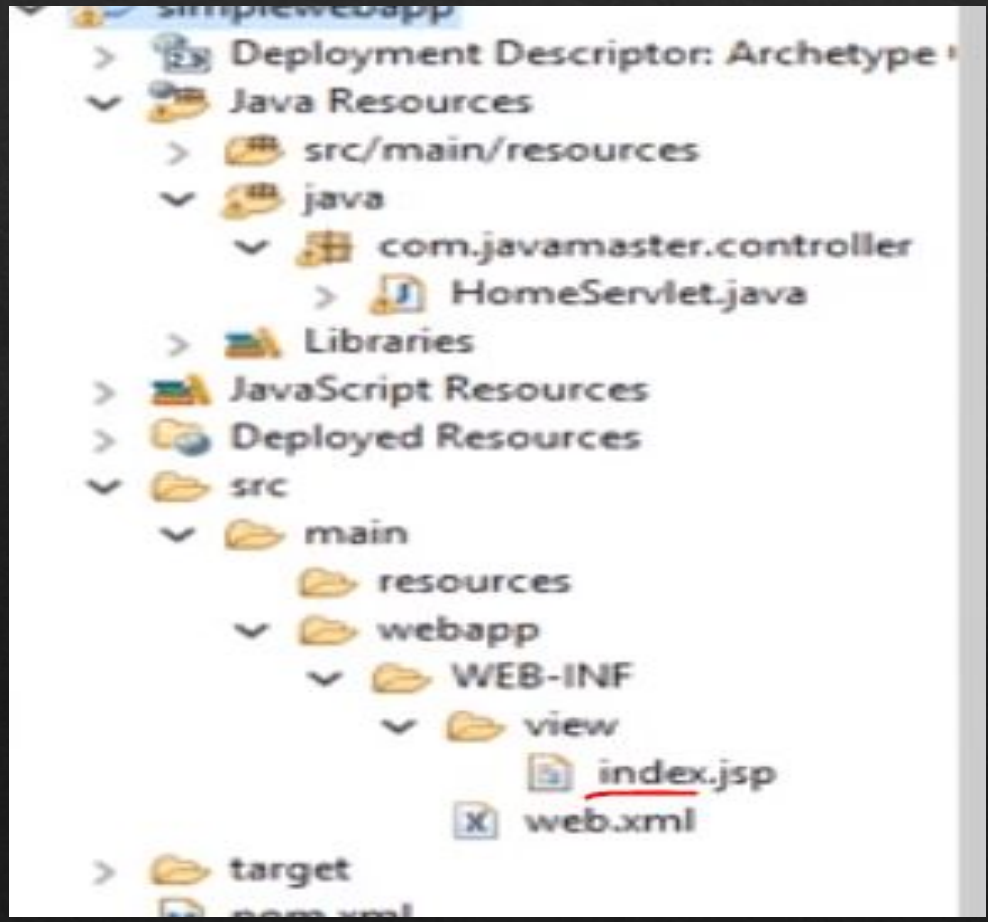
И заменим код:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
    version="3.0">
  <display-name>Archetype Created Web Application</display-name>
  <servlet>
        <description></description>
        <display-name>HomeServlet</display-name>
        <servlet-name>HomeServlet</servlet-name>
        <servlet-class>com.javamaster.controller.HomeServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HomeServlet</servlet-name>
        <url-pattern>/</url-pattern></servlet-mapping></web-app>
```

В коде, мы указали что передаём управление урл путями нашему классу HomeServlet.

Теперь осталось добавить файл, который будет отвечать за внешний вид. Система уже создала нам файл index.jsp и проделала всю работу за нас.

Добавьте новую папку view в папку WEB-INF. Когда появиться много файлов, чтобы не запутаться мы поместим все файлы, которые отвечают за внешний вид в папку view. Переместите файл index.jsp в только что созданную папку.



Теперь осталось отловить запрос и перенаправить его на нашу страницу index.jsp. Для этого в сервлете метод doGet нужно немного изменить.

```
public  class HomeServlet extends HttpServlet {
 @Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
 throws ServletException, IOException {
    String path = request.getServletPath();
    if (path.equals("/")){

request.getRequestDispatcher("/WEB-INF/view/ind
ex.jsp").forward(request, response);
    }
    else if (path.equals("/welcome")){

request.getRequestDispatcher("/WEB-INF/view/wel
come.jsp").forward(request, response);
       }
 }
}
```

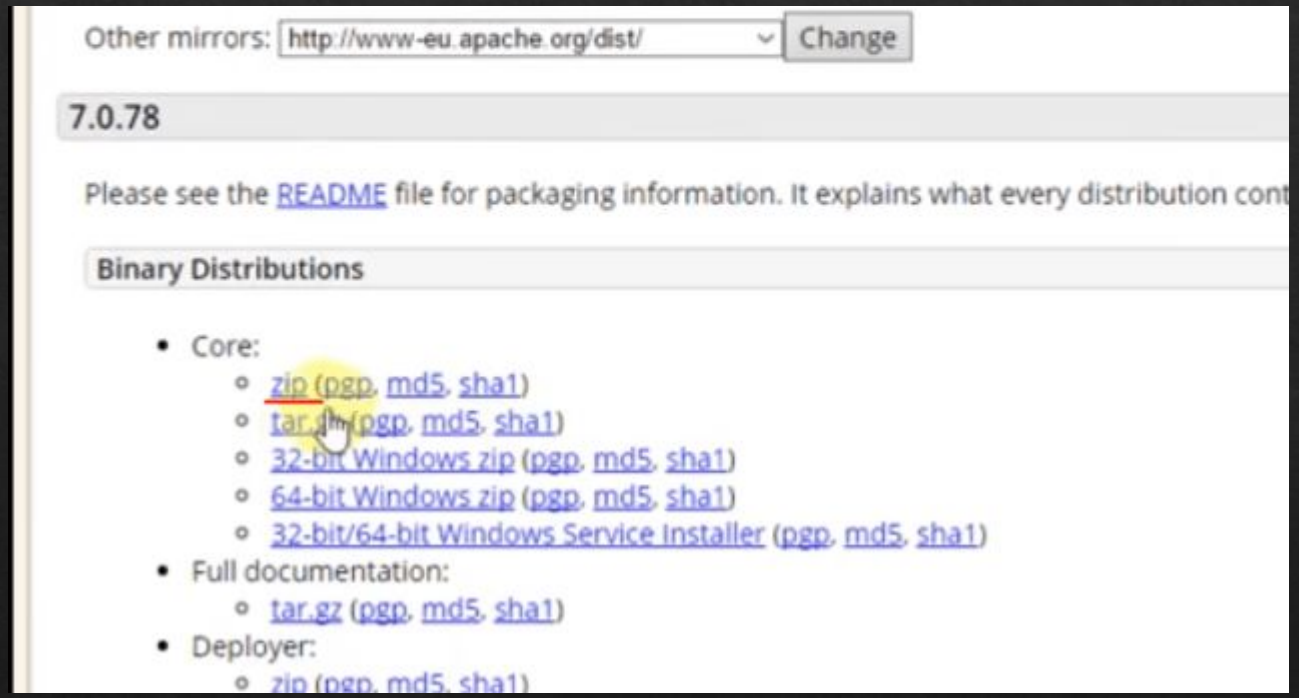
если запрос равен /, тогда указываем путь, куда перенаправить запрос: на страницу индекс, если же запрос равен /welcome тогда, перенаправить на страницу приветствия.

Теперь осталось только запустить наше приложение и поднять сервер.

Нам нужен контейнер сервлетов, который сможет управлять нашей программой и который реализует Java Servlets и Java Server Pages (jsp) спецификации. Для наших нужд отлично подойдет Apache Tomcat — он бесплатный и очень удобный в использовании.

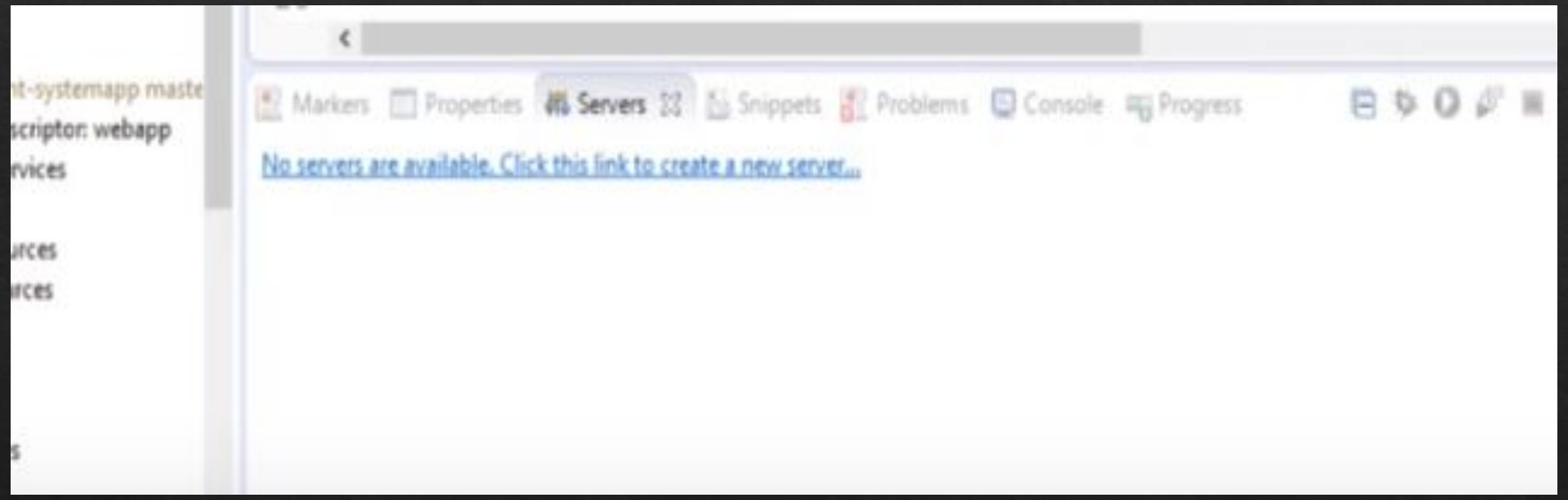
Apache Tomcat нужно скачать с официального сайта, распаковать и указать в настройках Eclipse путь к распакованному архиву. Сделаем все по порядку.

