Programování v Javě II

Java pro webové aplikace

Michal Pobucký

michal.pobucky@fpf.slu.cz

Ústav informatiky – léto 2020

SLEZSKÁ UNIVERZITA
FILOZOFICKO-PŘÍRODOVĚDECKÁ FAKULTA V OPAVĚ

Podmínky absolvování předmětu

- Vytvoření webové aplikace – hry pro dva hráče (pexeso, lodě, piškvorky, početní soutěž,…)
  - Hráč hraje proti hráči, klient – server – klient (nehraje hráč proti PC)
  - Hráčů proti sobě může hrát libovolné množství (vždy v párech)
  - Hráč se musí nejprve přihlásit jménem a heslem na server (nový hráč se musí zaregistrovat)
  - Hráč čeká dokud se neobjeví protihráč
  - Celá hra se loguje na serveru
  - Klient je proveden v jednoduchém grafickém prostředí (pokud je to nutné)
  - Hráči není povoleno přihlásit se podruhé

Podmínky absolvování předmětu

- 1. část – webová aplikace s možností registrace, přihlášení, grafické rozložení, návrh databázového řešení
  - Odevzdat v polovině semestru
- 2. část – kompletní webová aplikace
  - Odevzdat v posledním týdnu semestru

Java pro webové aplikace

- Předpoklady
  - Znalost JSE
  - Základní znalost HTML, SQL, CSS
- JEE

Java pro webové aplikace

- Statický web
  - Klient odesílá požadavek na stránku na server
  - Server vrací zpět HTML stránku

- Dynamický web (využívá JEE)
  - Klient odesílá požadavek na stránku na server
  - Server volá JEE
  - JEE na základě požadavků klienta generuje HTML stránku
  - Server vrací zpět HTML stránku

Java pro webové aplikace

- Aplikační servery
  - Apache Tomcat
  - JBoss Application Server
  - IBM WebSphere Application Server
  - Oracle WebLogic

Java pro webové aplikace

- Technologie JEE
  - Java Servlets – základní nástroj pro obsluhu protokolu HTTP
  - JSP (Java Server Pages) – vkládané příkazy do HTML kódu
  - JSF (Java Server Faces) – celý kód v XML, skládá se z připravených komponent
  - JDBC (Java DataBase Connectivity) – rozhraní pro práci s databázemi pomocí SQL
  - JPA (Java Persistence API) – rozhraní pro práci s daty pomocí mezivrstvy ORM
  - Frameworky třetích stran – Spring, Struts, Hibernate

Java pro webové aplikace

- Trocha historie
  - Mainfraime – aplikační software uložen pouze zde, ostatní přistupují z jednotlivých stanic pomocí sítě
  - Desktopové aplikace – každá stanice vlastní software
  - Webové stránky – statické, architektura klient-server
  - Webové aplikace – dynamické, webová stránka se chová stejně jako desktopová aplikace

