

Eclipse を使って「動的 Web プロジェクト」としての

Tomcat プログラムの作り方

OD 教室の Eclipse はどのパソコンを利用しても、同一の環境が再現できるように、ホームドライブに Eclipse の設定を保存している為、OD 教室では一般の PC 環境と違う仕組みとなります。

Eclipse の「動的 Web プロジェクト」としての Tomcat プログラムの作り方を説明します。

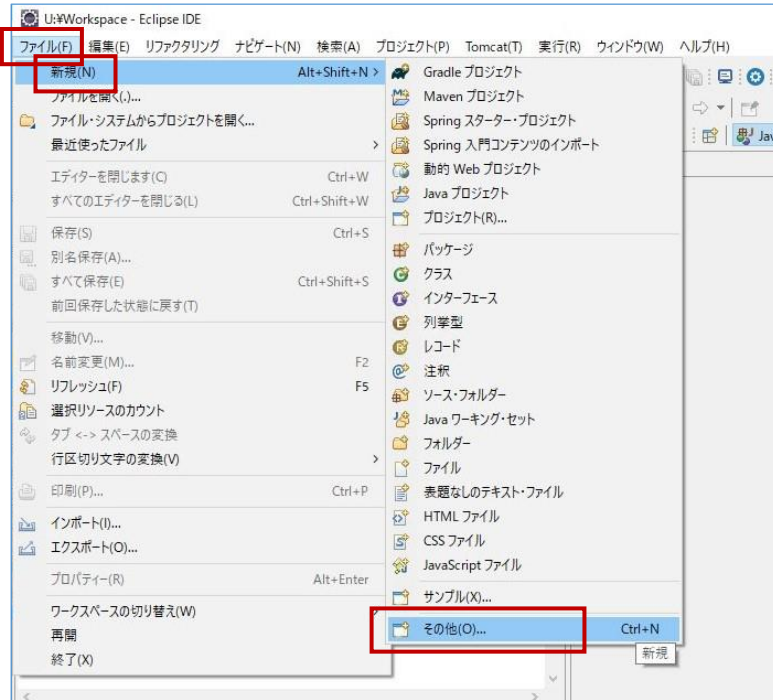
1 動的 Web プロジェクト

- 1.1 Tomcat サーバ用プロジェクトの作成
- 1.2 動的 Web プロジェクトの作成
- 1.3 動的 Web プロジェクトの設定
- 1.4 プログラムの作成
- 1.5 プログラムの実行

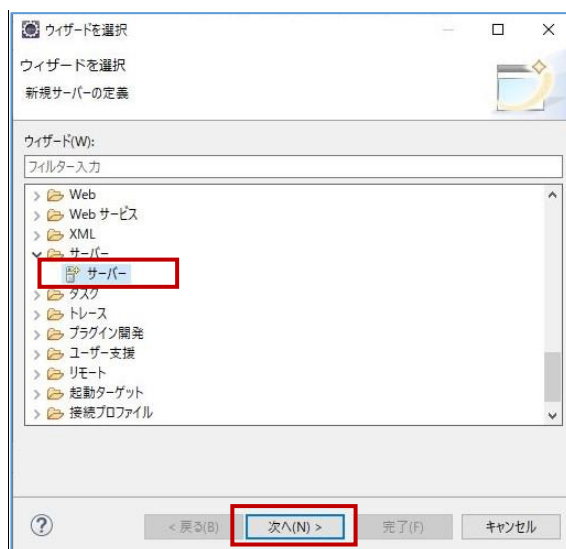
1 動的 Web プロジェクト

1.1 Tomcat サーバ用プロジェクトの作成

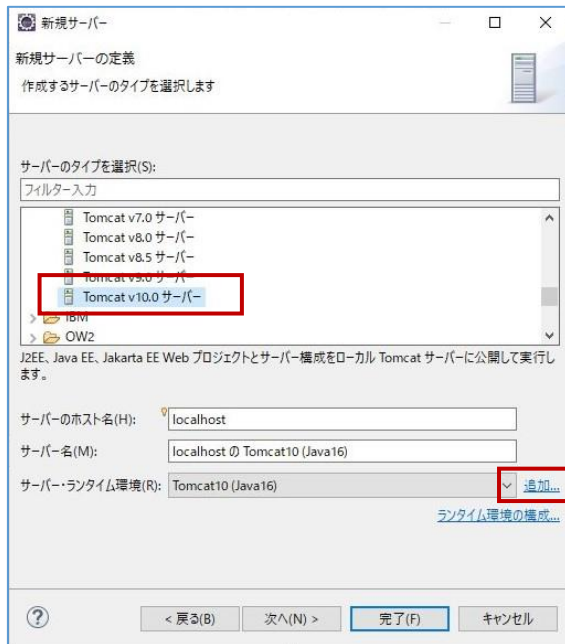
① Eclipseを起動し、メニューバーの[ファイル]⇒[新規]⇒[その他]とクリックします。



② [サーバー]⇒[サーバー]と選択し、[次へ]をクリックします。



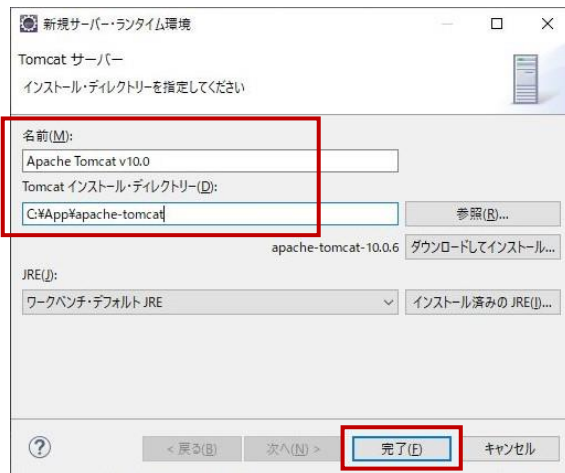
- ③ [Tomcat v10.0サーバー]を選択し、[追加]をクリックします。