Качаем архив с сайта:



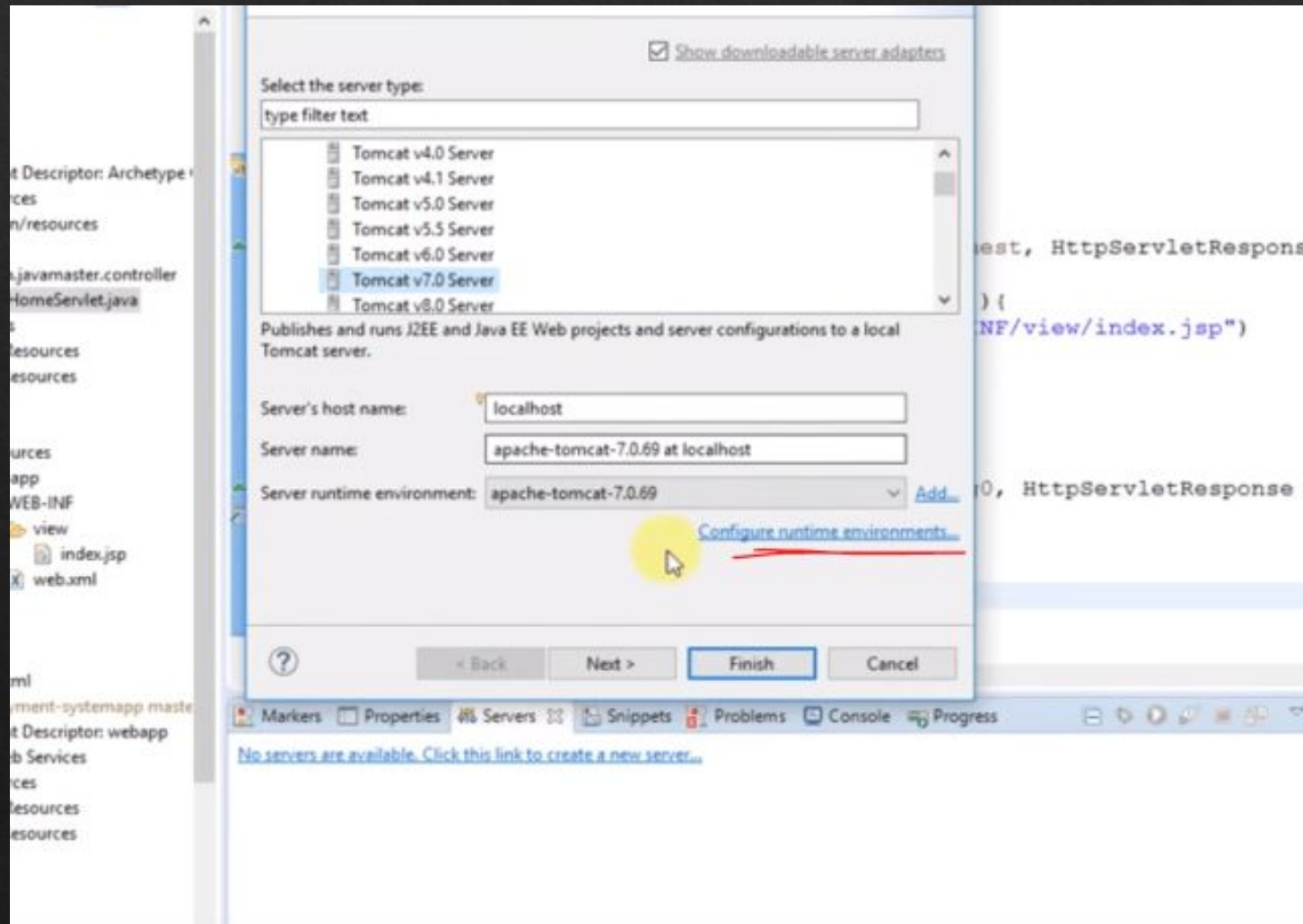
Распаковываем его в любое место папку на диске.

Переходим в Eclipse и во вкладке сервера нажимаем создать новый:



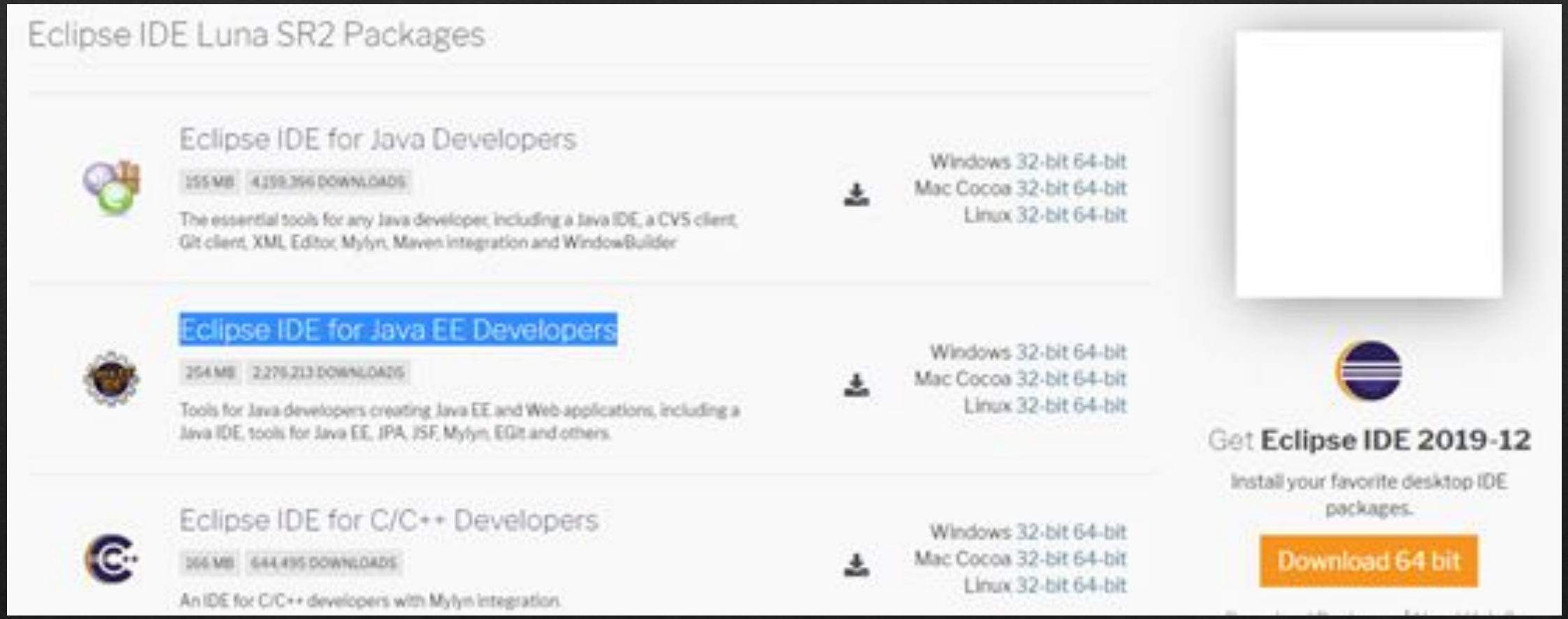
Если такой вкладки у Вас нет, её нужно добавить: Window -> View -> Servers

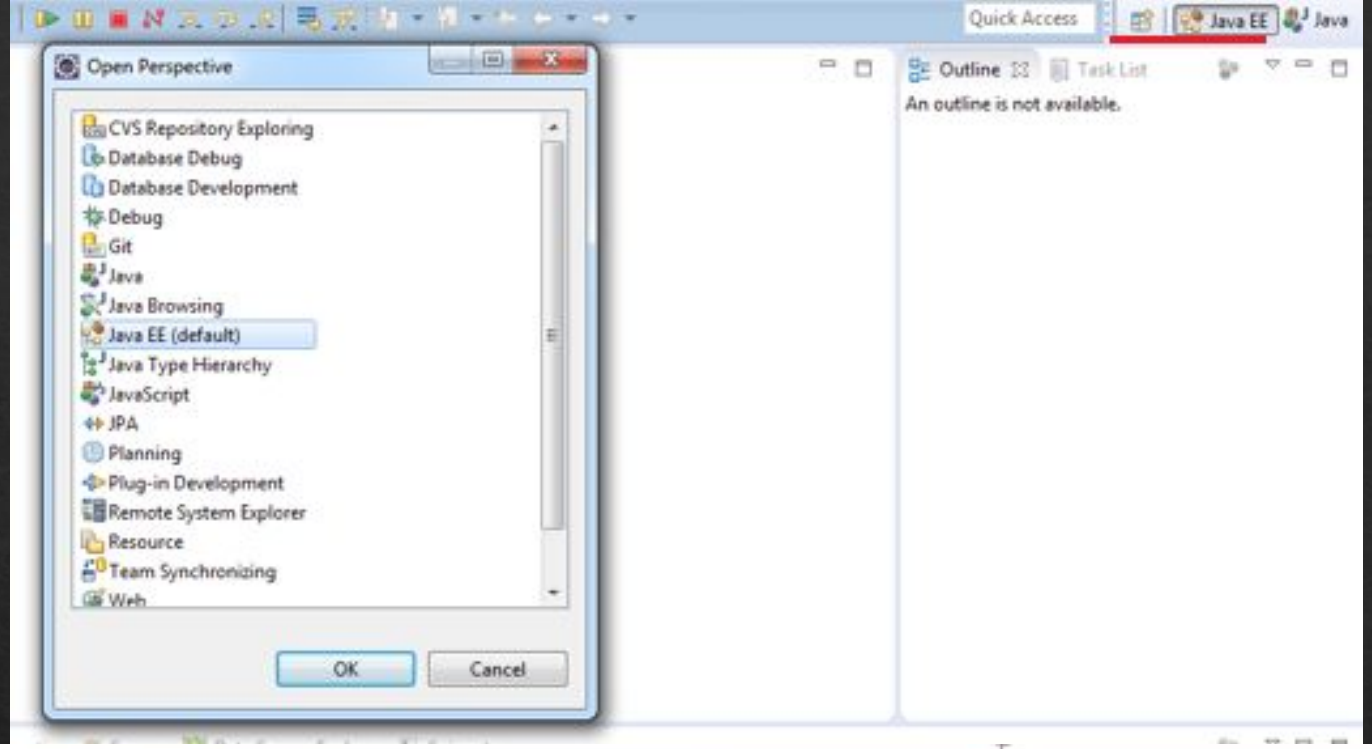
Теперь нужно выбрать нужный нам вариант из предложенных серверов, указать путь к нашему и готово.