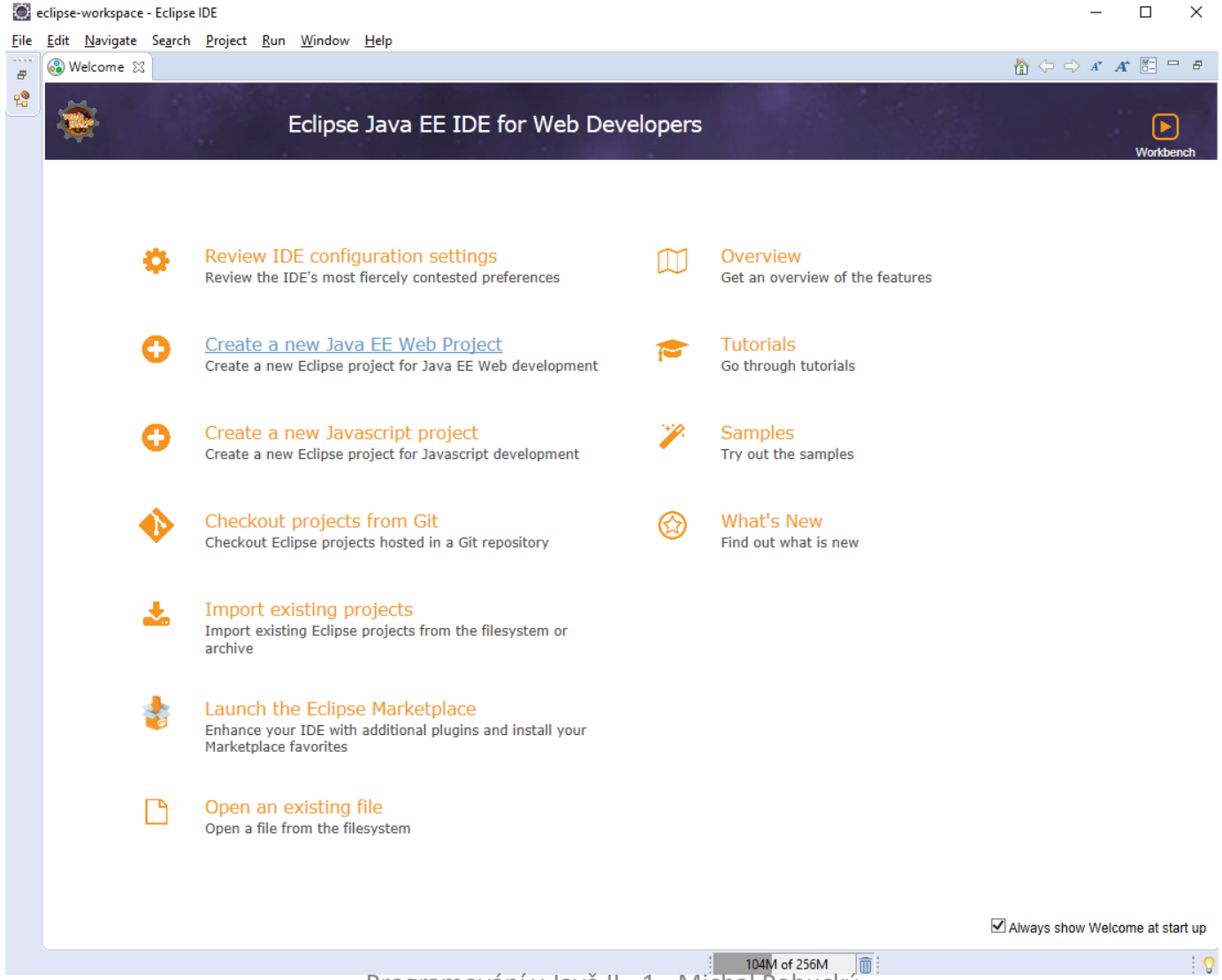
Java pro webové aplikace

- Vybavení
  - IDE – Eclipse, NetBeans, …
  - Server – Tomcat, GlassFish, …
- Instalace
  - Eclipse IDE for Enterprise Java Developers
  - Apache Tomcat

