

# CÁC BƯỚC TẠO 1 PROJECT WEB APPLICATION SỬ DỤNG INTELIJ VÀ TOMCAT SERVER

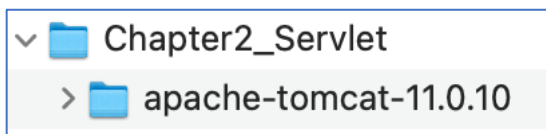
## 1. Tạo thư mục chứa Project

- Ví dụ: **Chapter2\_Servlet**

## 2. Download Tomcat server

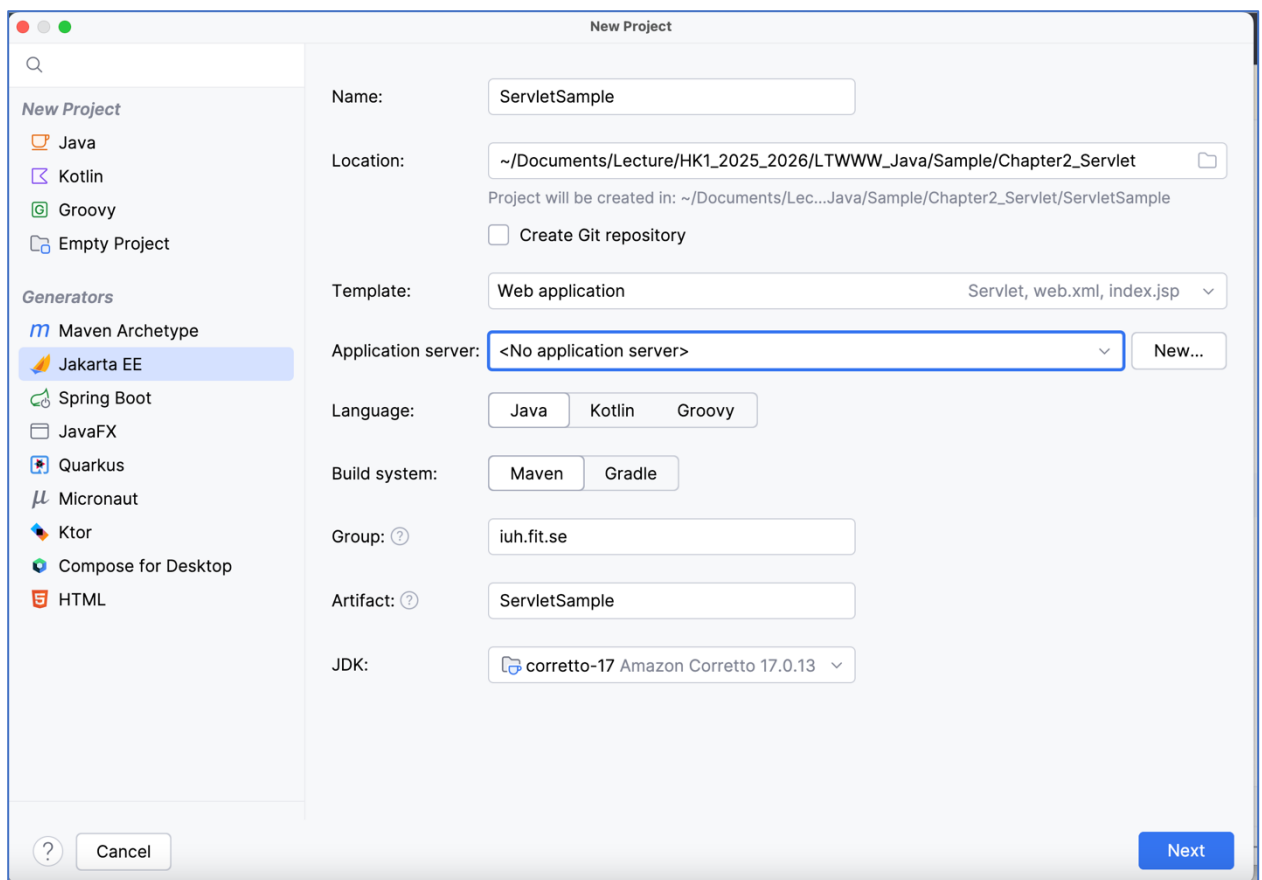
- Truy cập link: <https://tomcat.apache.org/download-11.cgi>
- Download: zip file