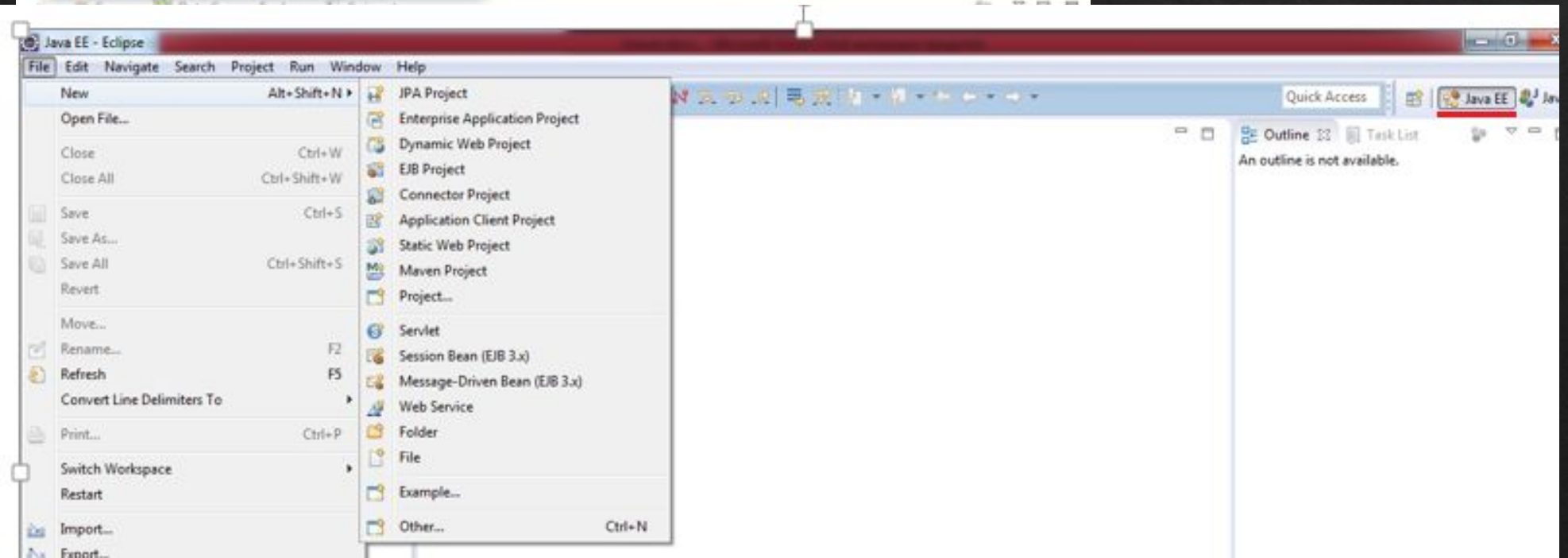


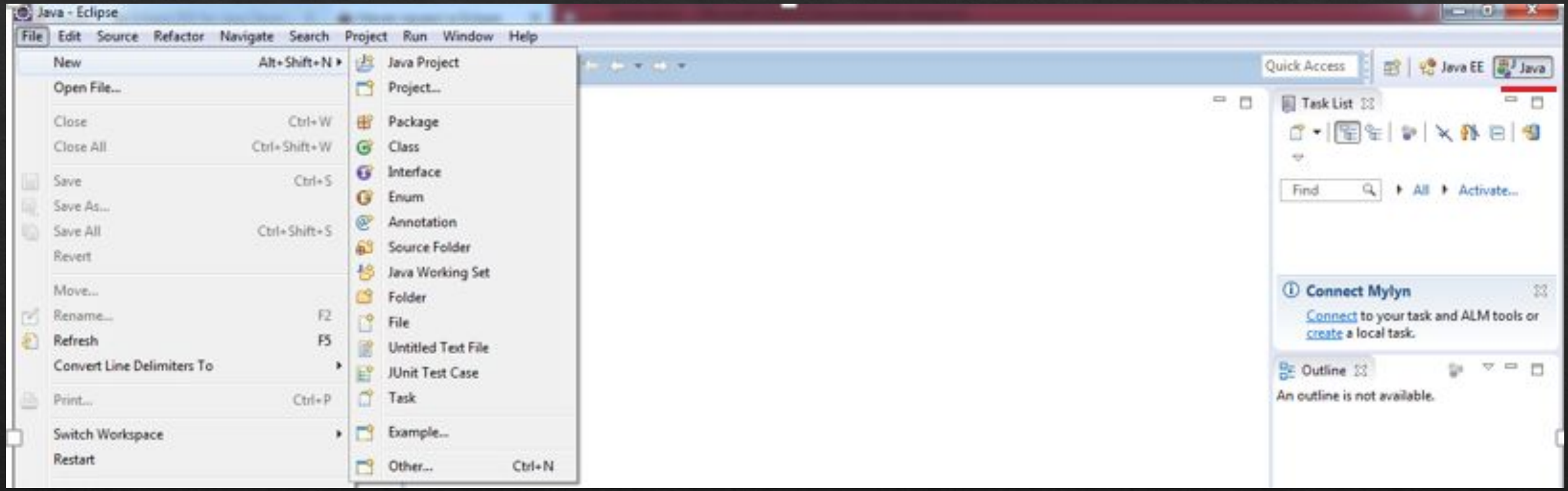
Переходим к нашему проекту. В разделе Project explorer нужно нажать правой клавишей мыши и выбрать: run as -> run on server. Далее в окошке выбрать добавленный сервер подождать, пока проект запуститься. Если у Вас нет опции run on server, её нужно добавить: run as -> run configuration -> run on server.

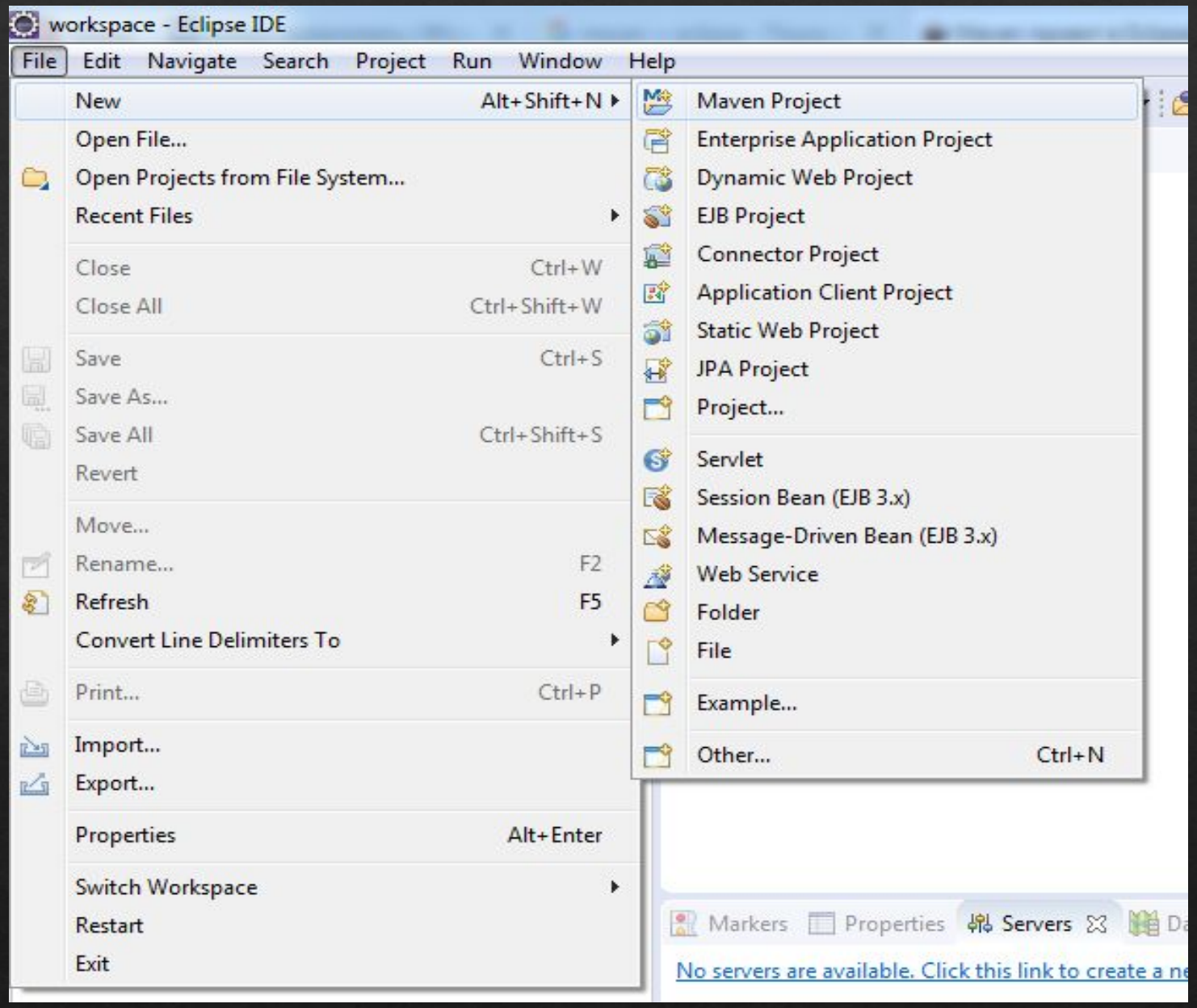
Информация для размышления
Нужен Java EE Developers




Quick Access
Java EE
Java
Open Perspective
CVS Repository Exploring
Database Debug
Database Development
Debug
Git
Java
Java Browsing
Java EE (default)
Java Type Hierarchy
JavaScript
JPA
Planning
Plug-in Development
Remote System Explorer
Resource
Team Synchronizing
Web
OK
Cancel
Outline
Task List
An outline is not available.