Java Servlet – úvod

- Vytvořte nový projekt – Create a new Java EE Web Project
  - Název: servlet01

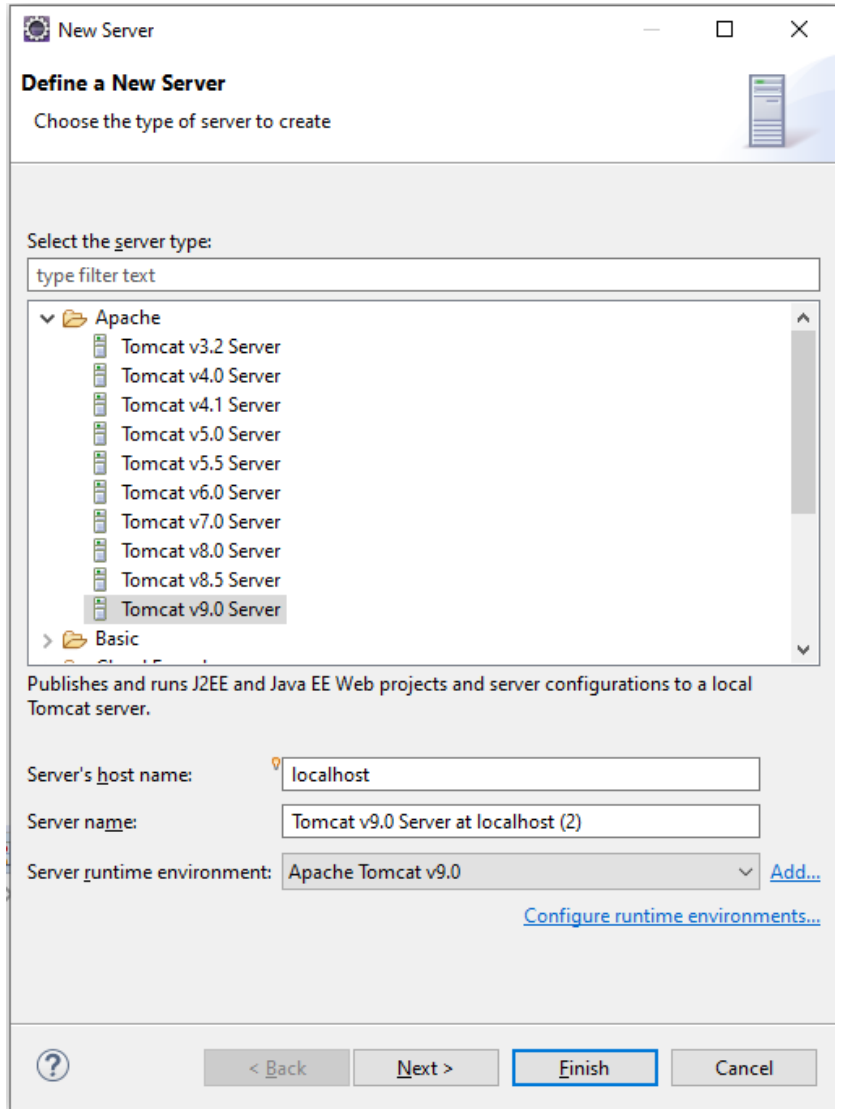
Java Servlet – úvod



Java Servlet – úvod

- Nastavení serveru
  - Window – Show view – Servers
  - V dialogovém okně – Define New Server
  - Vybrat Apache Tomcat v9.0 Server a odkázat na adresář, který jsme stáhli
  - Po instalaci lze server spustit pravým kliknutím myši - Start