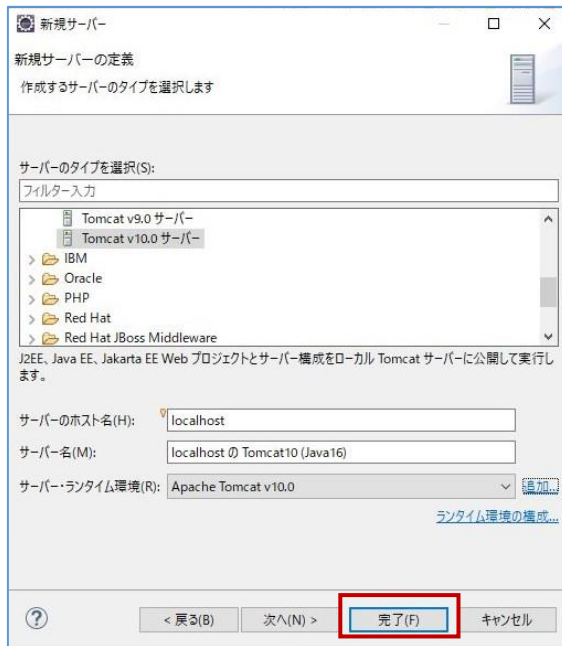
- ④ [名前]と[Tomcatインストールディレクトリ]を、以下の値を設定し、[完了]をクリックします。

名前 : 任意のランタイム名 (後手順で使用します)

