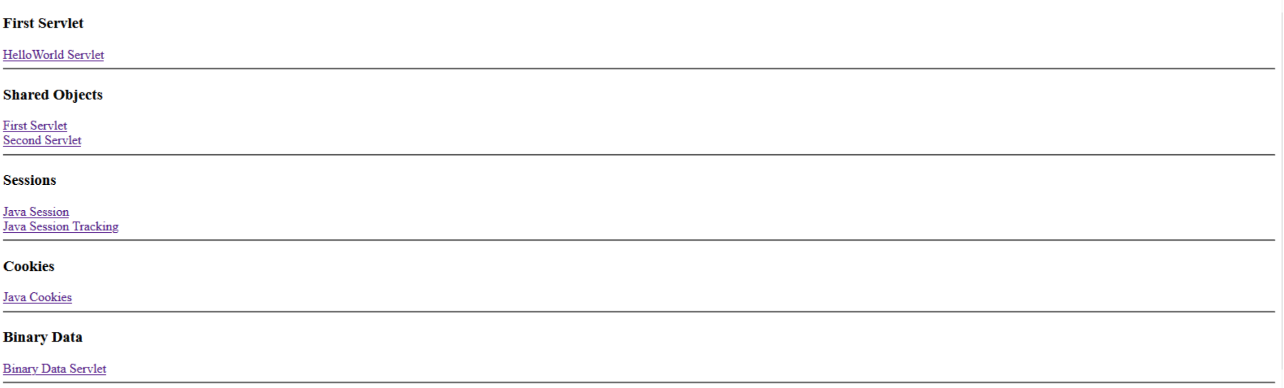
ИНТЕРНЕТ ПРОГРАМИРАЊЕ
ПРЕДАВАЊА – 005 – МАТЕРИЈАЛИ