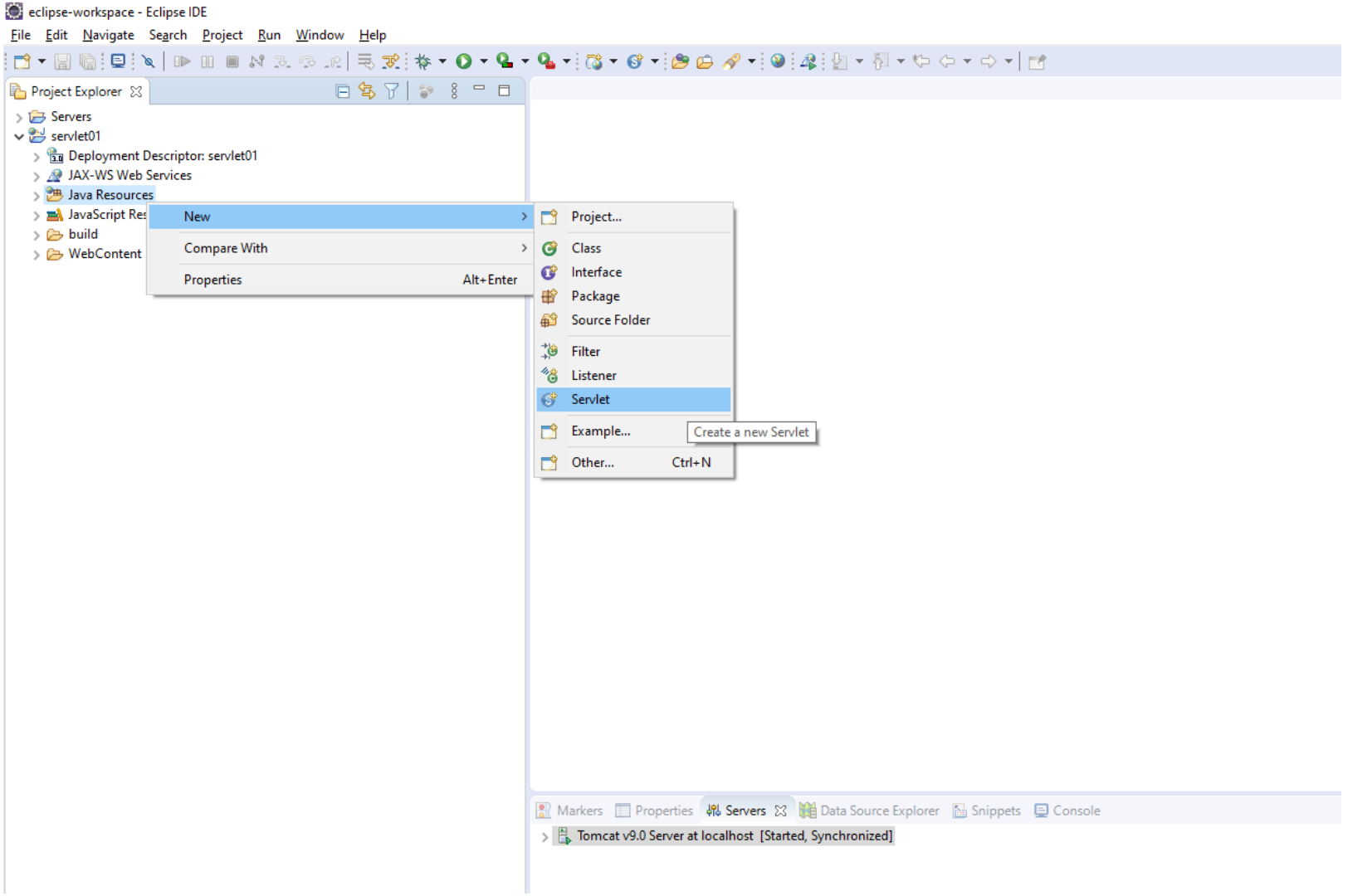
Java Servlet – úvod



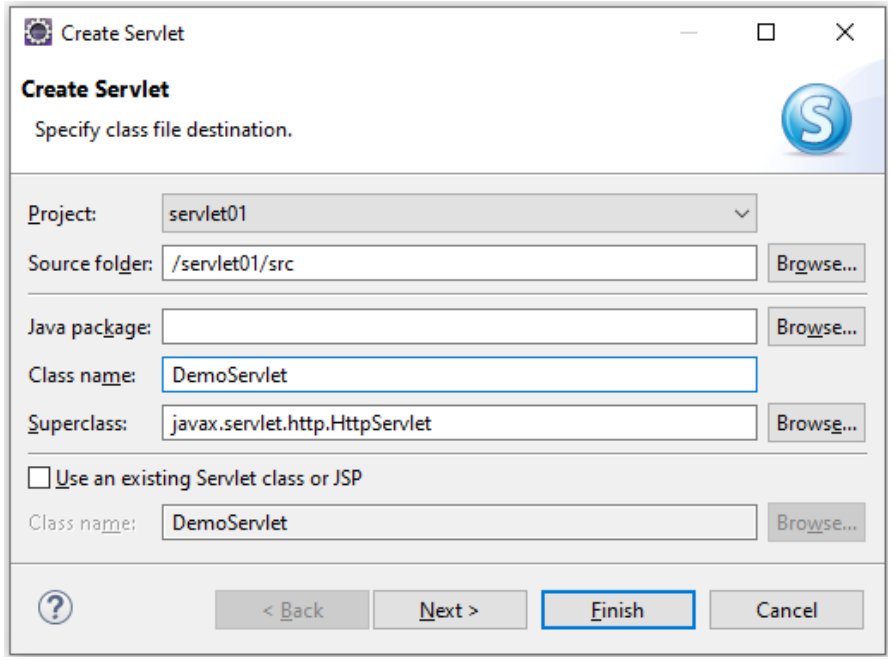
Java Servlet – úvod

- Vytvoříme nový servlet
  - Servlet01 – JavaResources – New – Servlet
  - ClassName označíme jako DemoServlet

Java Servlet – úvod



Java Servlet – úvod



Java Servlet – úvod

DemoServlet.java

```java

import java.io.IOException;

/**
 * Servlet implementation class DemoServlet
 */
@WebServlet("/DemoServlet")
public class DemoServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */
    public DemoServlet() {
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}