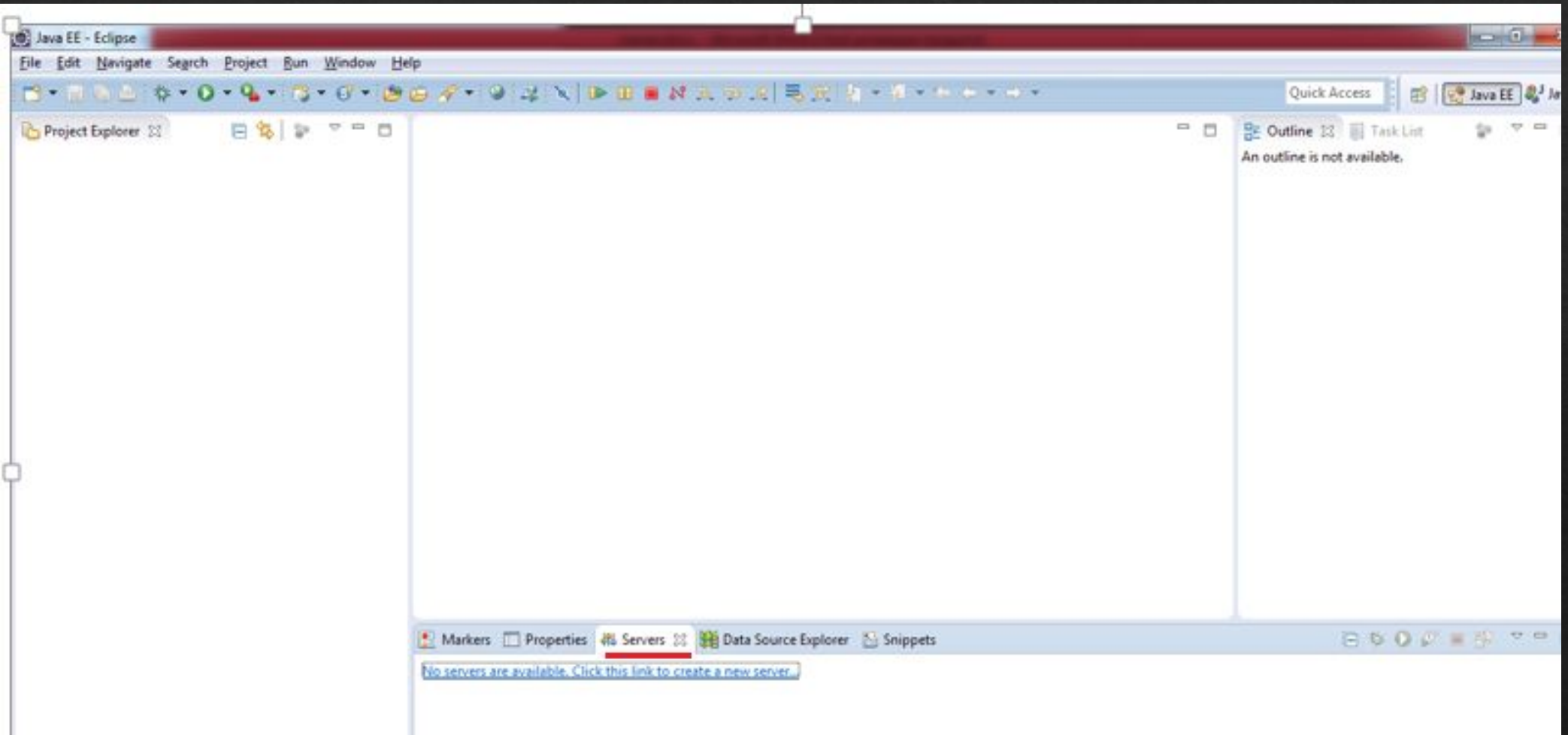

Java EE - Eclipse
File
Edit
Navigate
Search
Project
Run
Window
Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Restart
Import...
Export...
JPA Project
Enterprise Application Project
Dynamic Web Project
EJB Project
Connector Project
Application Client Project
Static Web Project
Maven Project
Project...
Servlet
Session Bean (EJB 3.x)
Message-Driven Bean (EJB 3.x)
Web Service
Folder
File
Example...
Other... Ctrl+N
Quick Access
Java EE
Outline
Task List
An outline is not available.

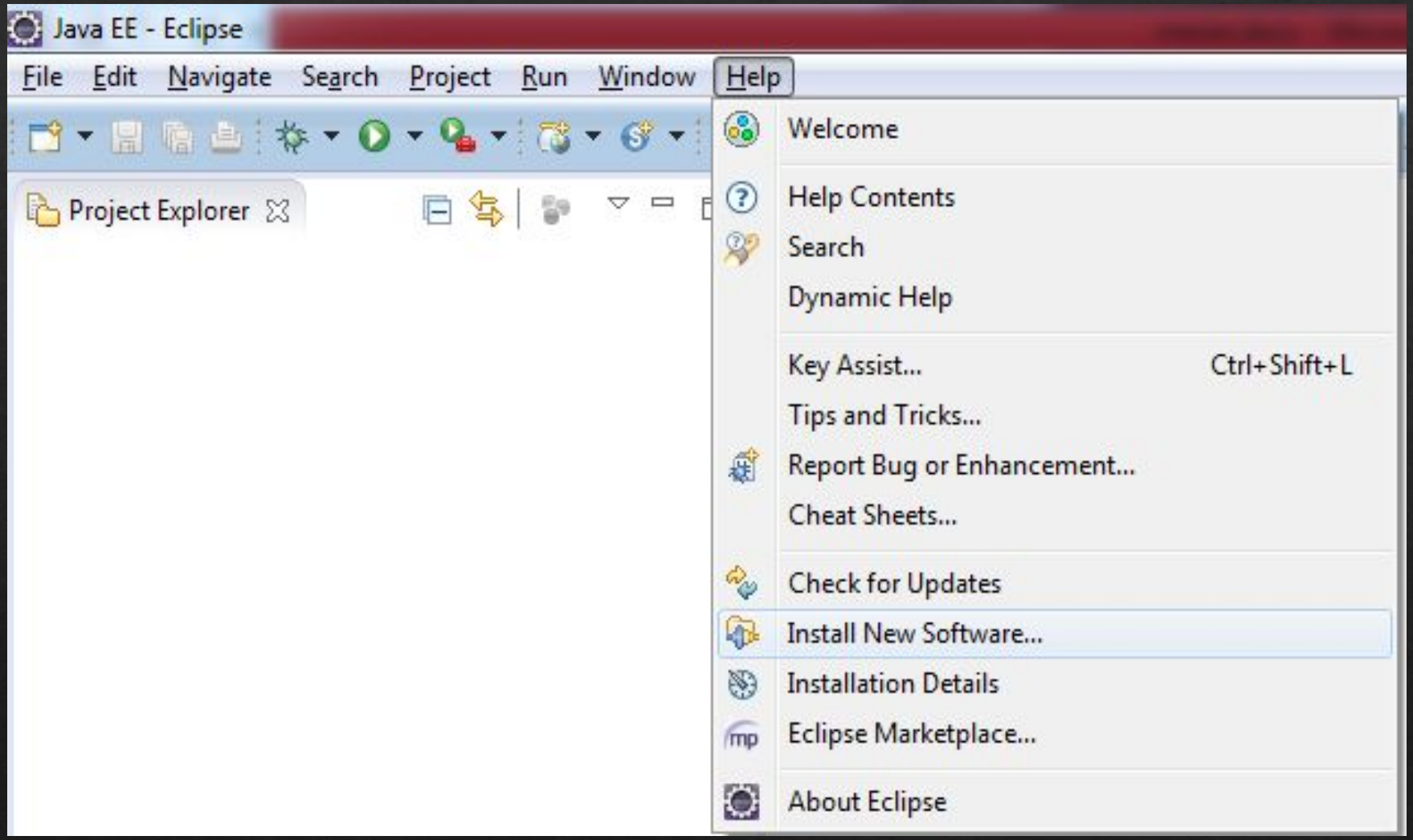

Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Restart
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Task
Example...
Other... Ctrl+N
Quick Access
Java EE
Java
Task List
Find
All
Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
An outline is not available.

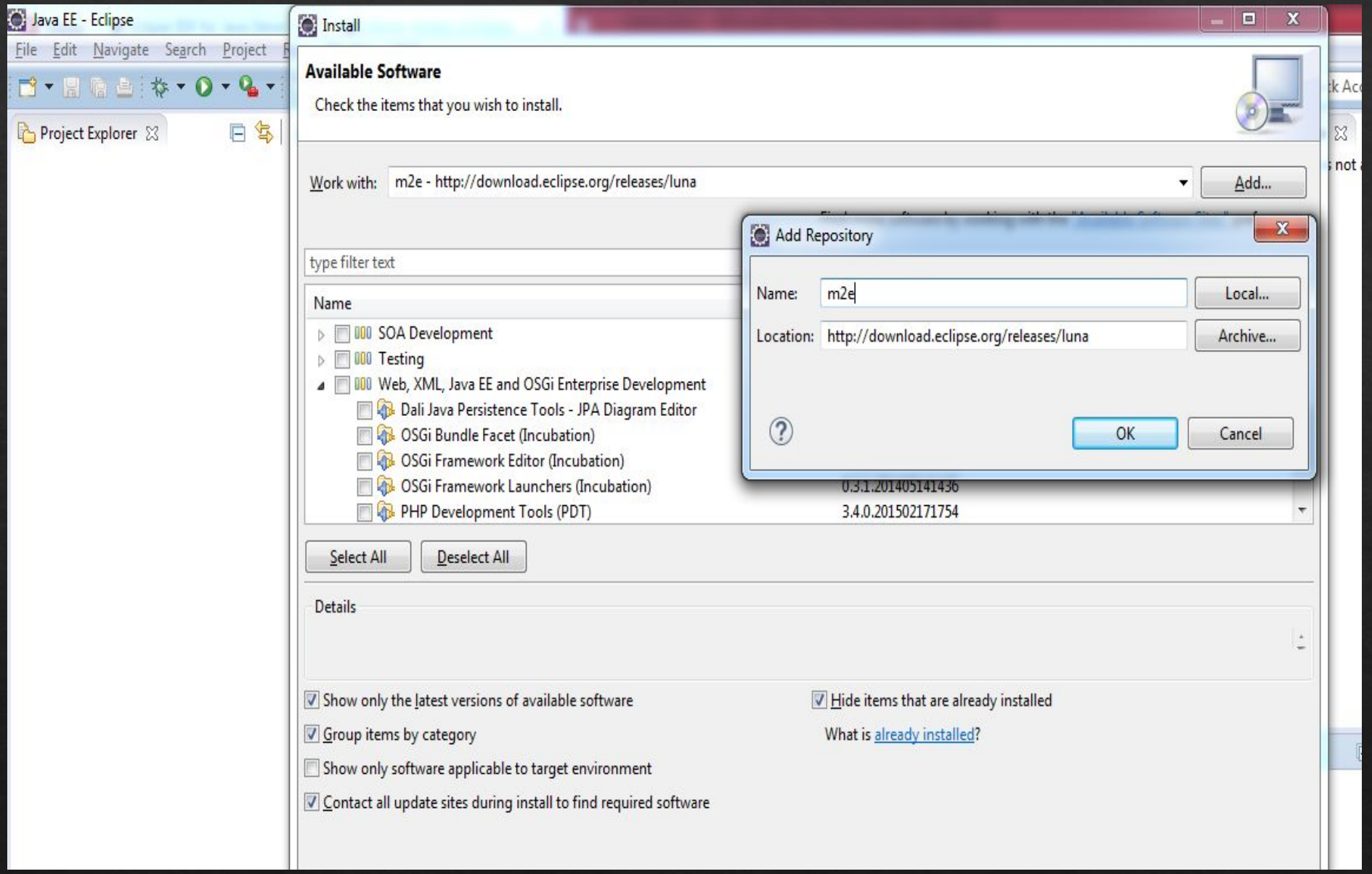

workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Open Projects from File System...
Recent Files
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Import...
Export...
Properties Alt+Enter
Switch Workspace
Restart
Exit
Maven Project
Enterprise Application Project
Dynamic Web Project
EJB Project
Connector Project
Application Client Project
Static Web Project
JPA Project
Project...
Servlet
Session Bean (EJB 3.x)
Message-Driven Bean (EJB 3.x)
Web Service
Folder
File
Example...
Other... Ctrl+N
Markers Properties Servers
No servers are available. Click this link to create a ne

+есть Servers.