First Servlet

HelloWorld Servlet

Shared Objects

First Servlet
Second Servlet

Sessions

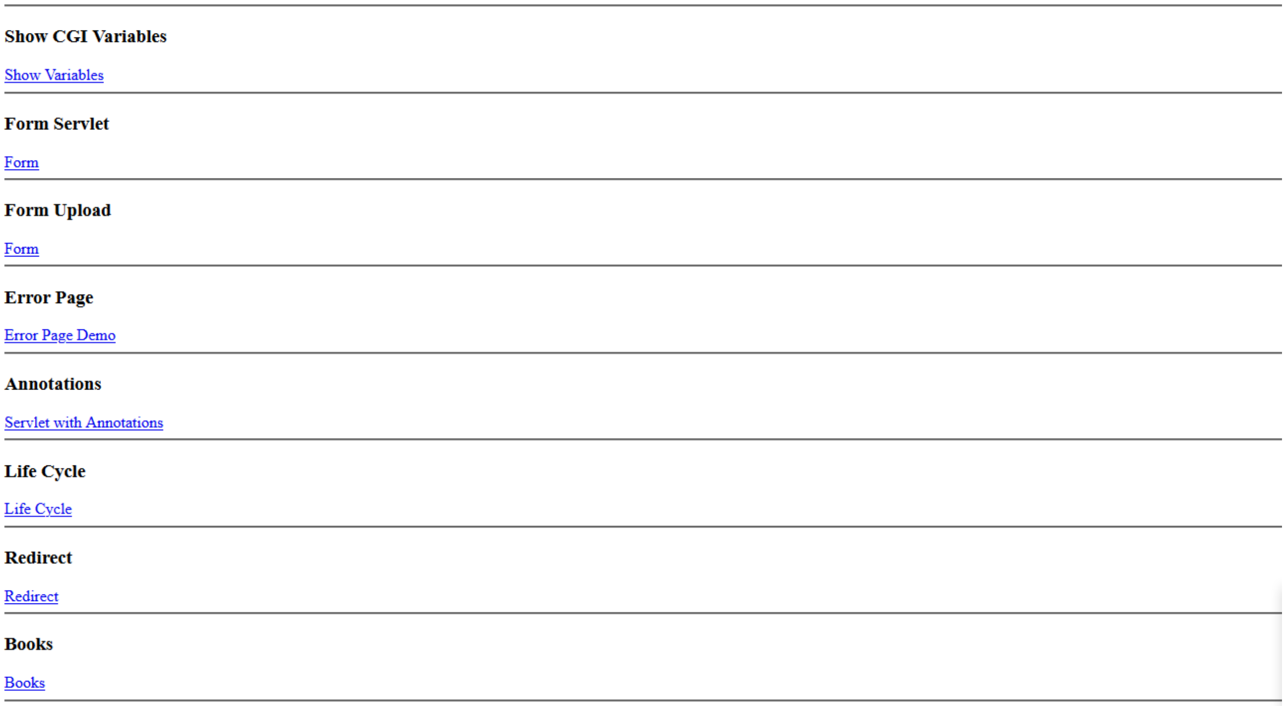
Java Session
Java Session Tracking

Cookies

Java Cookies

Binary Data

Binary Data Servlet



Show CGI Variables

Show Variables

Form Servlet

Form

Form Upload

Form

Error Page

Error Page Demo

Annotations

Servlet with Annotations

Life Cycle

Life Cycle

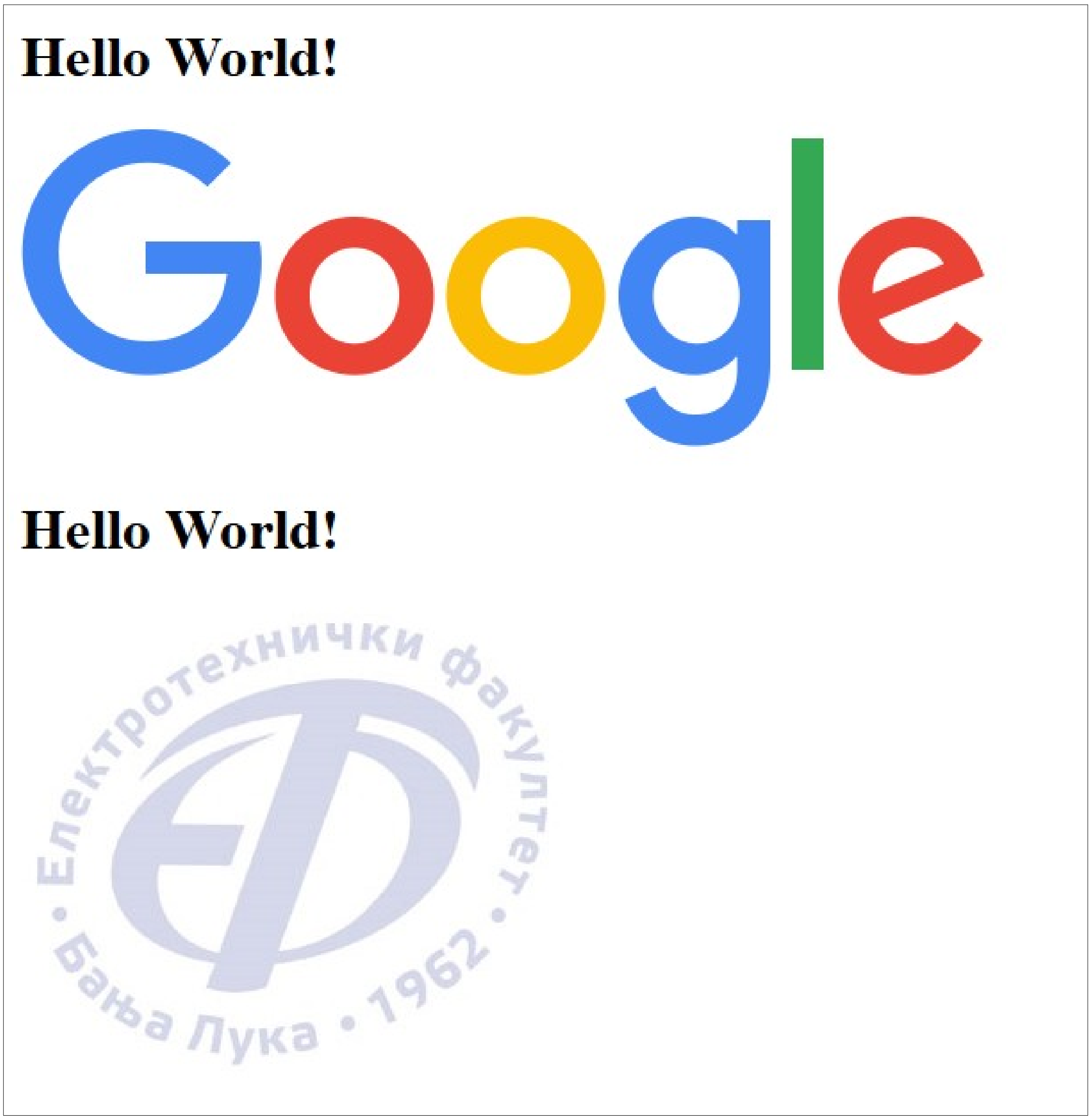
Redirect

Redirect

Books

Books

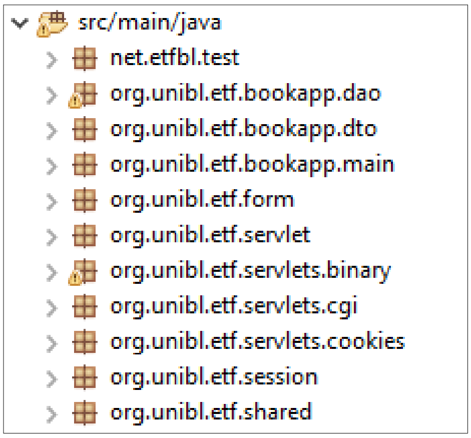