Tomcatインストール・ディレクトリ : C:¥App¥apache-tomcat

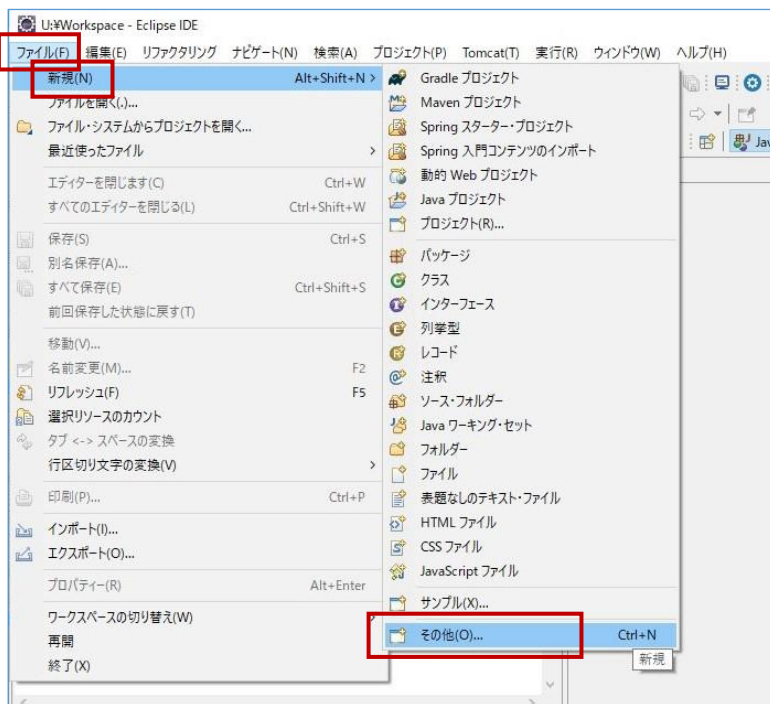


- ⑤ [完了]をクリックします。

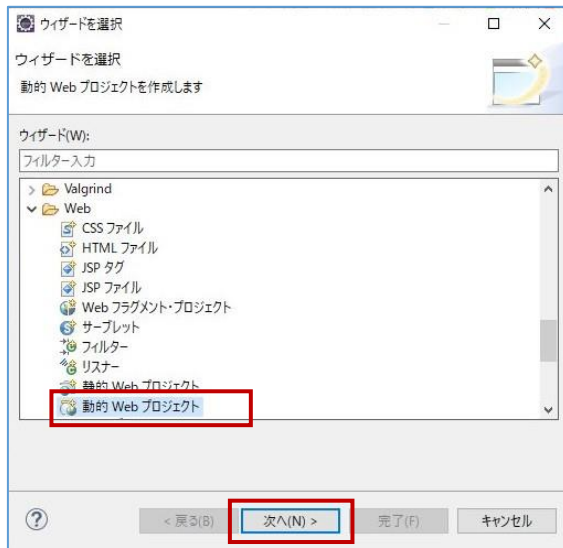


1.2 動的 We プロジェクトの作成

- ① Eclipseのメニューバーの[ファイル]⇒[新規]⇒[その他]とクリックします。



- ② [web]⇒[動的Webプロジェクト]を選択し、[次へ]をクリックします。

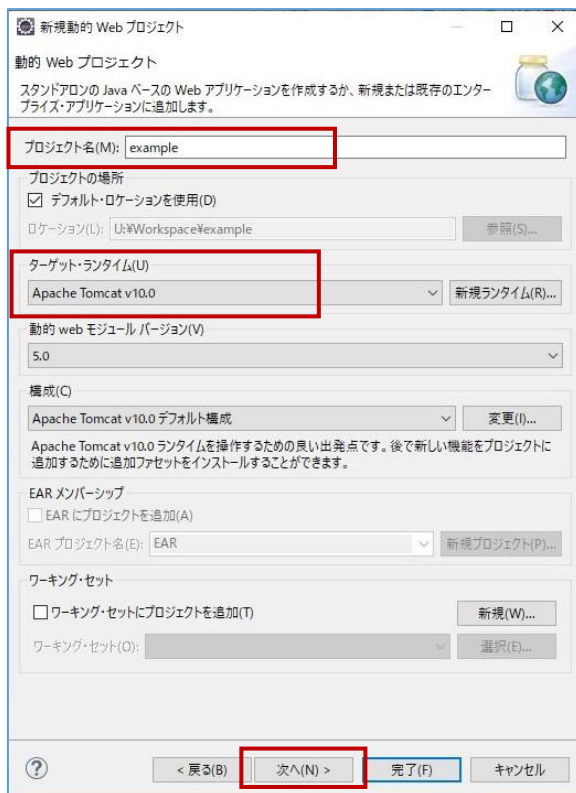


- ③ [プロジェクト名]に、任意のプロジェクト名を設定します。

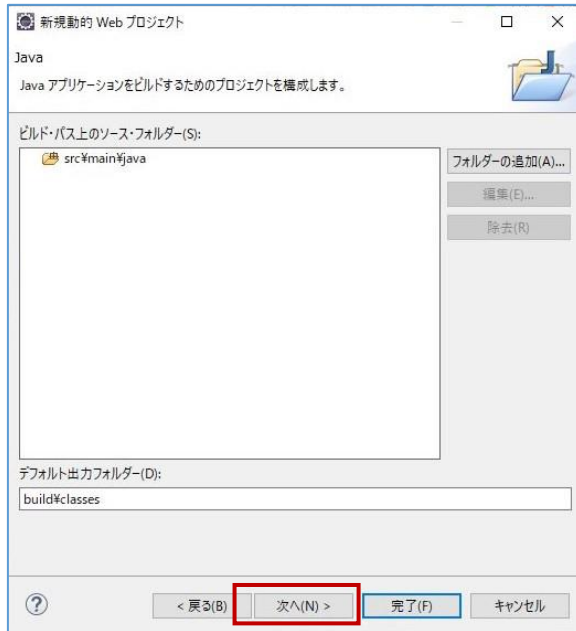
また、[ターゲット・ランタイム]は、前手順で作成したTomcatの設定のランタイム名を指定します。