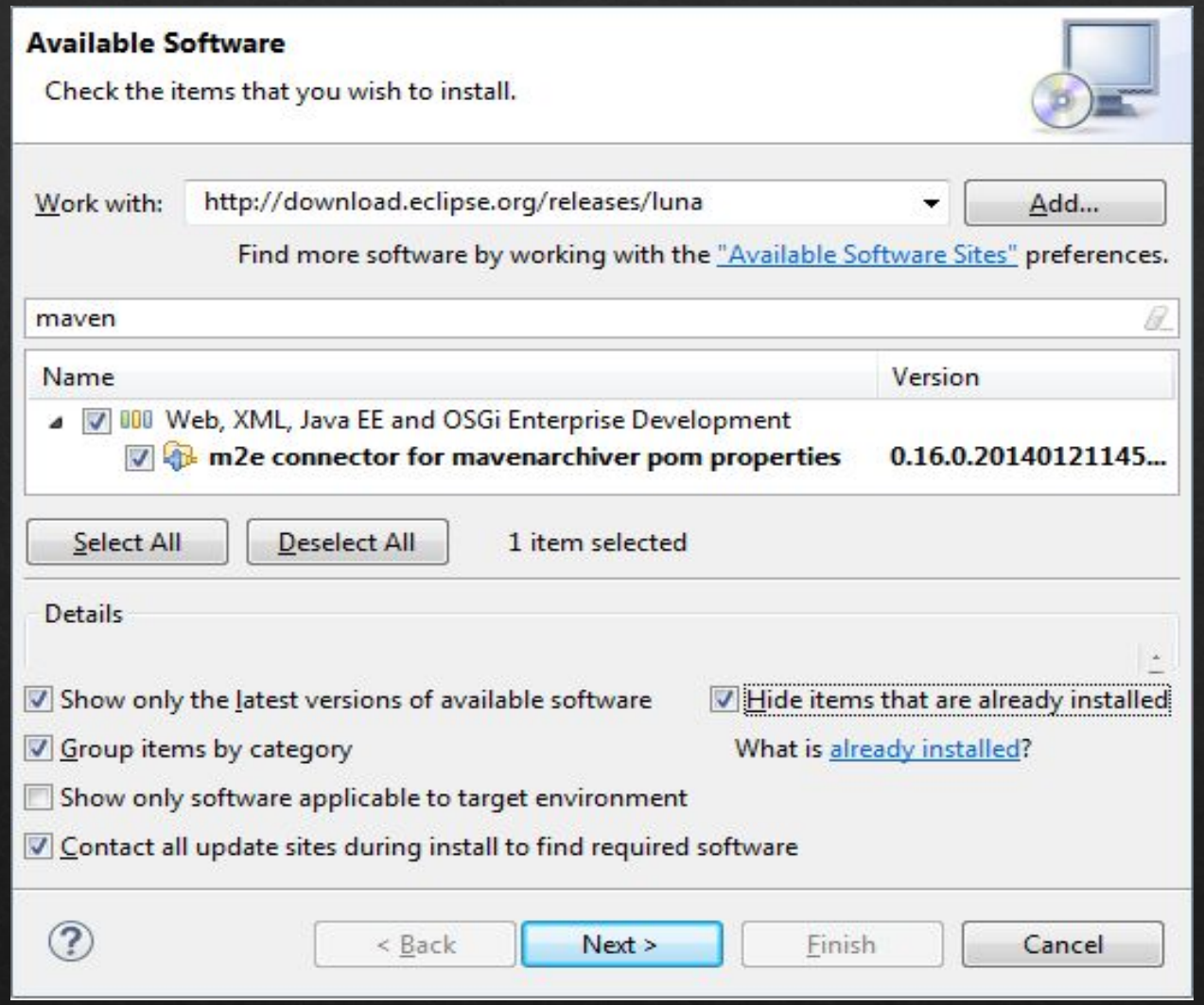
Если нет, то открыть его в настройках



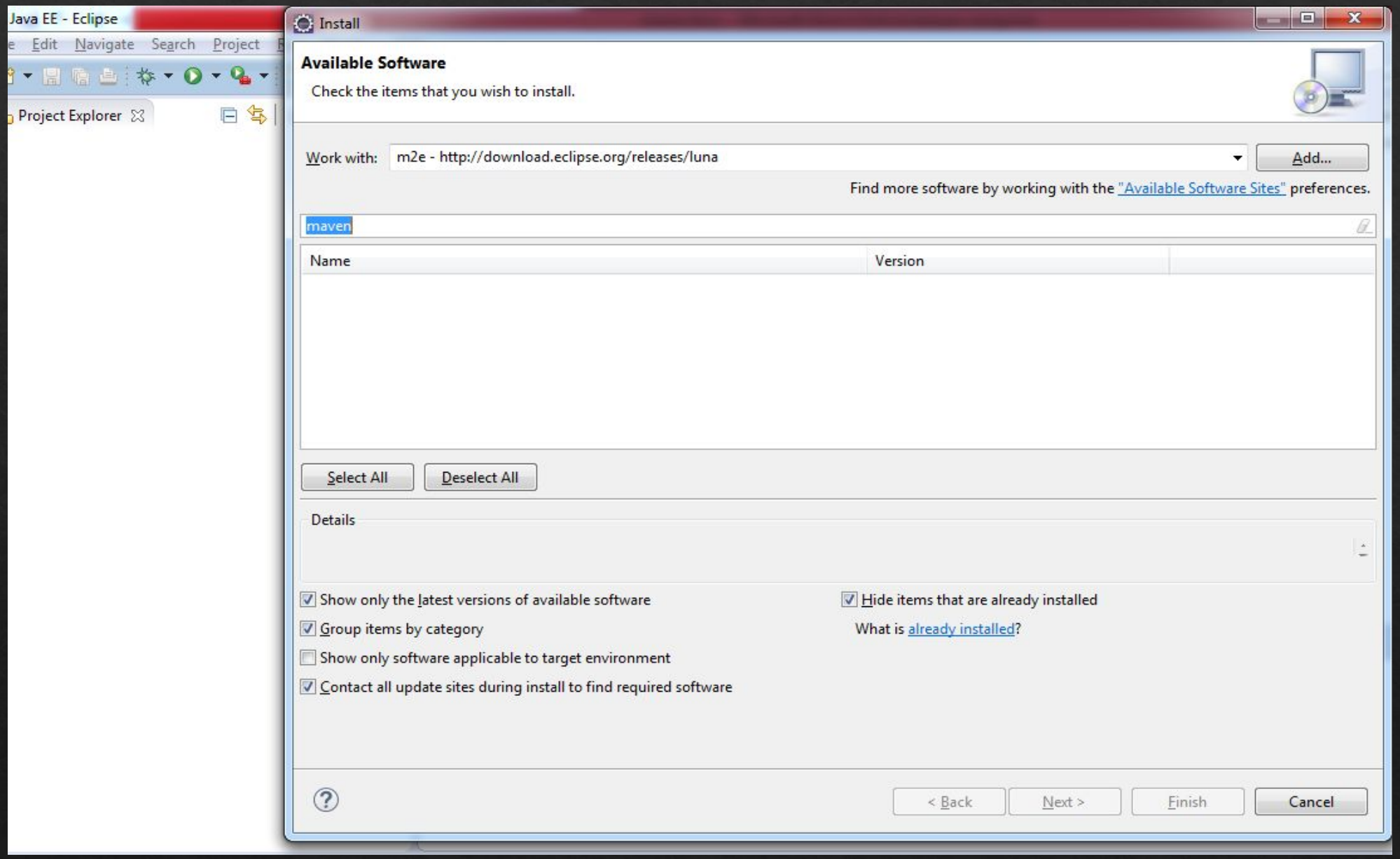

Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer
Welcome
Help Contents
Search
Dynamic Help
Key Assist... Ctrl+Shift+L
Tips and Tricks...
Report Bug or Enhancement...
Cheat Sheets...
Check for Updates
Install New Software...
Installation Details
Eclipse Marketplace...
About Eclipse

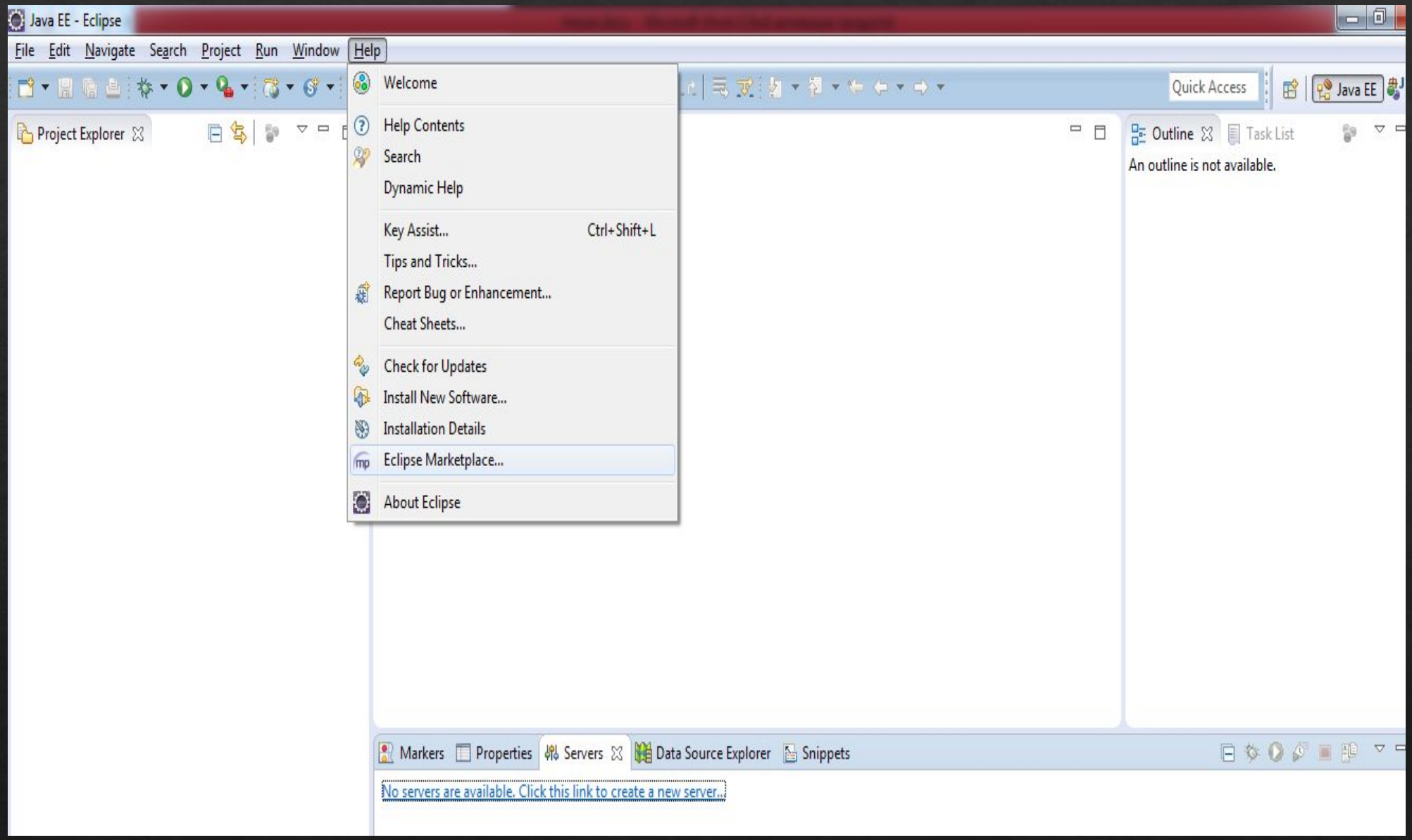

Java EE - Eclipse
File Edit Navigate Search Project
Project Explorer
Install
Available Software
Check the items that you wish to install.