Hello World Servlet/Binary Data Servlet



Hello World!

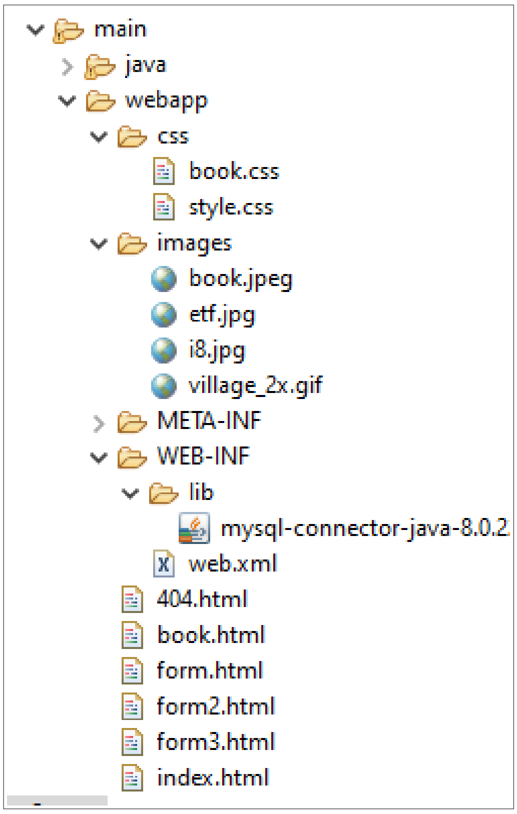
Hello World!

Raspored paketa



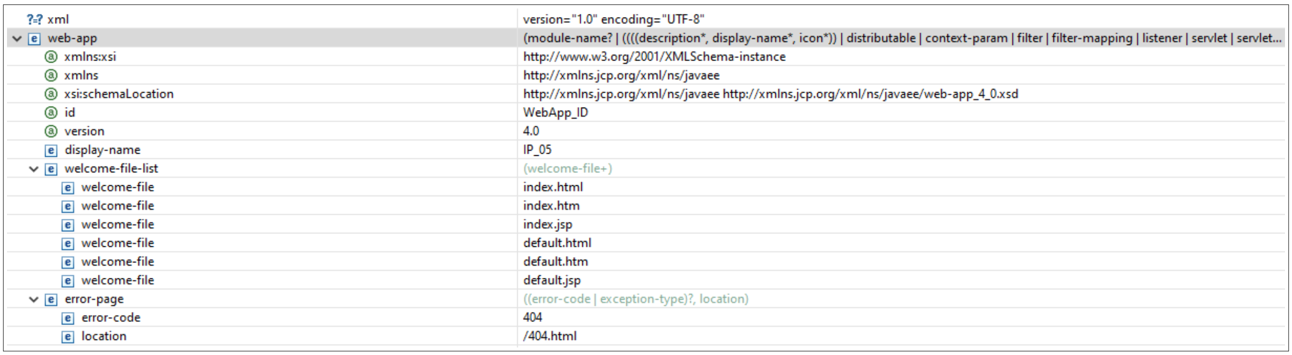
```
src/main/java
  net.etfbl.test
  org.unibl.etf.bookapp.dao
  org.unibl.etf.bookapp.dto
  org.unibl.etf.bookapp.main
  org.unibl.etf.form
  org.unibl.etf.servlet
  org.unibl.etf.servlets.binary
  org.unibl.etf.servlets.cgi
  org.unibl.etf.servlets.cookies
  org.unibl.etf.session
  org.unibl.etf.shared
```