※特に変更せず進めている場合は、[Apache Tomcat v10.0]となるはずですが。

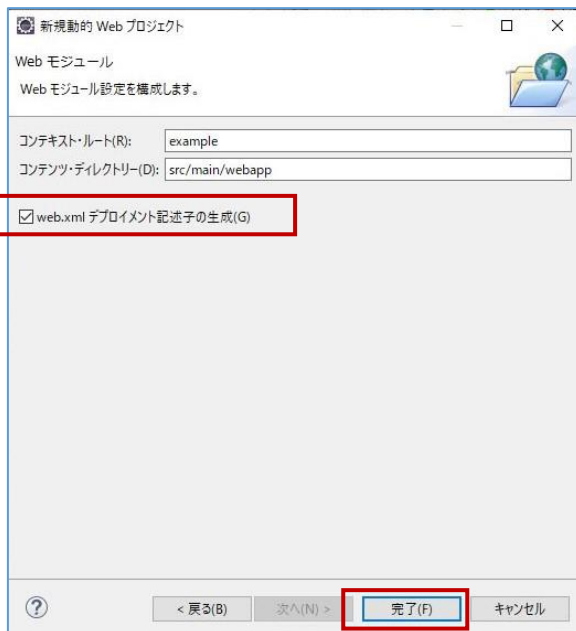
その後、[次へ]をクリックします。



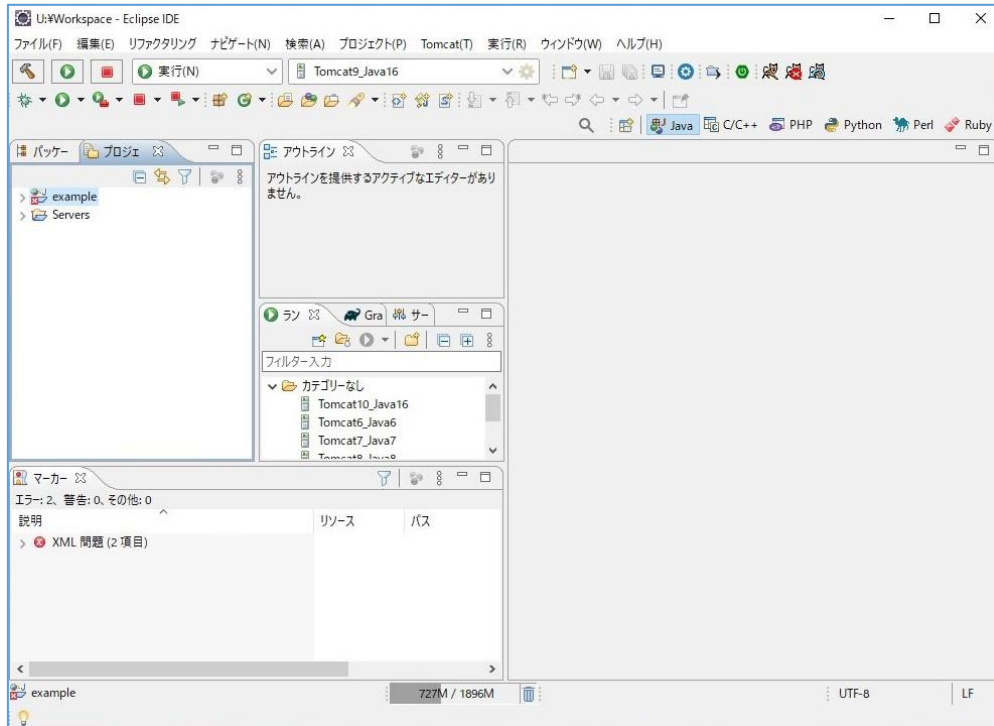
- ④ [次へ]をクリックします。



- ⑤ [web.xmlデプロイメント記述子の生成]のチェックがONになっていることを確認し、[完了]をクリックします。

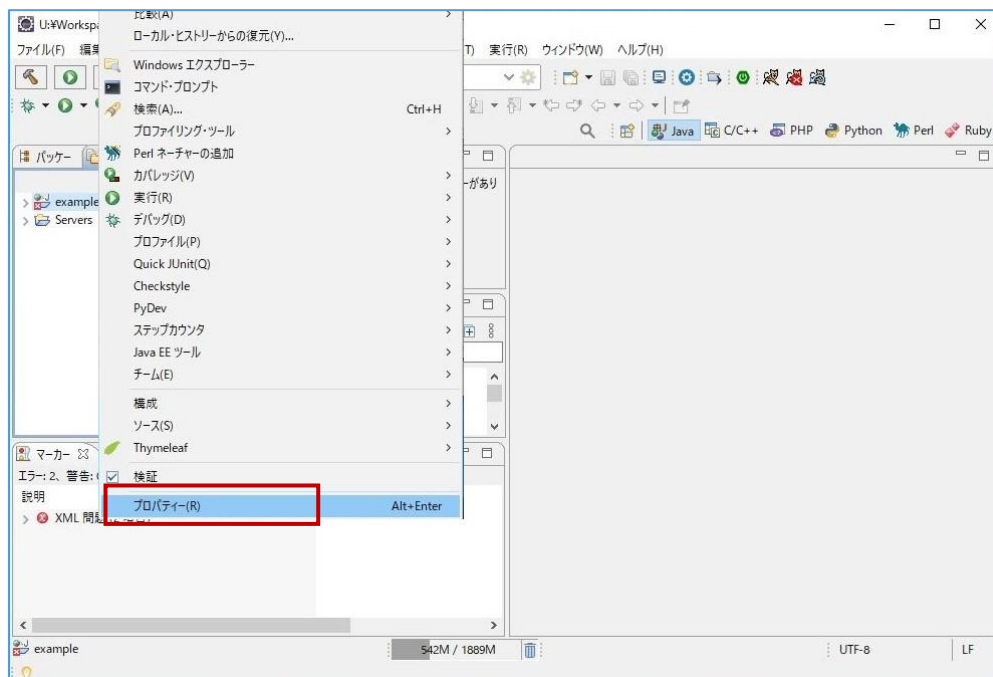


⑥ プロジェクトが作成されます。

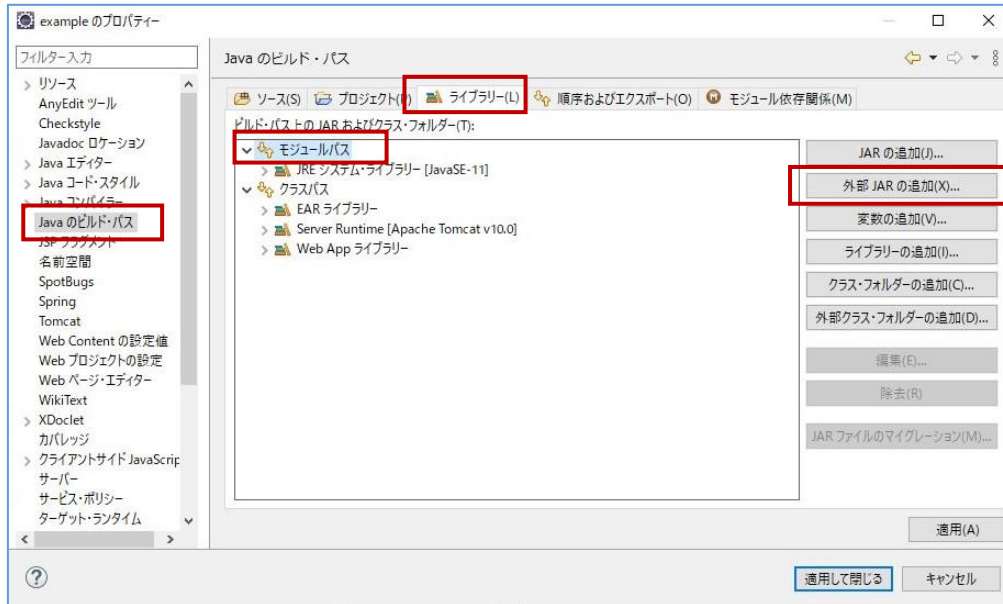


1.3 動的 We プロジェクトの設定

① 作成されたプロジェクトを右クリックし、[プロパティ]を選択します。

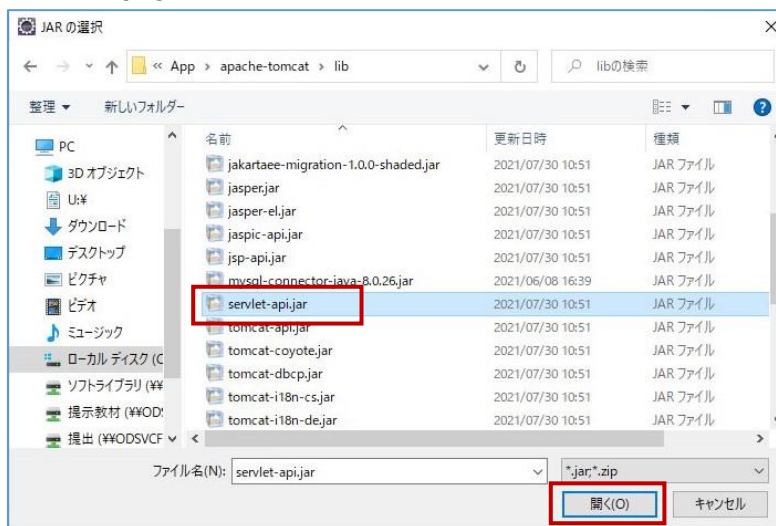


- ② 左側のエリアで[Javaのビルド・パス]を選択し、[ライブラリ]タブを開きます。
その後、[モジュールパス]を選択した状態で、[外部JARの追加]ボタンをクリックします。