```

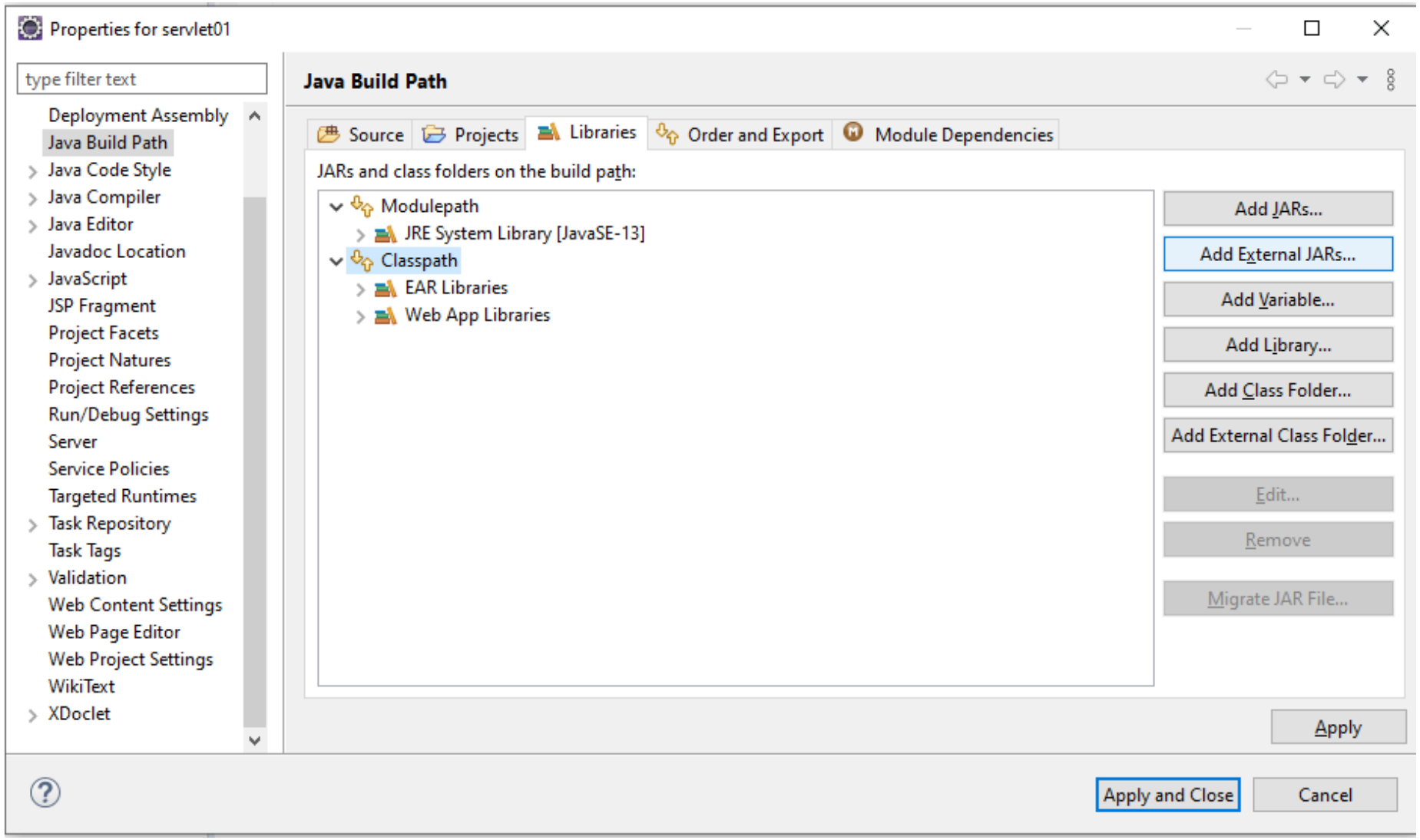
Java Servlet – úvod

- Chyba: The import javax.servlet cannot be resolved

- Oprava: kliknout pravou myší na jméno projektu – Properties – Java Build Path – Libraries – Add External JARs…
- Najít /apache-tomcat-9.0.30/lib/servlet-api.jar
- Klik na Apply and Close

- Chybová označení zmizela

Java Servlet – úvod



Java Servlet – úvod

- Doplníme import java.io.PrintWriter;
- Změníme metodu doGet()