## 3. Giải nén và Copy Tomcat server vào thư mục đã tạo ở Mục 1

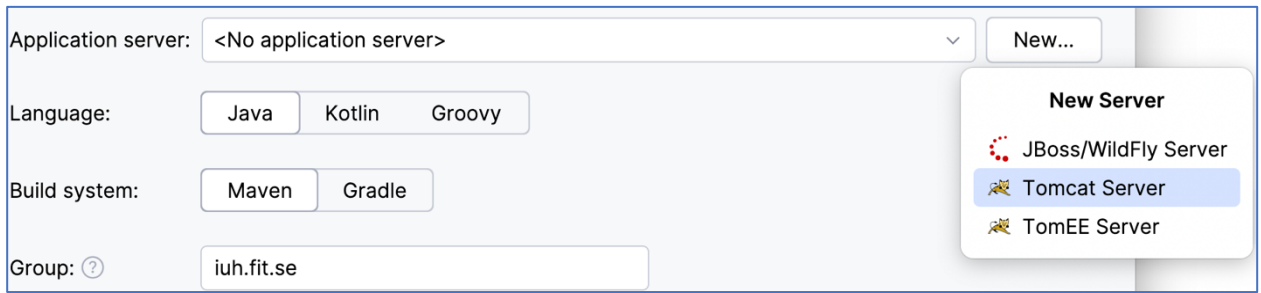


## 4. Mở IntelliJ và tạo project

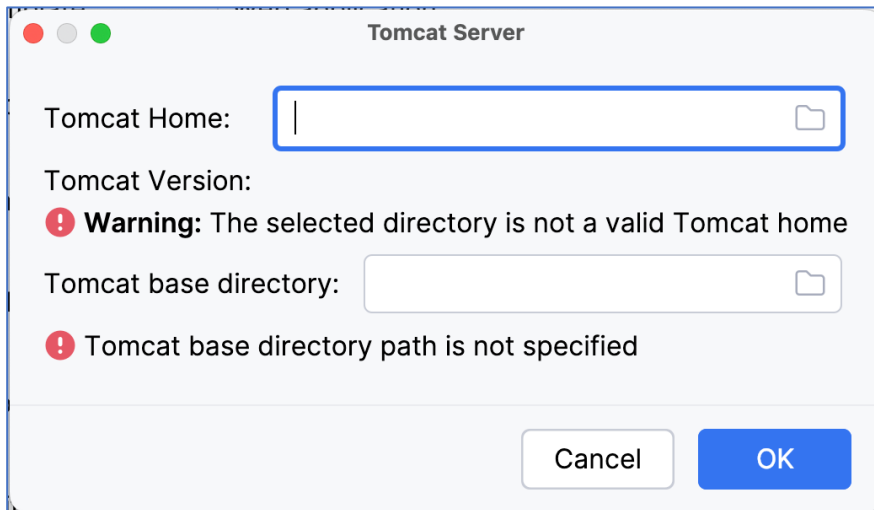
- IntelliJ → File → New Project



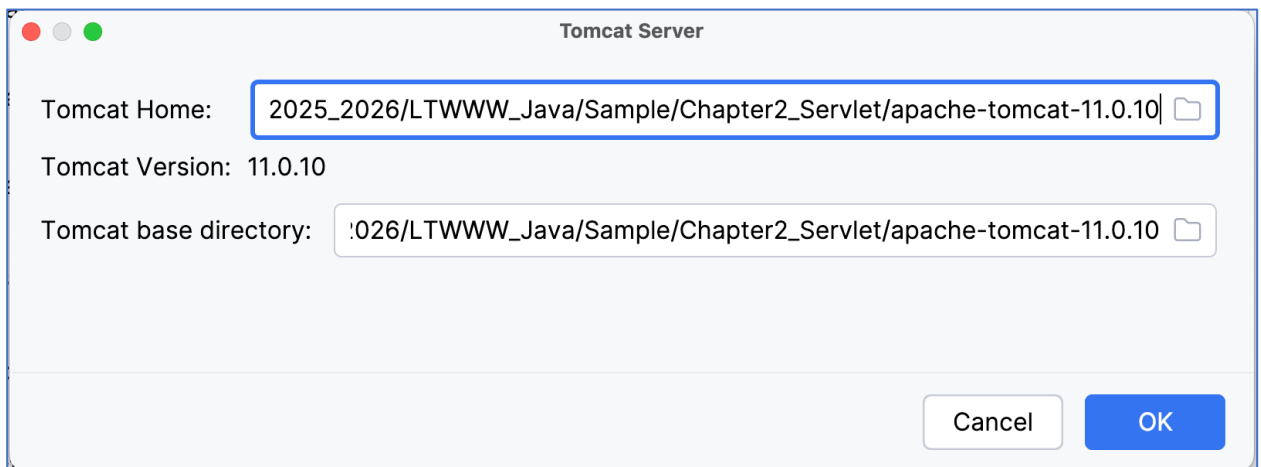
b. Tại mục **Application server** → Click “New” → Chọn **Tomcat Server**

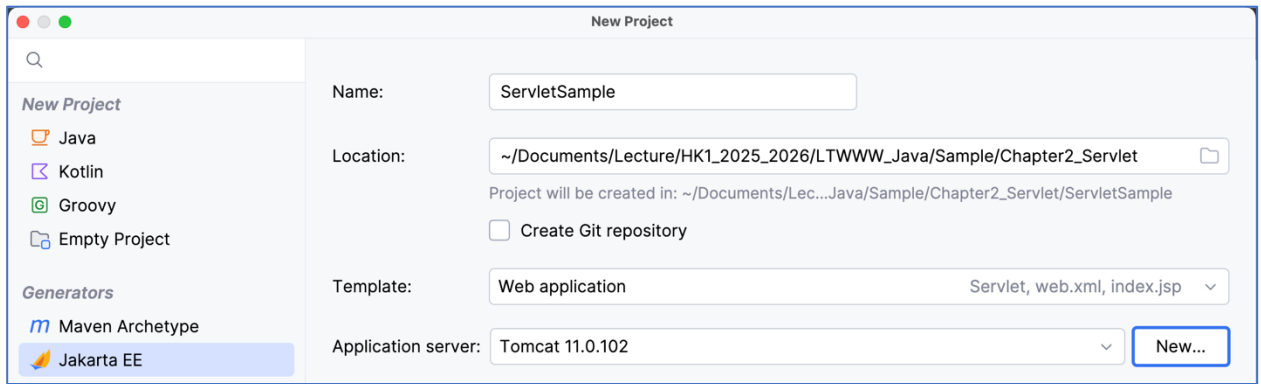


c. Click chọn **Tomcat Server** hiển thị hộp thoại như bên dưới:

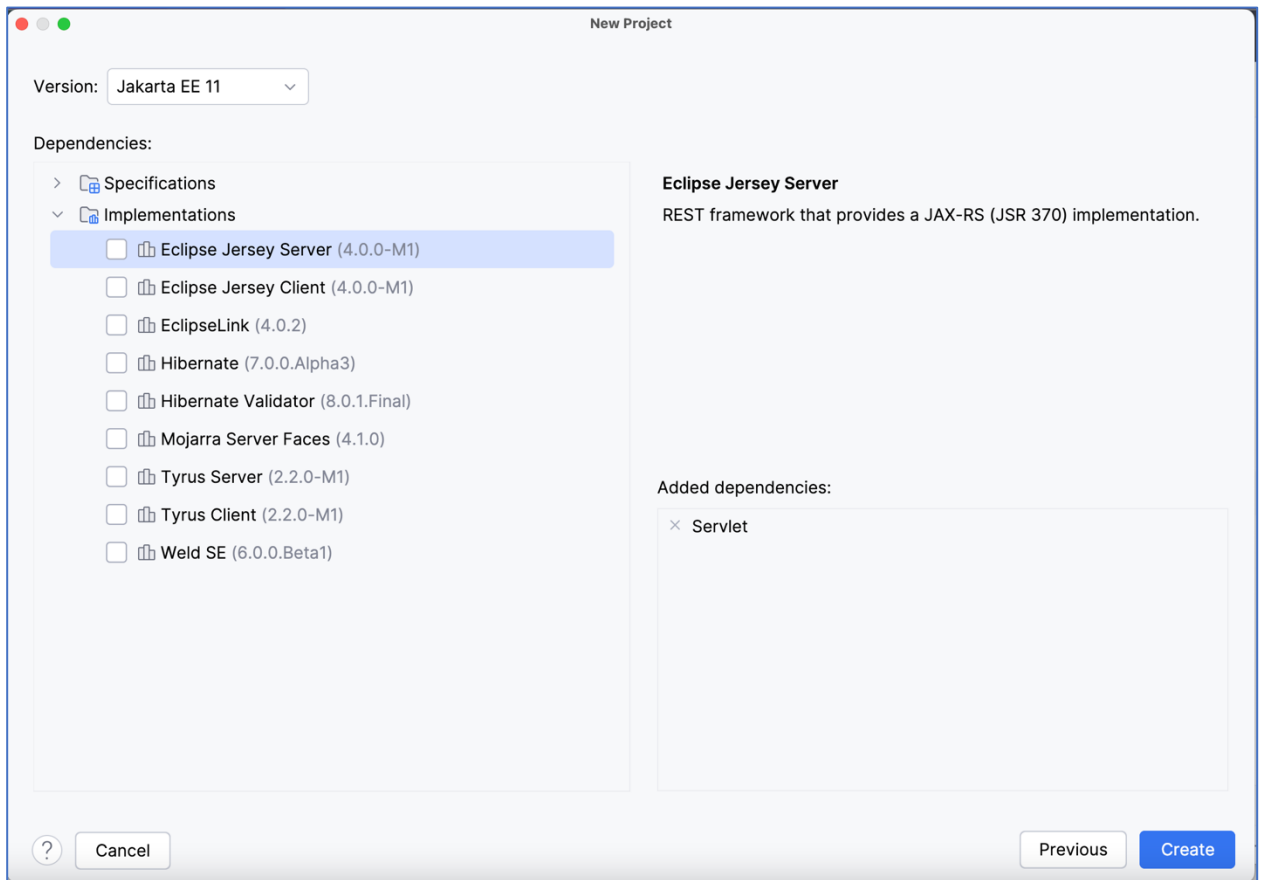


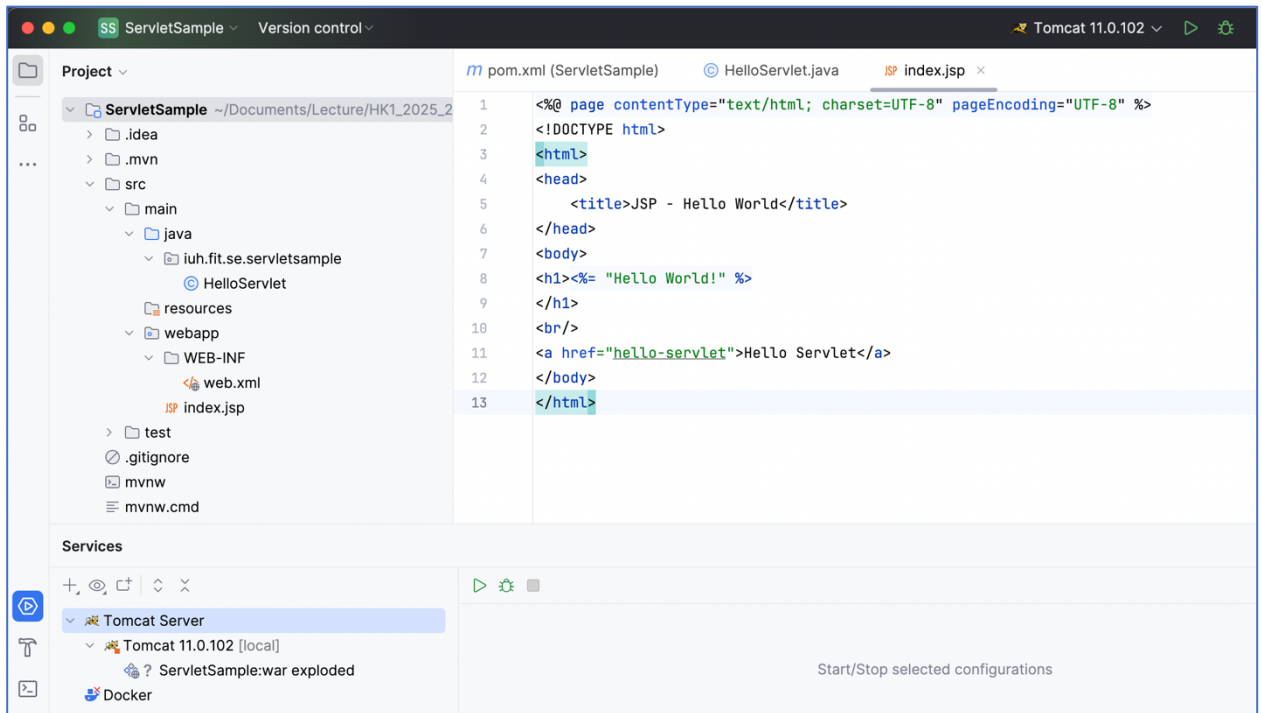
d. Tại mục **Tomcat Home** → Chọn đường dẫn đến **Folder chứa server Tomcat tại Mục 3** → Click chọn **OK**