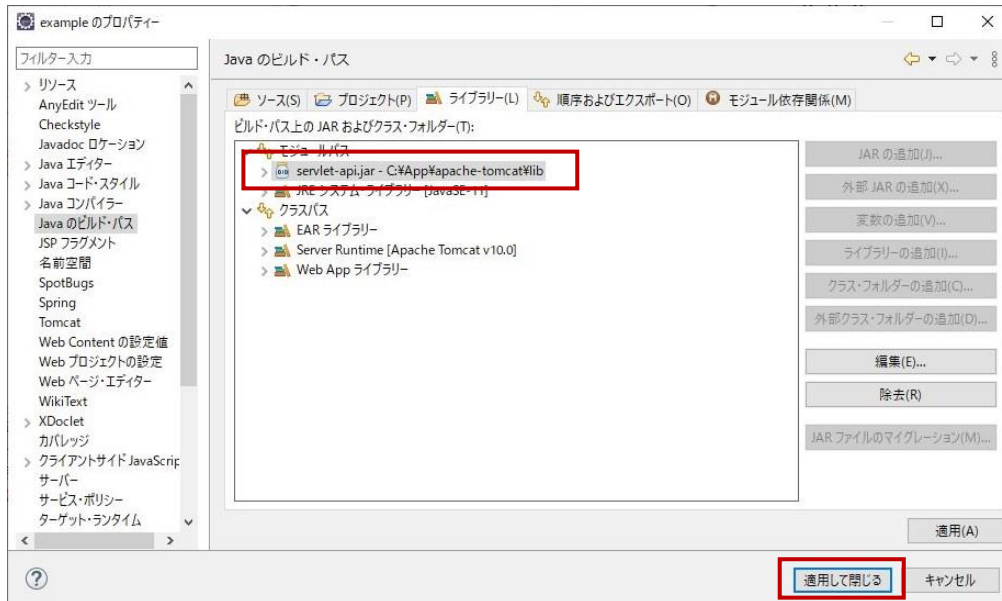


- ③ ファイル選択画面で、以下のフォルダーの中の、
C:¥App¥apache-tomcat¥lib
以下のファイルを選択し、[開く]をクリックします。

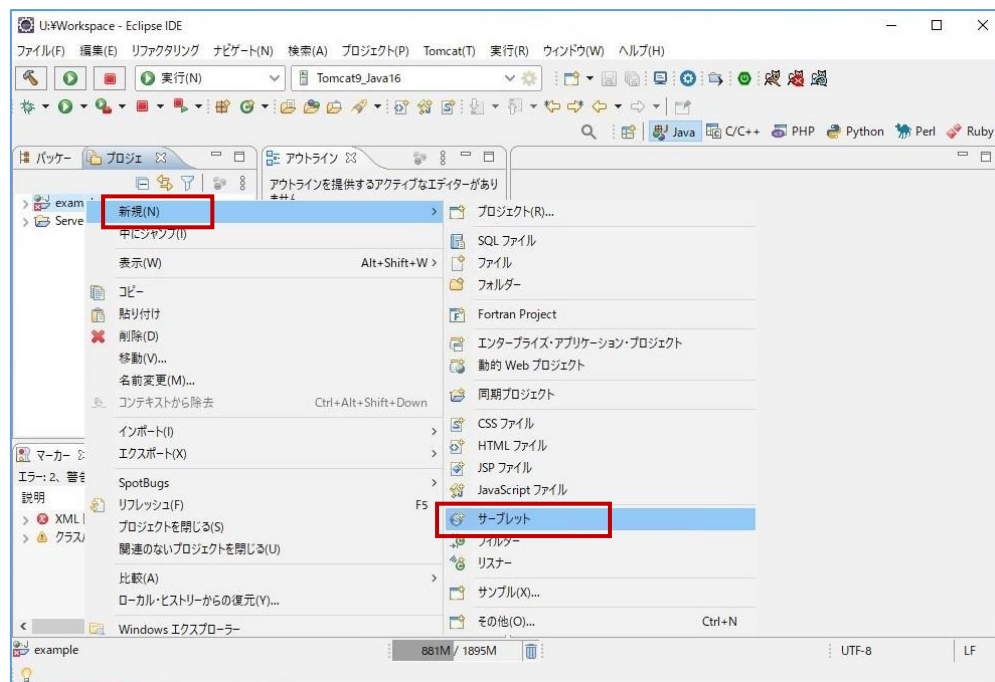
servlet-api.jar



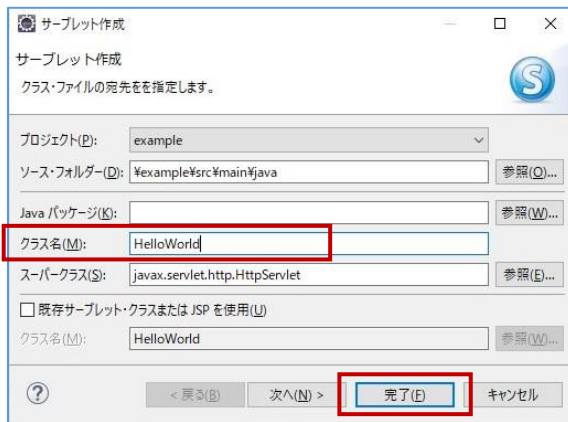
- ④ 前手順で選択したファイルが追加されていることを確認し、[適用して閉じる]をクリックします。



- ⑤ 次に、作成したプロジェクトを右クリックし、[新規]⇒[サーブレット]とクリックします。



- ⑥ [クラス名]に、任意のクラス名を入力し、[完了]をクリックします。



The screenshot shows the 'Servlet Creation' dialog box. The 'Project' is set to 'example'. The 'Source Folder' is '%example%src%main%java'. The 'Java Package' is empty. The 'Class Name' field is highlighted with a red box and contains 'HelloWorld'. The 'Superclass' is 'javax.servlet.http.HttpServlet'. The 'Use Existing Servlet Class or JSP' checkbox is unchecked. The 'Class Name' field at the bottom is also 'HelloWorld'. The 'Finish' button is highlighted with a red box.

1.4 プログラムの作成

- ① 作成されたHelloWorld.javaファイルへ、コードを記述します。

※以下はサンプルです。実際に作成する場合は、各自でコードを記述してください。

HelloWorld.java ※サンプル

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import jakarta.servlet.ServletException;
4 import jakarta.servlet.http.HttpServlet;
5 import jakarta.servlet.http.HttpServletRequest;
6 import jakarta.servlet.http.HttpServletResponse;
7 public class HelloWorld extends HttpServlet {