Java Servlet – úvod

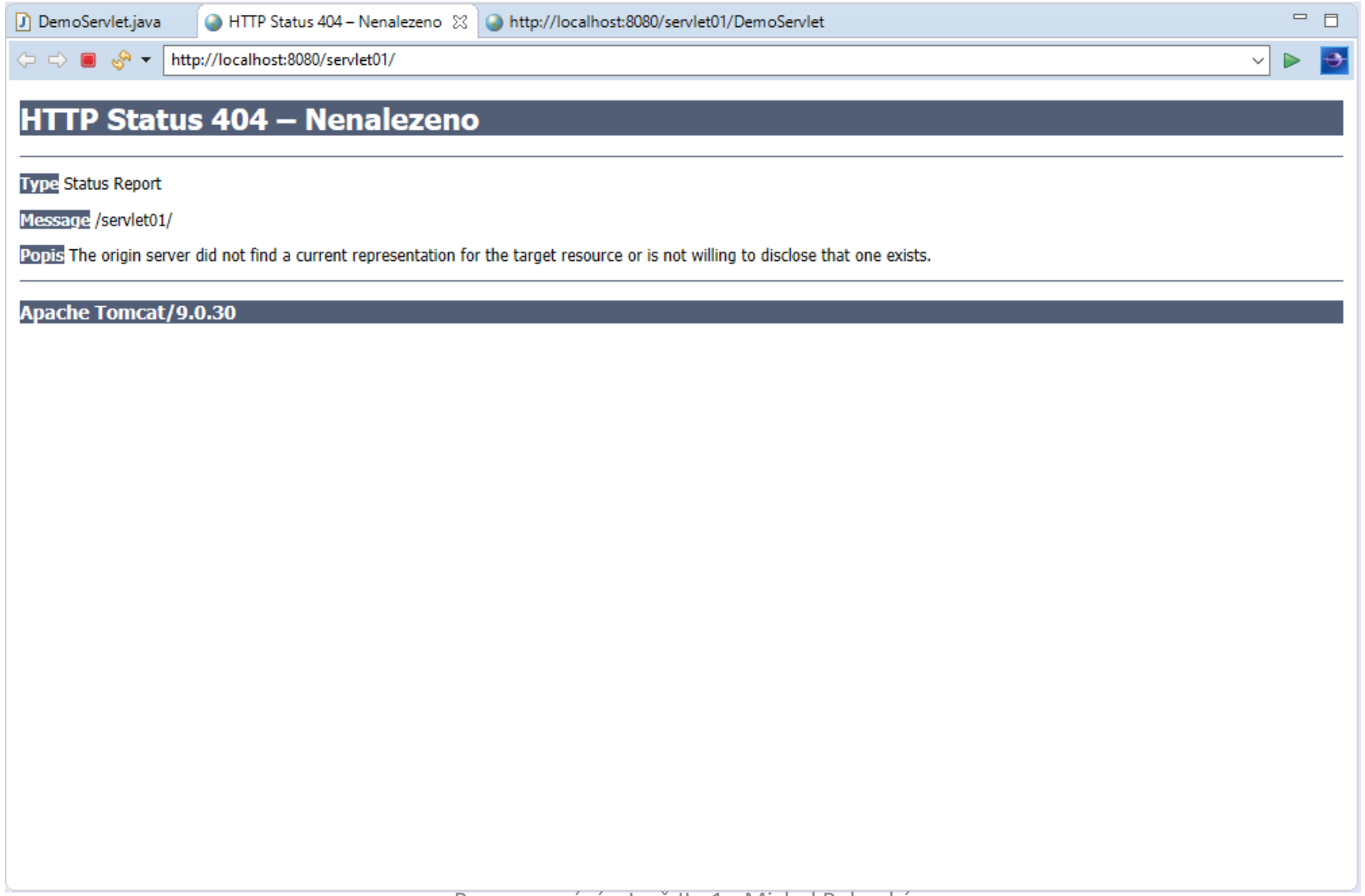
DemoServlet.java | HTTP Status 404 – Nenalezeno | http://localhost:8080/servlet01/DemoServlet

```java

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class DemoServlet
 */
@WebServlet("/DemoServlet")
public class DemoServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */
    public DemoServlet() {
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();

        out.print("<html><body>");
        out.print("<h3>První servlet</h3>");
        out.print("</body></html>");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