Raspored veb stranica



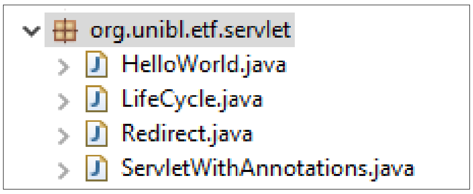
```
main
  java
  webapp
    css
      book.css
      style.css
    images
      book.jpeg
      etf.jpg
      i8.jpg
      village_2x.gif
    META-INF
    WEB-INF
      lib
        mysql-connector-java-8.0.2
      web.xml
    404.html
    book.html
    form.html
    form2.html
    form3.html
    index.html
```

web.xml



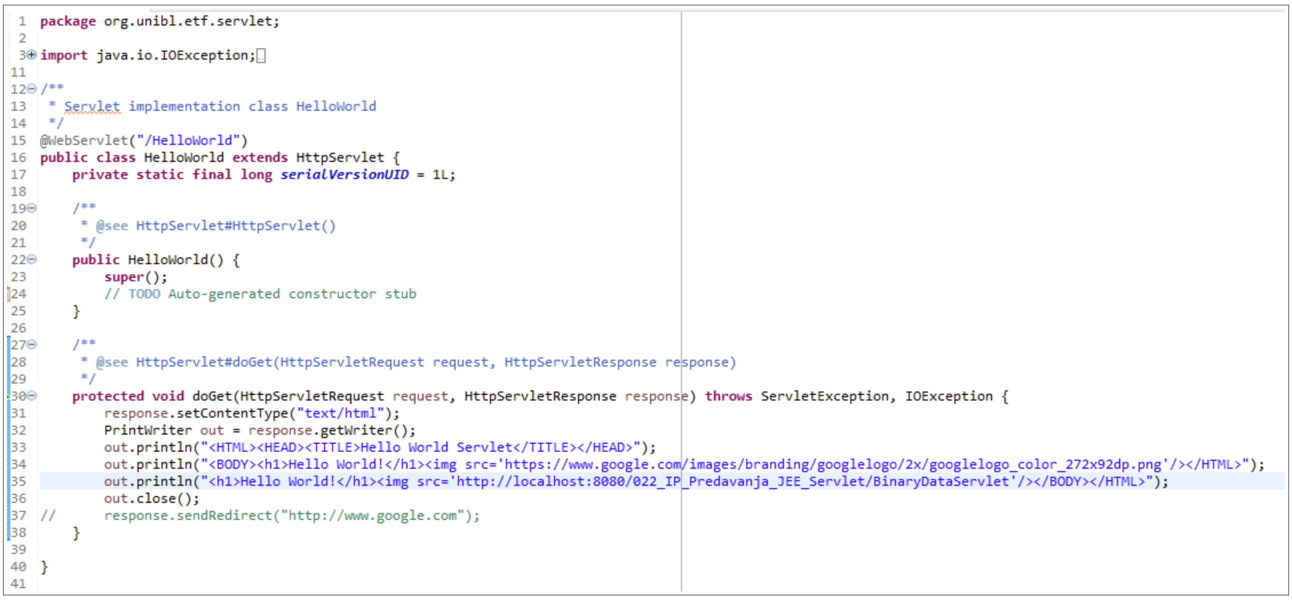
?-? xml	version="1.0" encoding="UTF-8"
web-app	(module-name? | ((((description*, display-name*, icon*)) | distributable | context-param | filter | filter-mapping | listener | servlet | servlet...
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xmlns	http://xmlns.jcp.org/xml/ns/javaee
xsi:schemaLocation	http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd
id	WebApp_ID
version	4.0
display-name	IP_05
welcome-file-list	(welcome-file+)
welcome-file	index.html
welcome-file	index.htm
welcome-file	index.jsp
welcome-file	default.html
welcome-file	default.htm
welcome-file	default.jsp
error-page	((error-code | exception-type)?, location)
error-code	404
location	/404.html