8     public HelloWorld() {}
10     protected void doGet(HttpServletRequest request,
11         HttpServletResponse response)
12         throws ServletException, IOException {
13         PrintWriter out = response.getWriter();
14         out.println(
15             "<html><body>Hello World!</body></html>");
16     }
17 }
```

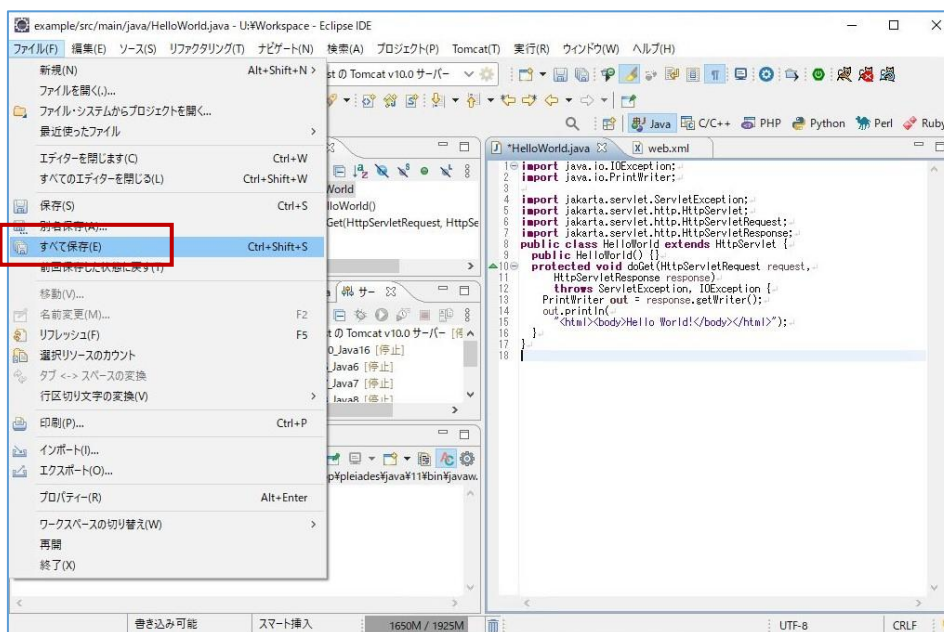
- ② 必要に応じて、web.xmlファイルも更新します。

※以下はサンプルです。実際に作成する場合は、各自でコードを記述してください。

HelloWorld.java ※サンプル

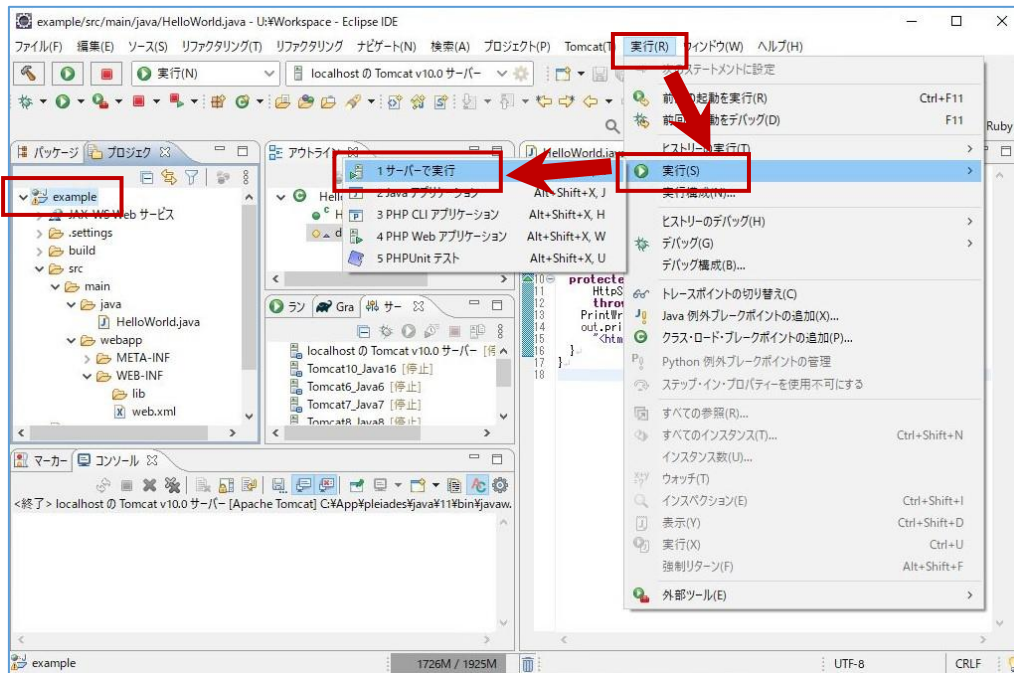
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
5 http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID"
6 version="4.0">
7   <display-name>Example02</display-name>
8   <welcome-file-list>
9     <welcome-file>index.html</welcome-file>
10    <welcome-file>index.htm</welcome-file>
11    <welcome-file>index.jsp</welcome-file>
12    <welcome-file>default.html</welcome-file>
13    <welcome-file>default.htm</welcome-file>
14    <welcome-file>default.jsp</welcome-file>
15  </welcome-file-list>
16  <servlet>
17    <servlet-name>Hello</servlet-name>
18    <servlet-class>HelloWorld</servlet-class>
19  </servlet>
20  <servlet-mapping>
21    <servlet-name>Hello</servlet-name>
22    <url-pattern>/</url-pattern>
23  </servlet-mapping>
24 </web-app>
```