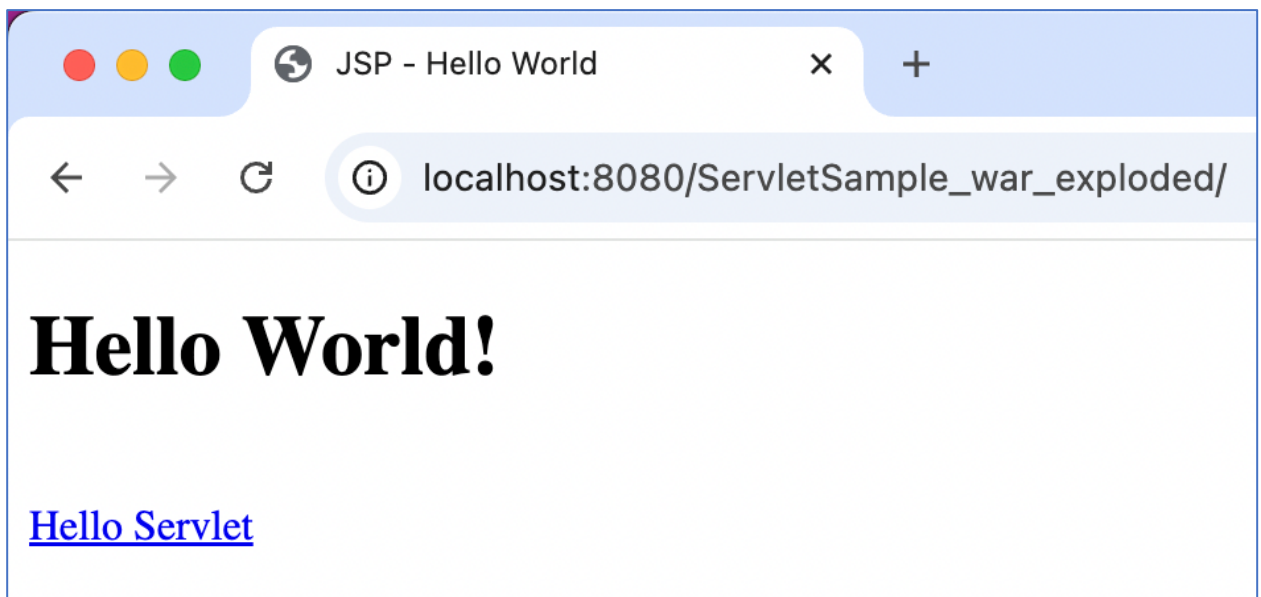


e. Click **Next** → **Chọn Jakarta EE 11** → **Create**

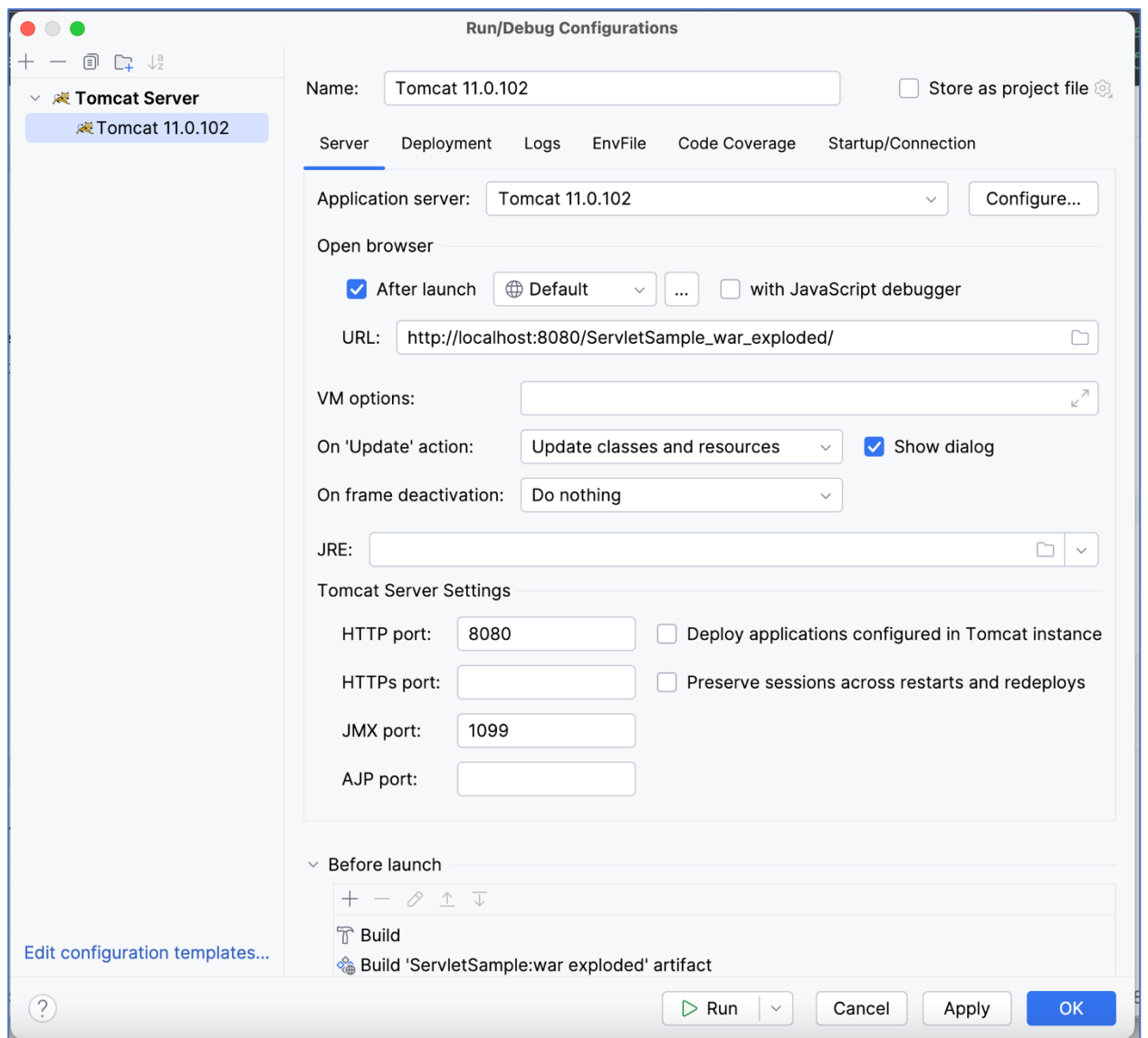
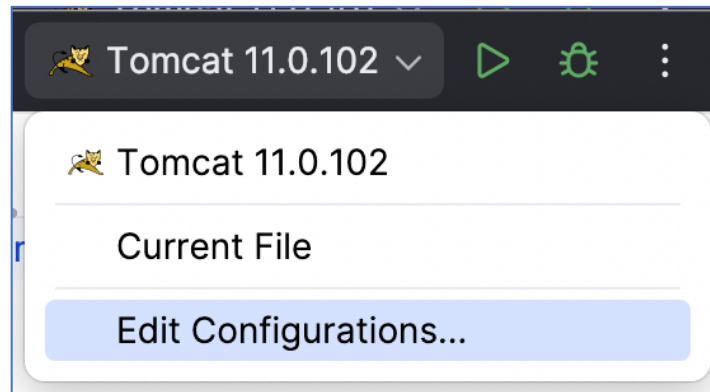