servleti



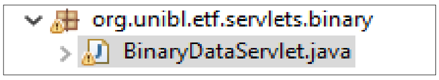
```
org.unibl.etf.servlet
  HelloWorld.java
  LifeCycle.java
  Redirect.java
  ServletWithAnnotations.java
```

HelloServlet.java



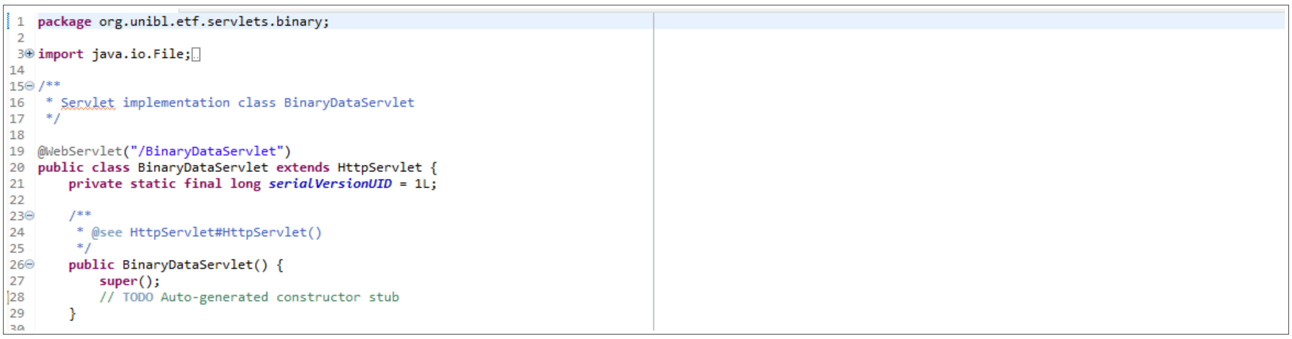
```java
package org.unibl.etf.servlet;

import java.io.IOException;□

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HelloWorld() {
        super();
        // 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.println("<HTML><HEAD><TITLE>Hello World Servlet</TITLE></HEAD>");
        out.println("<BODY><h1>Hello World!</h1><img src='https://www.google.com/images/branding/googlelogo/2x/googlelogo_color_272x92dp.png'/></HTML>");
        out.println("<h1>Hello World!</h1><img src='http://localhost:8080/022_IP_Predavanja_JEE_Servlet/BinaryDataServlet'/></BODY></HTML>");
        out.close();
//      response.sendRedirect("http://www.google.com");
    }

}