- ③ プログラムのコード完成後、メニューバーの[ファイル]⇒[すべて保存]で、保存します。

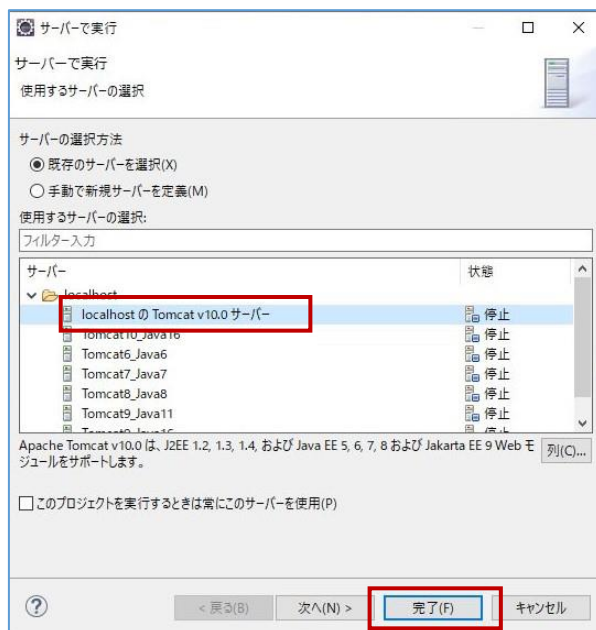


1.5 プログラムの実行

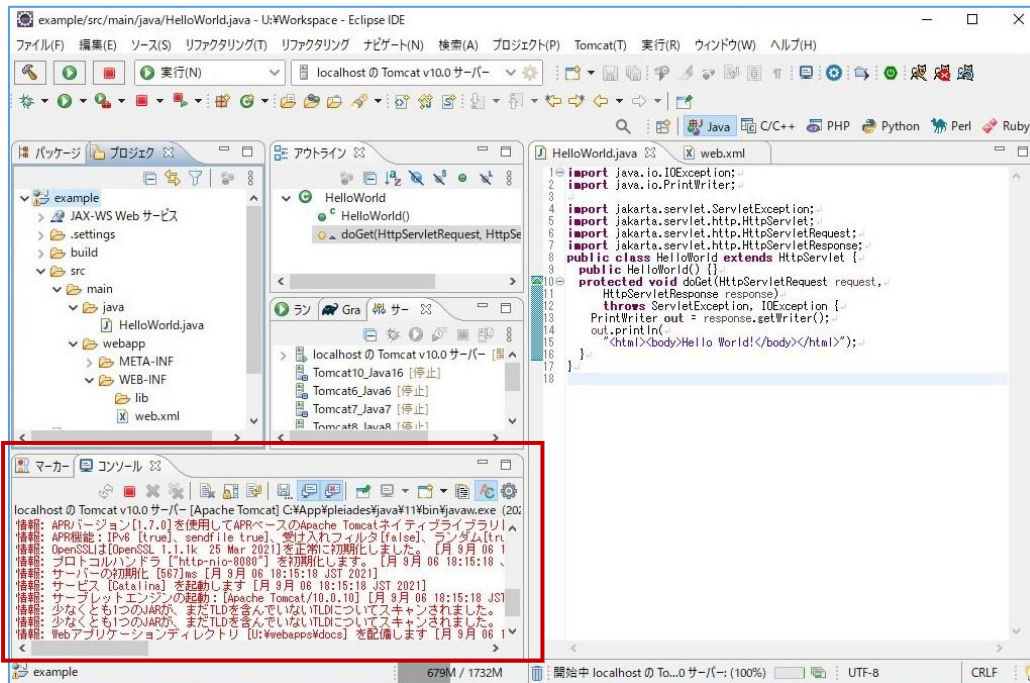
- ① 作成したプロジェクトを選択した状態で、メニューバーの[実行]⇒[実行]⇒[サーバーで実行]とクリックします。