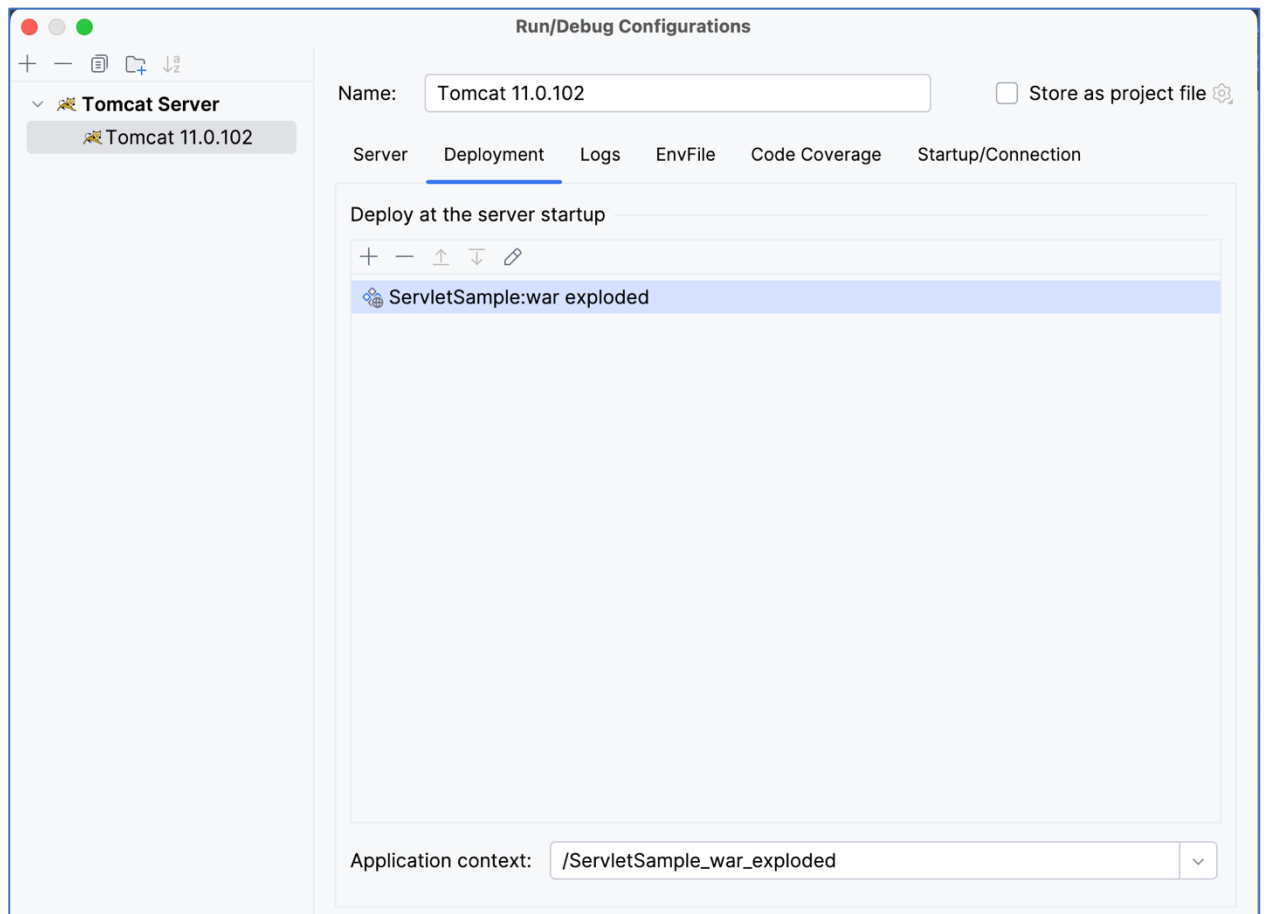


*f. Run web → Click vào **Icon Play** bên trái **Tomcat Server** → Web sẽ hiển thị ở browser → Hiển thị nội dung page **index.jsp***



g. Tomcat → Edit Configurations ....