Work with: m2e - http://download.eclipse.org/releases/luna
Add...
type filter text
Name
SOA Development
Testing
Web, XML, Java EE and OSGi Enterprise Development
Dali Java Persistence Tools - JPA Diagram Editor
OSGi Bundle Facet (Incubation)
OSGi Framework Editor (Incubation)
OSGi Framework Launchers (Incubation)
PHP Development Tools (PDT)
0.3.1.201405141436
3.4.0.201502171754
Select All
Deselect All
Details
Show only the latest versions of available software
Group items by category
Show only software applicable to target environment
Contact all update sites during install to find required software
Hide items that are already installed
What is already installed?
Add Repository
Name: m2e
Local...
Location: http://download.eclipse.org/releases/luna
Archive...
OK
Cancel

У меня нет то что
надо

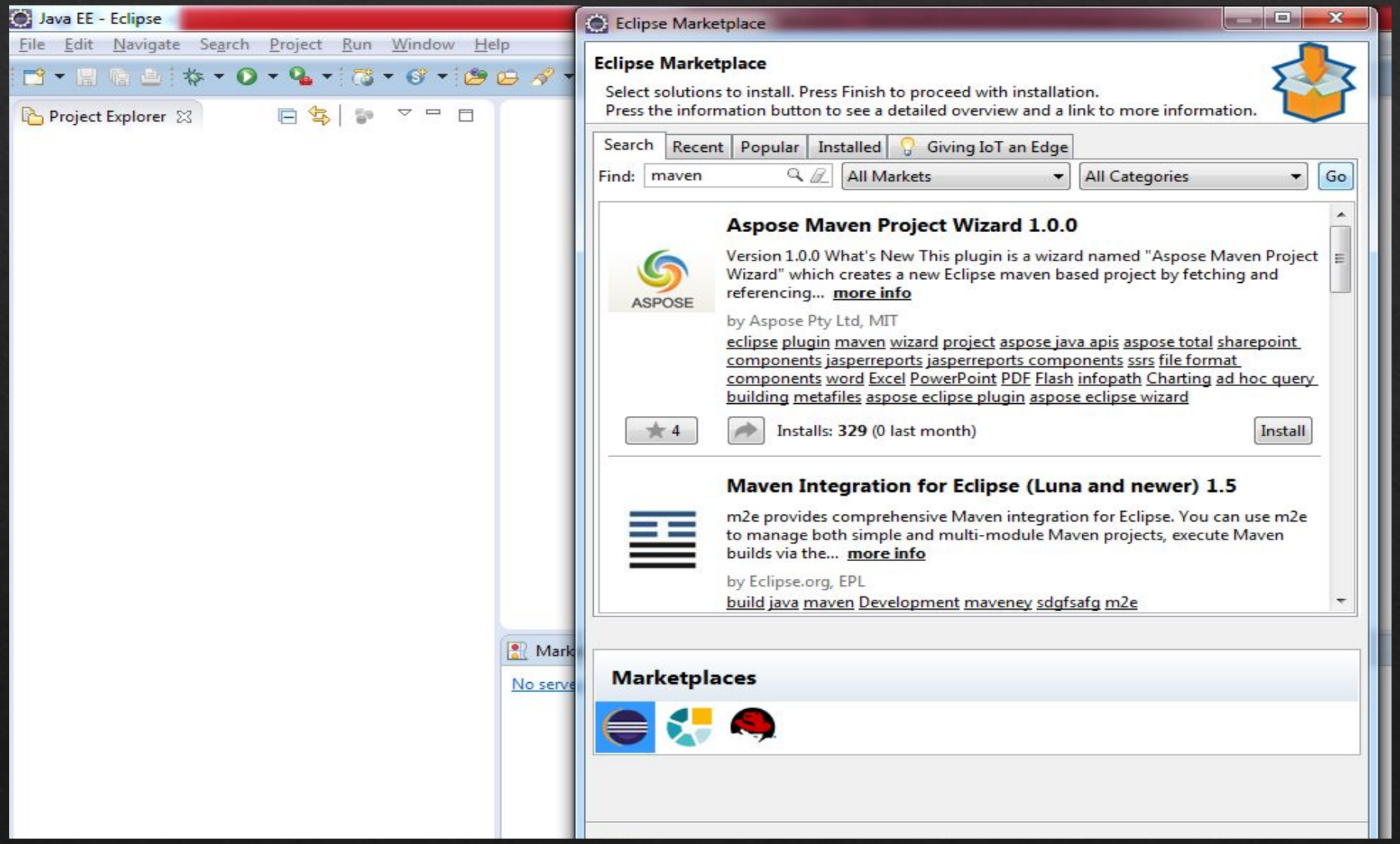


Поиск

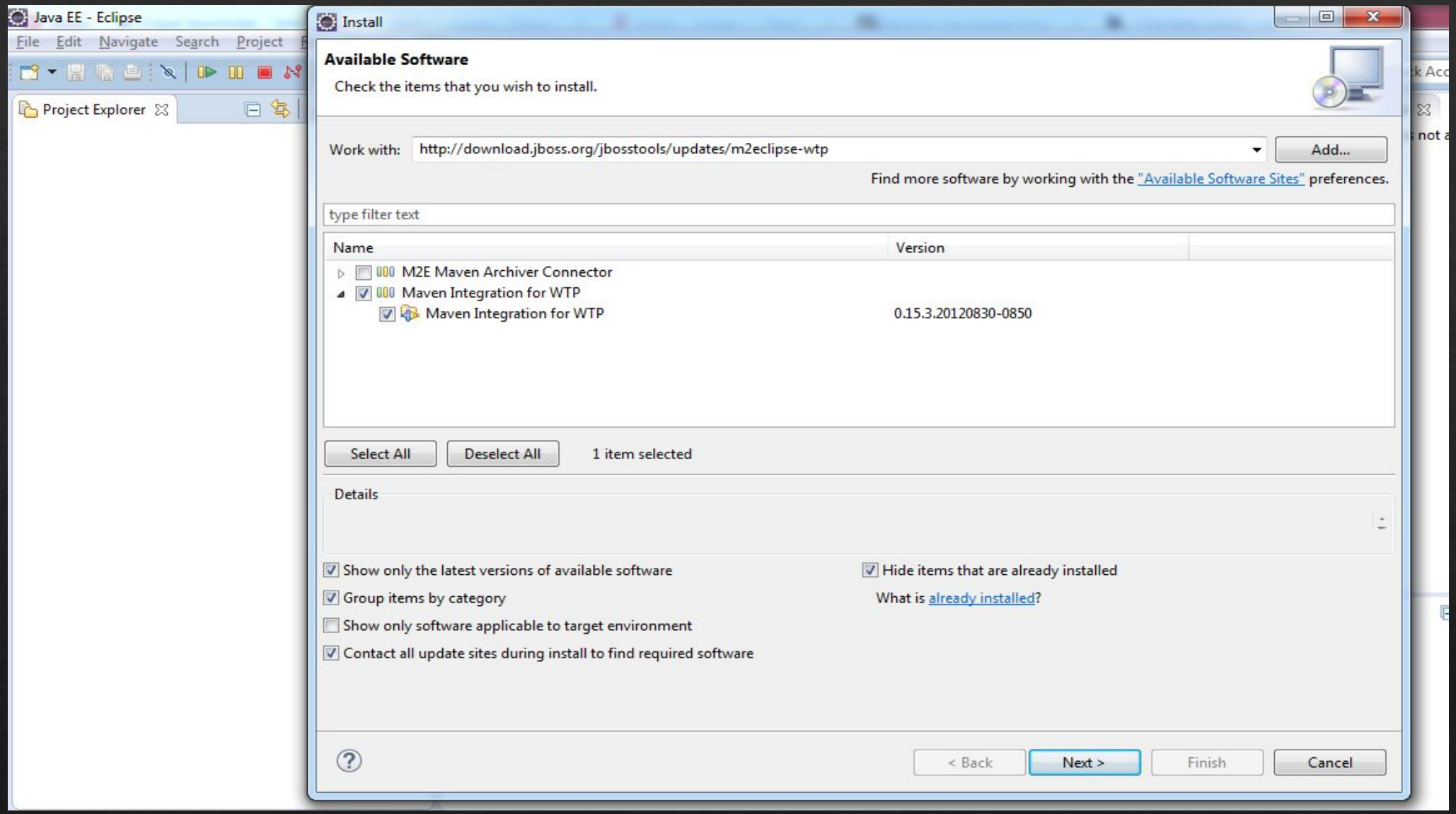

Java EE - Eclipse
Edit Navigate Search Project
Project Explorer
Install
Available Software
Check the items that you wish to install.
Work with: m2e - http://download.eclipse.org/releases/luna
Add...
Find more software by working with the "Available Software Sites" preferences.
maven
Name
Version
Select All
Deselect All
Details
Show only the latest versions of available software
Group items by category
Show only software applicable to target environment
Contact all update sites during install to find required software
Hide items that are already installed
What is already installed?