Java Servlet – úvod

- Spustíme tlačítkem Run As (Run On Server)

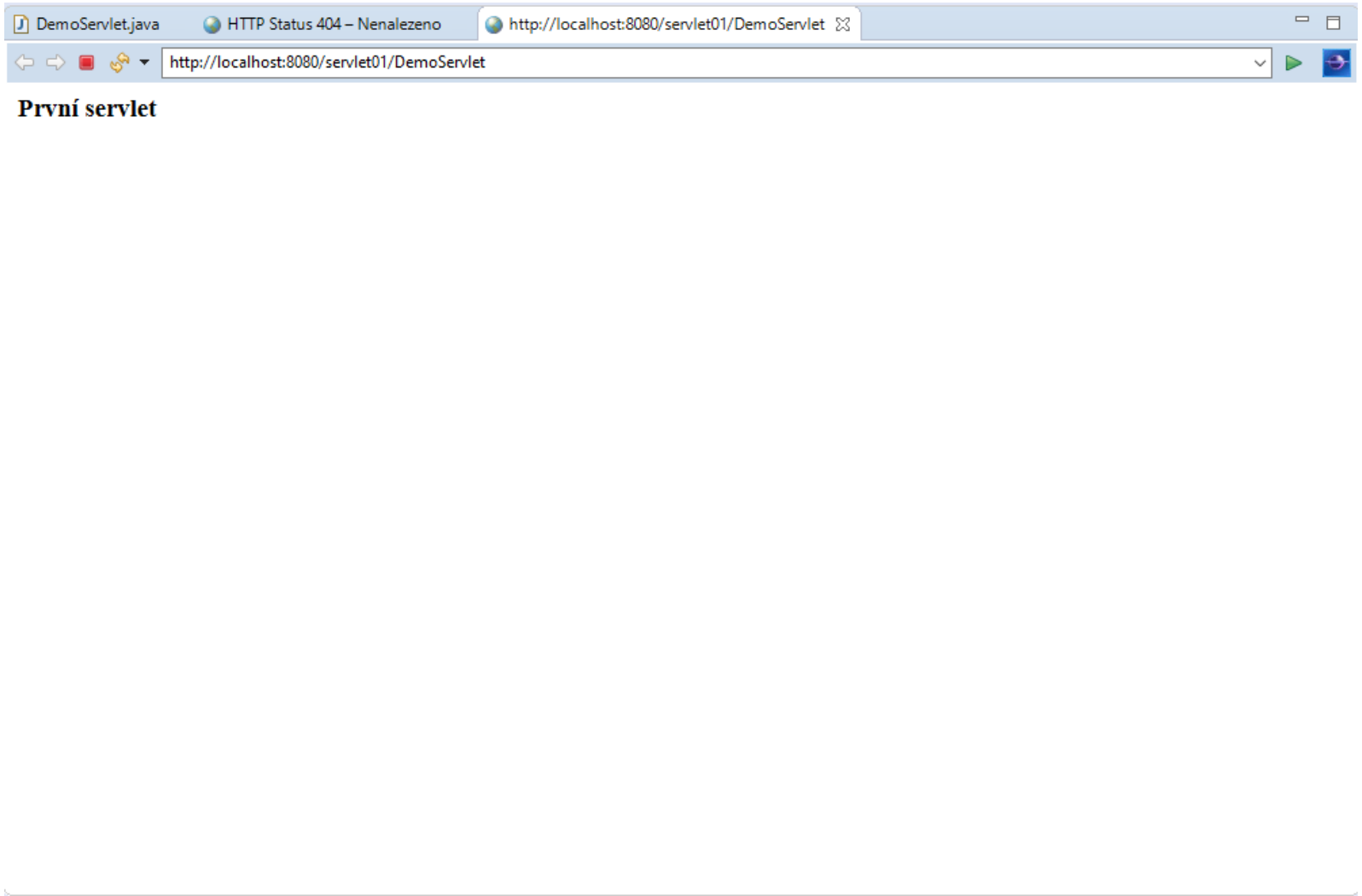
Java Servlet – úvod



Java Servlet – úvod

- Nemáme kód stránky, pouze servlet.
- Přepíšeme adresu z původní
  - http://localhost:8080/servlet01/
- Na novou adresu
  - http://localhost:8080/servlet01/DemoServlet

Java Servlet – úvod



Zdroje

- https://kore.fi.muni.cz/wiki/index.php/%C3%9Avod_do_webov%C3%BDch_aplikac%C3%AD
- https://www.zdrojak.cz/clanky/java-na-webovem-serveru-prvni-web/
- https://www.itnetwork.cz/java/jee
- https://www.javatpoint.com/java-tutorial