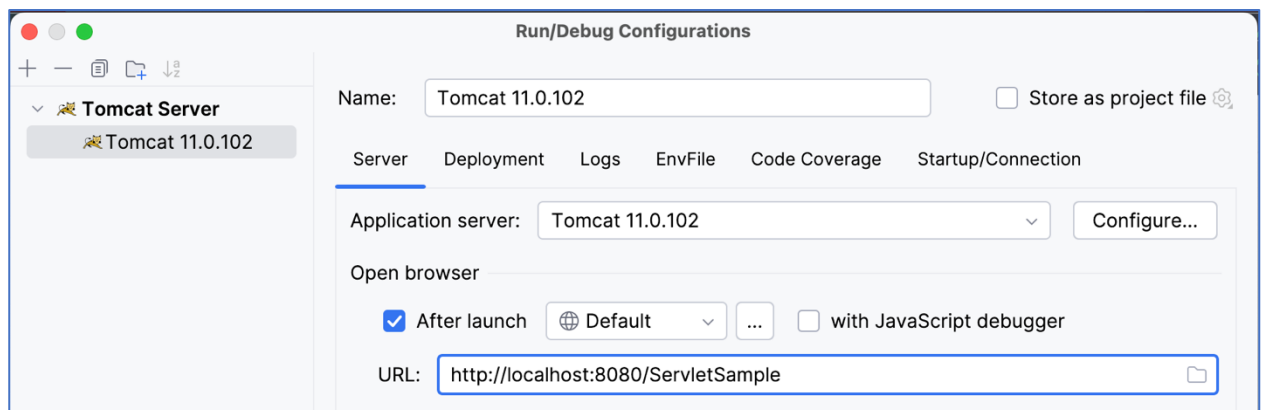


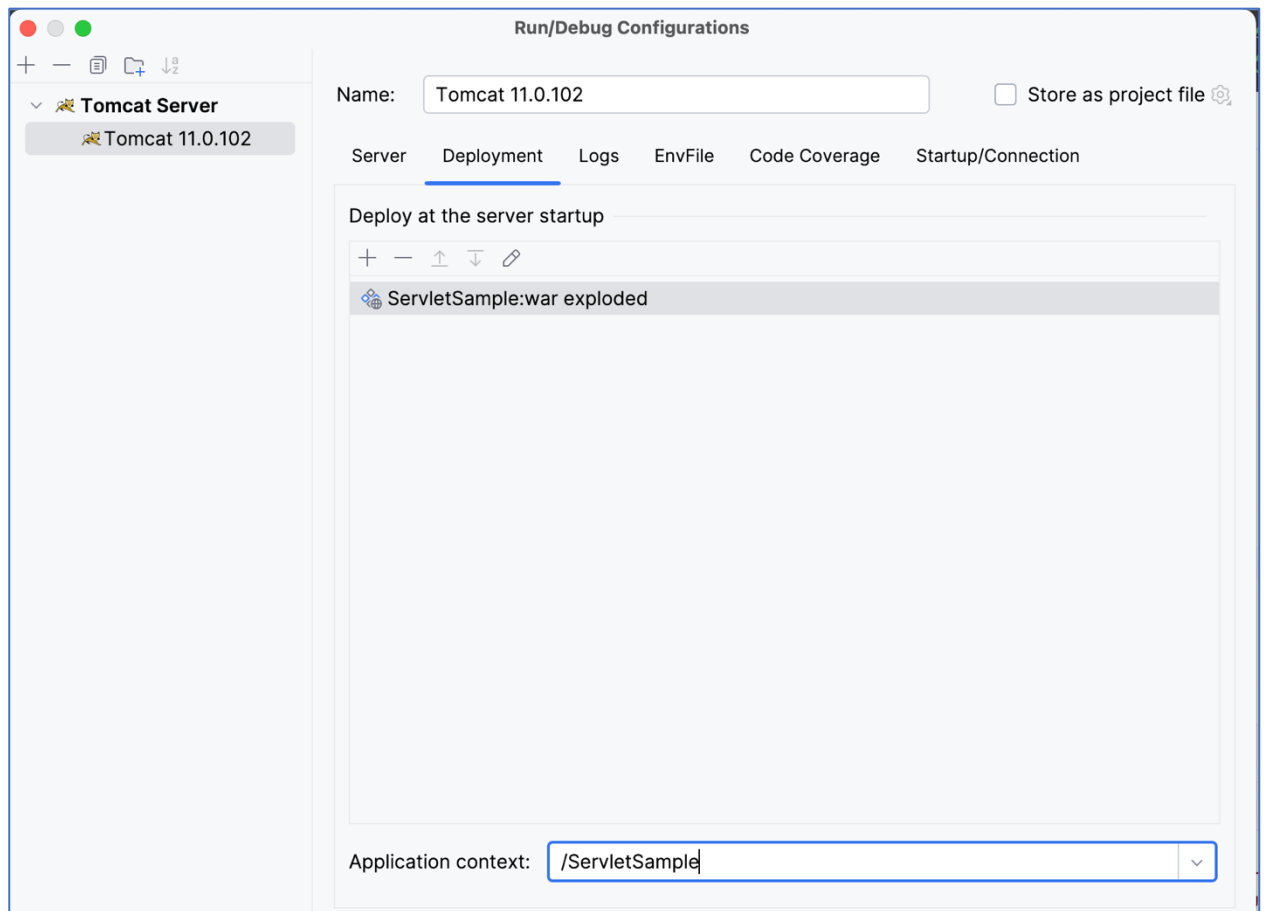
h. Để chỉnh URL:

[http://localhost:8080/ServletSample\\_war\\_exploded](http://localhost:8080/ServletSample_war_exploded) →

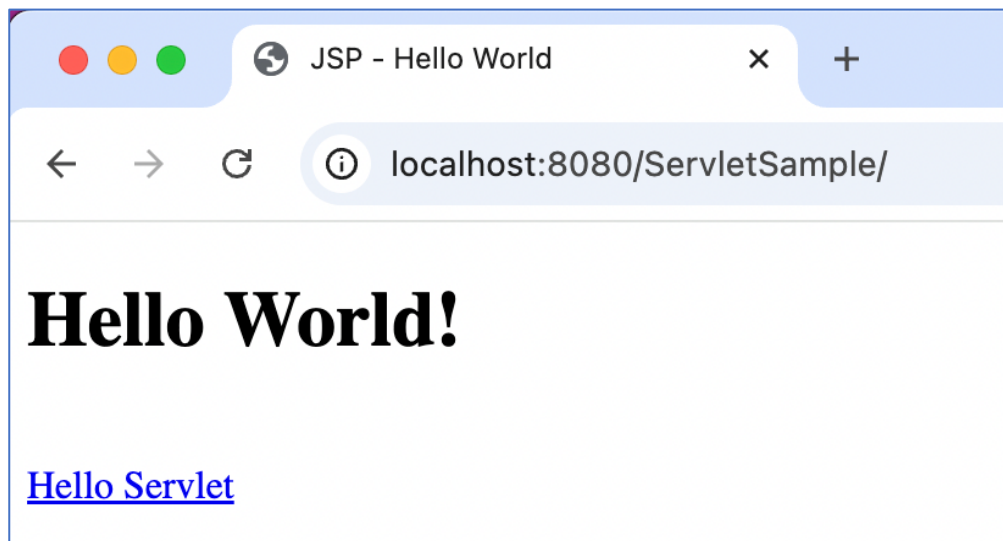
<http://localhost:8080/ServletSample>

→ chỉnh sửa ở 2 vị trí:

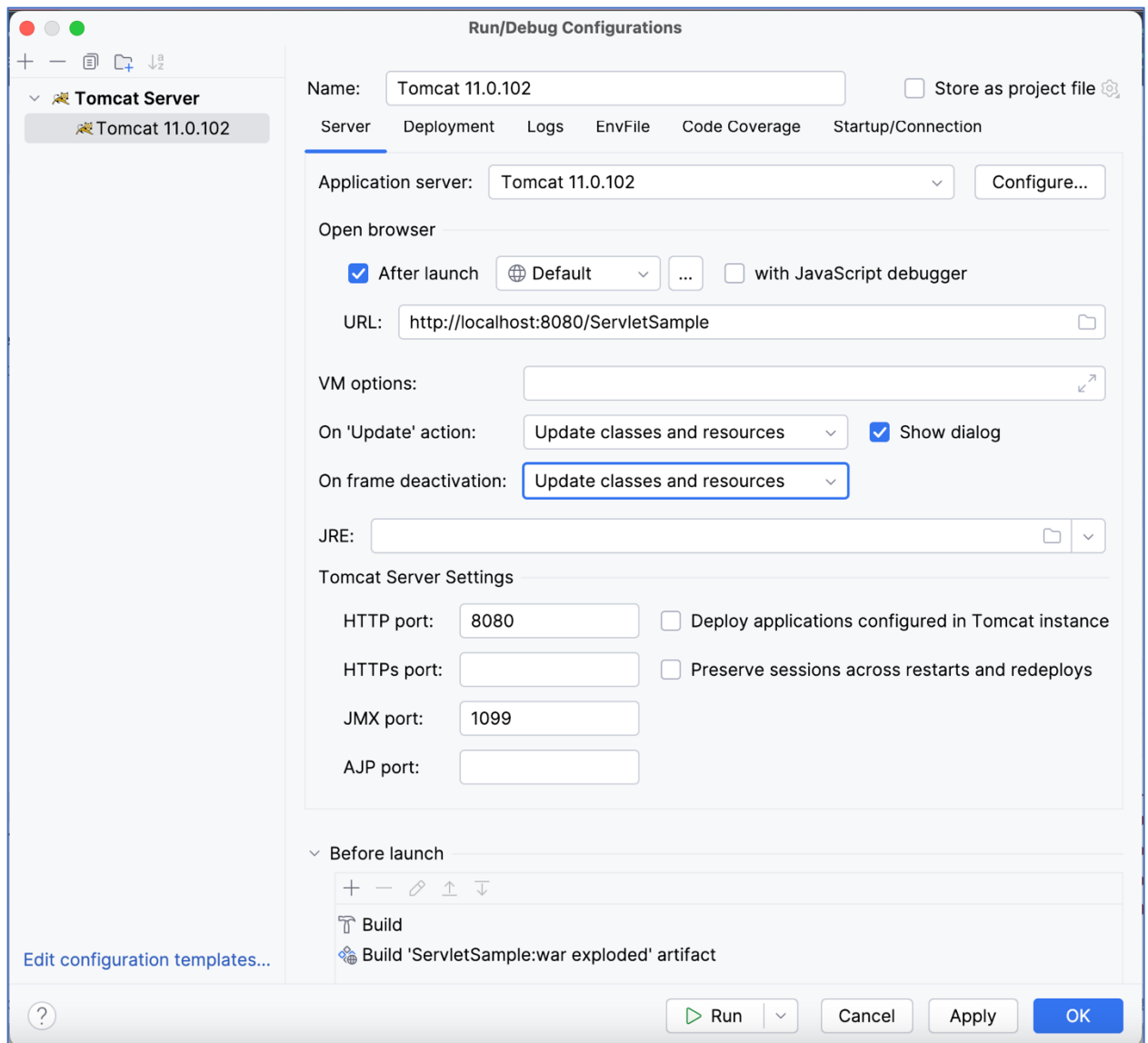




→ Click OK → Run web



5. Đối với những file .htm, .css, .js thì không cần restart lại Tomcat Server. Khi chỉnh sửa những file này thì chỉ cần Refresh lại Browser thì nội dung chỉnh sửa sẽ được hiển thị ngay tức thì. Ở Config Tomcat server cần config như bên dưới:



- Do nothing: không làm gì
- Update classes and resources: tự động deploy lại các thay đổi trong mã nguồn (hot reload)
- Update resources: chỉ cập nhật lại các thay đổi html, css, .... không cần redeploy lại class Java



6. Triển khai Servlet. Có 2 cách triển khai:

a. **Cách 1**: thực hiện trong file **WEB-INF/web.xml**

```
<servlet>
  <servlet-name>UserServlet(Tên Servlet)</servlet-name>
  <servlet-class>iuh.fit.se.servletsample.UserServlet(path đến class
của Servlet)</servlet-class>
  <init-param>
    <param-name>name(Tên parameter)</param-name>
    <param-value>Not provided(Value của parameter)</param-value>
  </init-param>
  <init-param>
    <param-name>email</param-name>
    <param-value>Not provided</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>UserServlet(Tên Servlet)</servlet-name>
  <url-pattern>/userServlet(url - route)</url-pattern >
</servlet-mapping>
```

b. **Cách 2**: sử dụng Annotation

```
@WebServlet(
  name = "UserServlet", urlPatterns = "/userServlet",
  initParams={ @WebInitParam(name="name", value="Not provided"),
               @WebInitParam(name="email", value="Not provided") }
)
public class UserServlet extends HttpServlet {
  @Override
  protected void doGet( HttpServletRequest request,
  HttpServletResponse response) throws ServletException, IOException {
    // TODO
  } }
```

Name	Type	Required	Description
<b>value</b> or <b>urlPatterns</b>	<i>String[]</i>	Required	Specify one or more URL patterns of the servlet. Either of attribute can be used, but not both.
<b>name</b>	<i>String</i>	Optional	Name of the servlet
<b>displayName</b>	<i>String</i>	Optional	Display name of the servlet
<b>description</b>	<i>String</i>	Optional	Description of the servlet
<b>asyncSupported</b>	<i>boolean</i>	Optional	Specify whether the servlet supports asynchronous operation mode. Default is false.
<b>initParams</b>	<i>WebInitParam[]</i>	Optional	Specify one or more initialization parameters of the servlet. Each parameter is specified by <a href="#">@WebInitParam</a> annotation type.
<b>loadOnStartup</b>	<i>int</i>	Optional	Specify load-on-startup order of the servlet. Default: -1 if the integer is negative: The container loads servlet at any time. if the integer is 0 or positive: The servlet marked with lower integers are loaded before servlets marked with higher integers.
<b>smallIcon</b>	<i>String</i>	Optional	Specify name of the small icon of the servlet.
<b>largeIcon</b>	<i>String</i>	Optional	Specify name of the large icon of the servlet.

7. Khởi tạo parameter (Init Paramaters):

- a. Sử dụng trong Servlet: sử dụng **init-param** hoặc sử dụng annotation **initParams** và khai báo bên trong Servlet

```
<servlet>
  <servlet-name>UserServlet(Tên Servlet)</servlet-name>
  <servlet-name>UserServlet(Tên class của Servlet)</servlet-name>
  <init-param>
    <param-name>name(Tên parameter)</param-name>
    <param-value>Not provided(Value của parameter)</param-value>
  </init-param>
  <init-param>
    <param-name>email</param-name>
    <param-value>Not provided</param-value>
  </init-param>
</servlet>
```

```
@WebServlet(
    name = "UserServlet",
    urlPatterns = "/userServlet",
    initParams={
        @WebInitParam(name="name", value="Not provided"),
        @WebInitParam(name="email", value="Not provided")
    }
)
```

- b. Sử dụng trong **toàn cục (global)**: khi cần sử dụng các bất biến (immutable data) và truy xuất toàn cục thì chúng ta nên sử dụng application context để khởi tạo parameters để lưu trữ dữ liệu (web.xml).

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="https://jakarta.ee/xml/ns/jakartaee"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd"
id="WebApp_ID" version="6.0">
  <context-param>
    <param-name>province</param-name>
    <param-value>Mendoza</param-value>
  </context-param>

  <context-param>
    <param-name>country</param-name>
    <param-value>Argentina</param-value>
  </context-param>
</web-app>
```

Cách sử dụng trong Servlet:

```
@WebServlet(
    name = "UserServlet",
    urlPatterns = "/userServlet",
    initParams={
        @WebInitParam(name="province", value="Not provided"),
        @WebInitParam(name="country", value="Not provided")
    }
)
public class UserServlet extends HttpServlet {
    @Override
    protected void doGet( HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        ServletContext context = getServletContext();

        String province = context.getInitParameter("province");
        String country = context.getInitParameter("country");
    }
}
```

```
response.setContentType("text/plain;charset=UTF-8");  
response.getWriter().println("Province: " + province);  
response.getWriter().println("Country: " + country);  
}  
}
```