< Back
Next >
Finish
Cancel

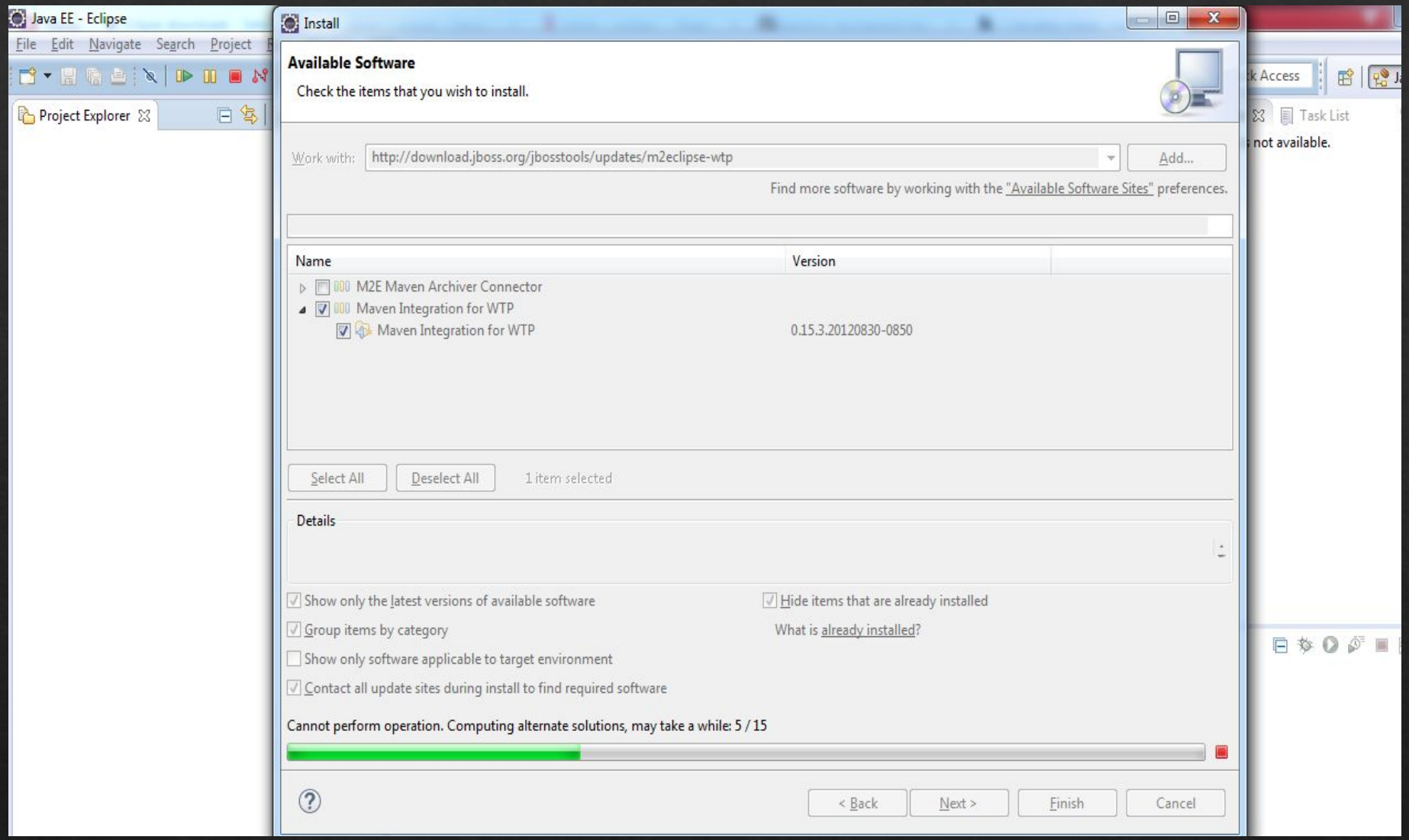

Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
Welcome
Help Contents
Search
Dynamic Help
Key Assist... Ctrl+Shift+L
Tips and Tricks...
Report Bug or Enhancement...
Cheat Sheets...
Check for Updates
Install New Software...
Installation Details
Eclipse Marketplace...
About Eclipse
Quick Access
Java EE
Project Explorer
Outline
Task List
An outline is not available.
Markers
Properties
Servers
Data Source Explorer
Snippets
No servers are available. Click this link to create a new server...

Maven integration



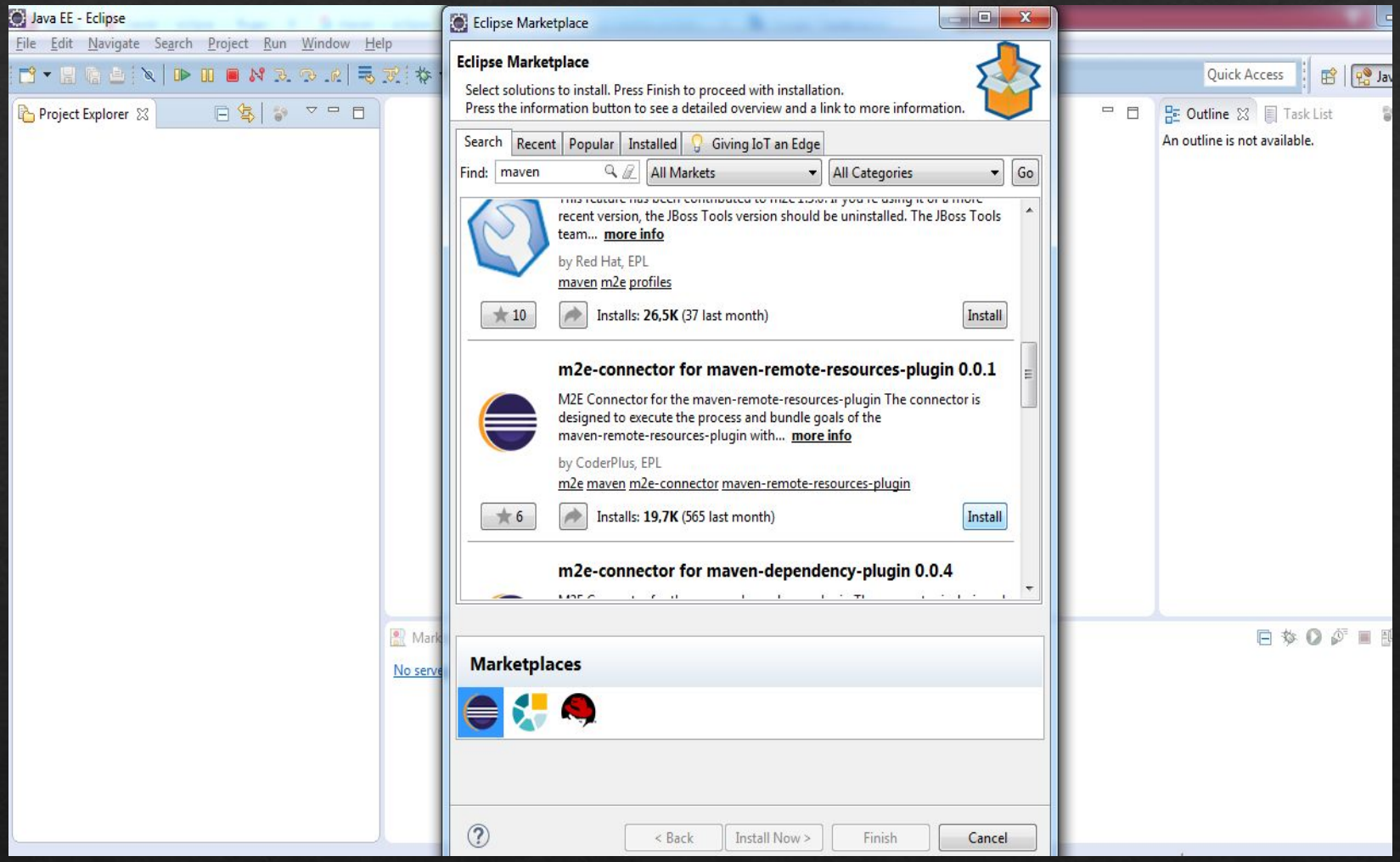
Если Maven проект не создаётся, то возможно:





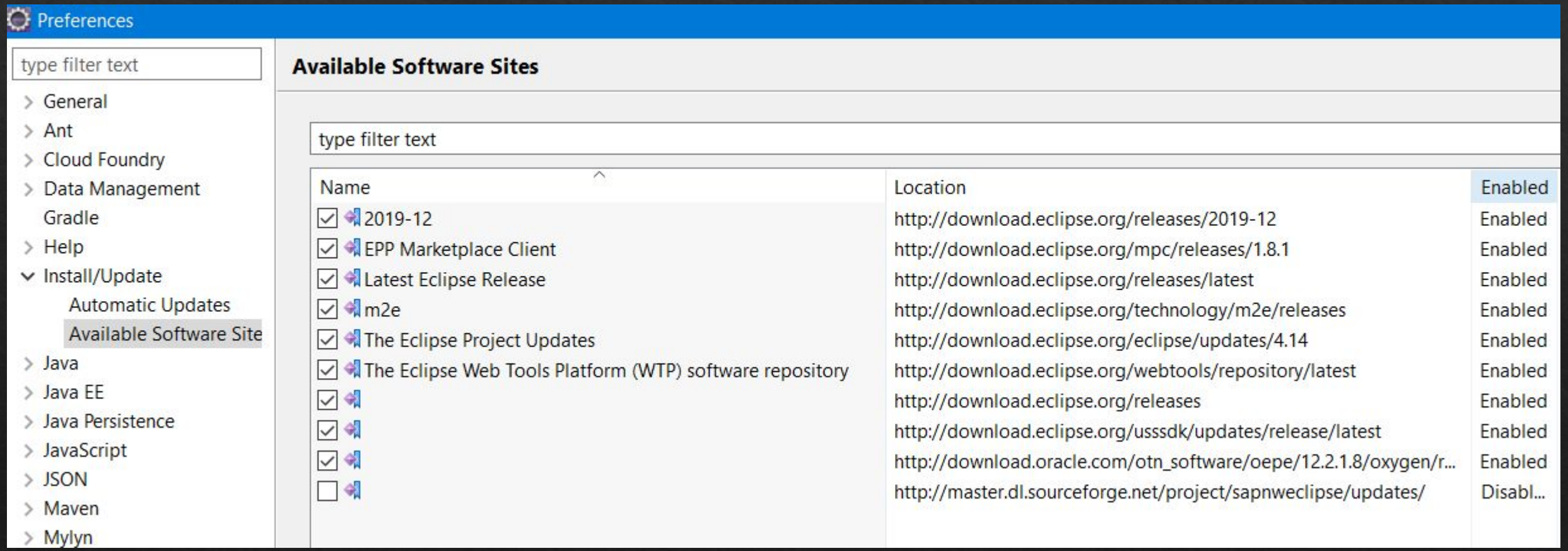
Reboot Eclipse

Если ещё косяк, то help – eclipse Marketplace. Из списка установить нужно m2e-connetcor и тд

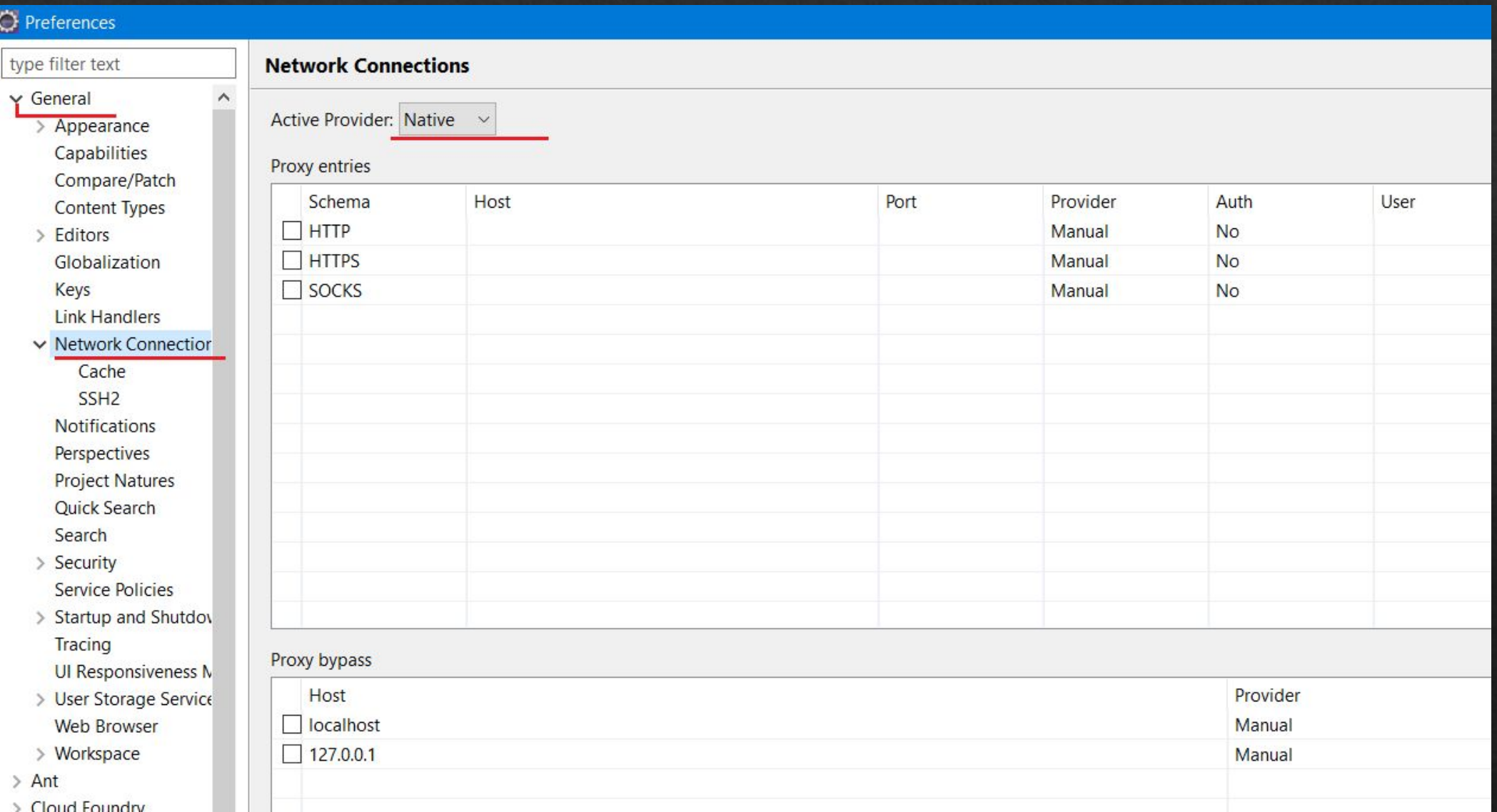


Если Eclipse ругается на директорию, сервер и прочее (вся что связано с сетью)

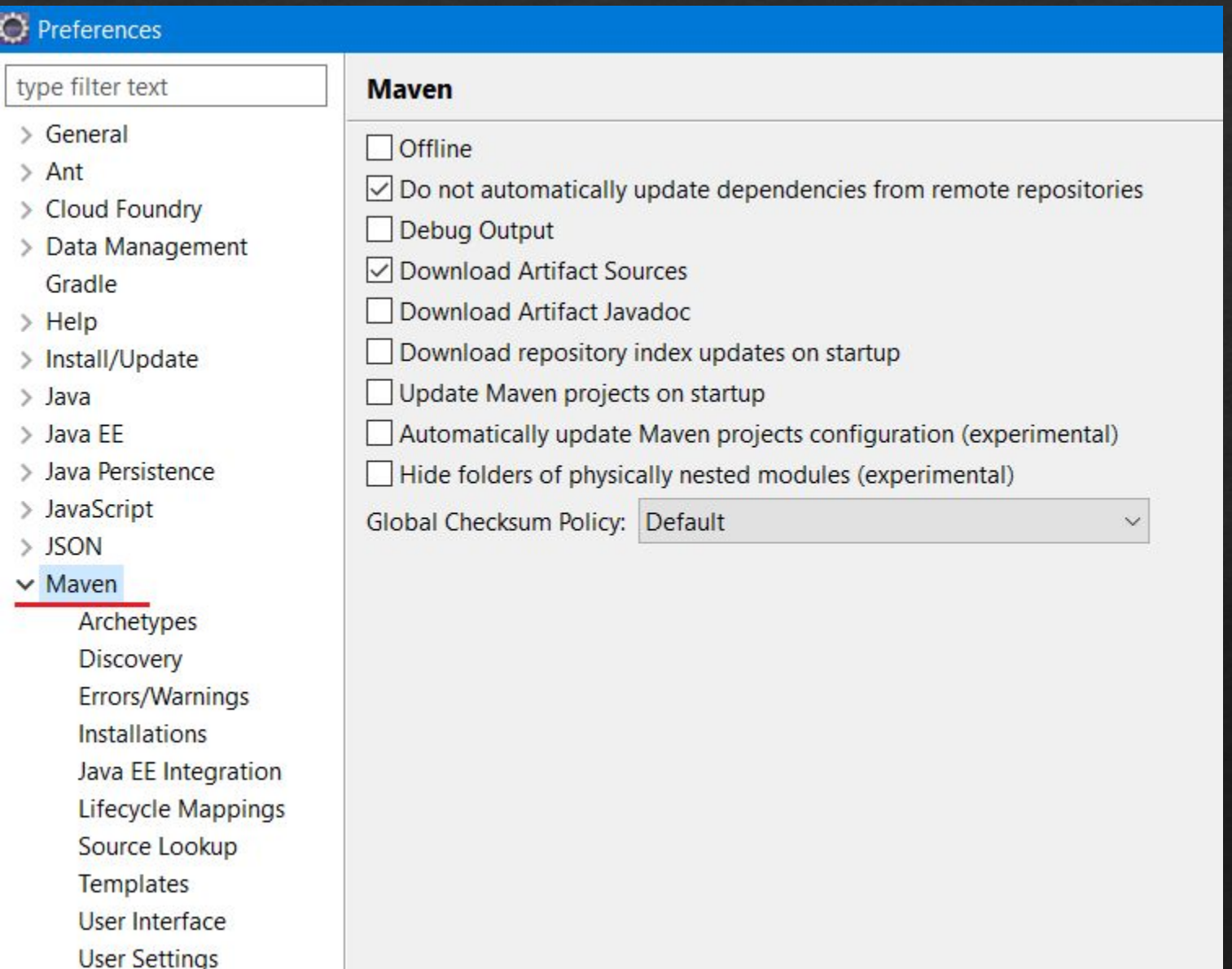
То скорее всего нужно добавить Location и Name, чтобы скачать пакет



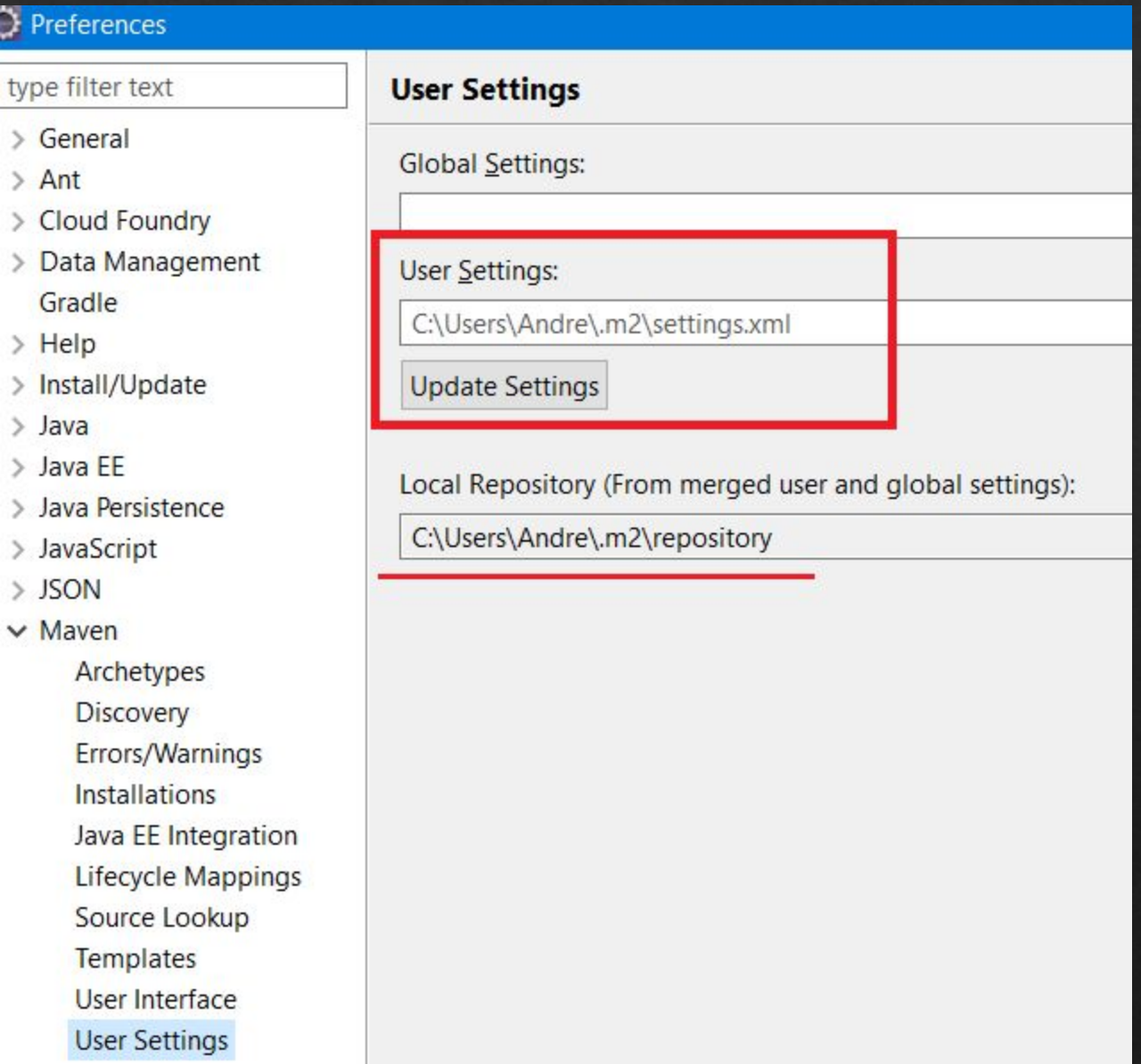
Так же проверить Network Connection
Возможно поменять Active Provider



Так же проверить Maven директорию



В User Settings пишут настройки прокси (если таковые имеются) в формате XML



Проверить Templates