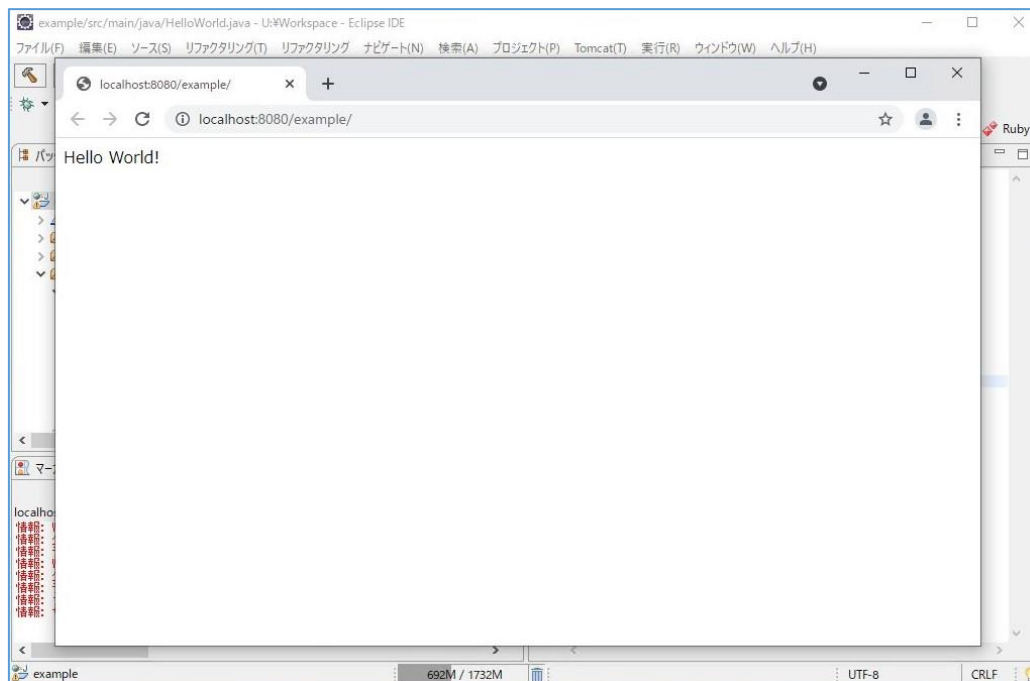
- ② [localhostのTomcat v10.0サーバー]を選択し、[完了]をクリックします。



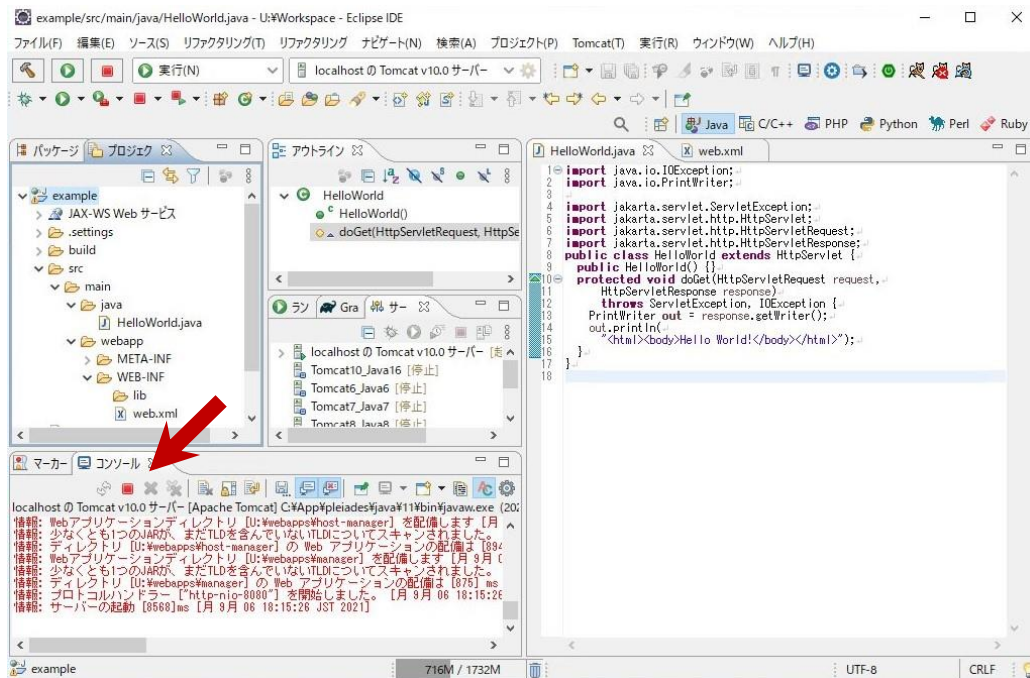
- ③ コンソール部を確認し、Tomcatサーバが起動していることを確認します。



- ④ しばらく待機していると、ブラウザが起動し、Webページが表示されます。
記述したコードのとおりに表示されていることを確認します。



- ⑤ 本手順で起動したTomcatサーバを停止する場合は、コンソール部にある **■** ボタンをクリックします。



- ⑥ Eclipseの終了は、右上の「×」ボタンで終了します。

以上