```

binarni servleti

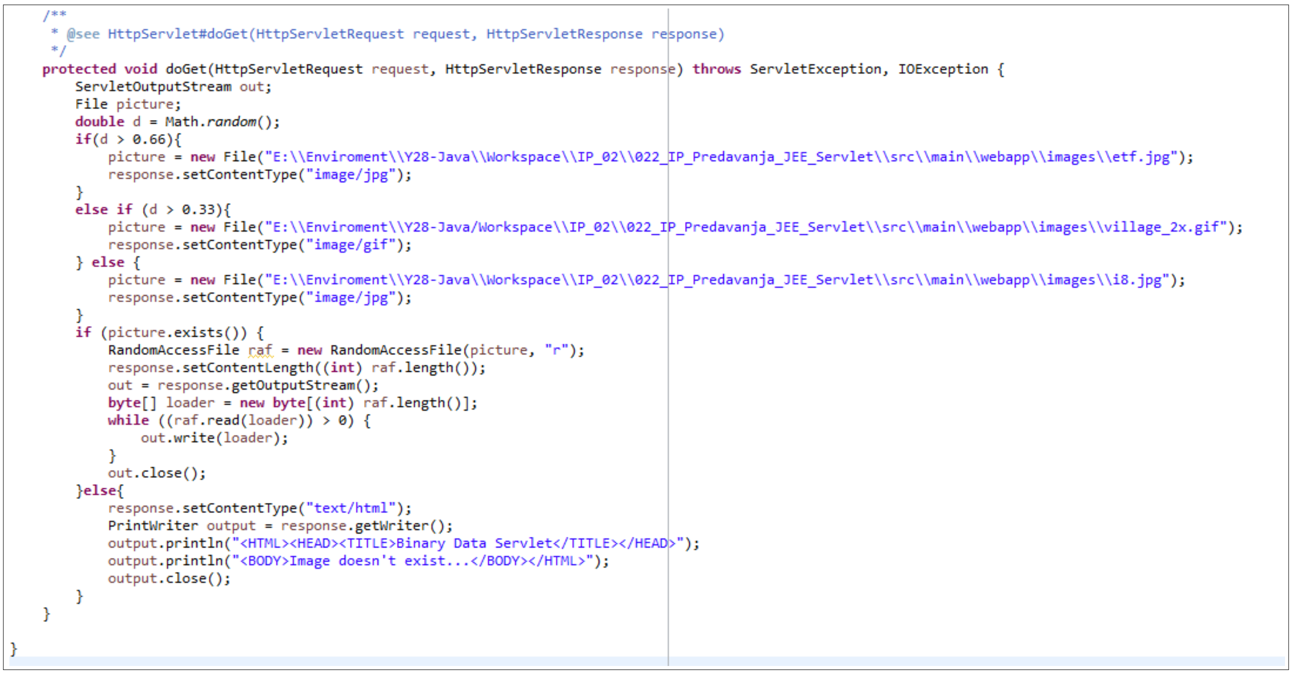


```
org.unibl.etf.servlets.binary
  BinaryDataServlet.java
```

BinaryServlet.java

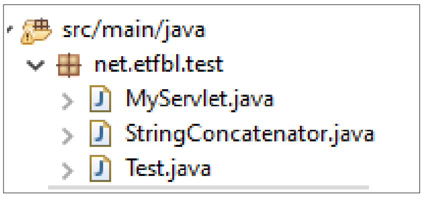


```java
package org.unibl.etf.servlets.binary;

import java.io.File;□

/**
 * Servlet implementation class BinaryDataServlet
 */

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public BinaryDataServlet() {
        super();
        // TODO Auto-generated constructor stub
    }
```



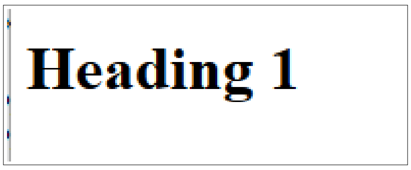
```java
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ServletOutputStream out;
        File picture;
        double d = Math.random();
        if(d > 0.66){
            picture = new File("E:\\Enviroment\\Y28-Java\\Workspace\\IP_02\\022_IP_Predavanja_JEE_Servlet\\src\\main\\webapp\\images\\etf.jpg");
            response.setContentType("image/jpg");
        }
        else if (d > 0.33){
            picture = new File("E:\\Enviroment\\Y28-Java/Workspace\\IP_02\\022_IP_Predavanja_JEE_Servlet\\src\\main\\webapp\\images\\village_2x.gif");
            response.setContentType("image/gif");
        } else {
            picture = new File("E:\\Enviroment\\Y28-Java\\Workspace\\IP_02\\022_IP_Predavanja_JEE_Servlet\\src\\main\\webapp\\images\\i8.jpg");
            response.setContentType("image/jpg");
        }
        if (picture.exists()) {
            RandomAccessFile raf = new RandomAccessFile(picture, "r");
            response.setContentLength((int) raf.length());
            out = response.getOutputStream();
            byte[] loader = new byte[(int) raf.length()];
            while ((raf.read(loader)) > 0) {
                out.write(loader);
            }
            out.close();
        }else{
            response.setContentType("text/html");
            PrintWriter output = response.getWriter();
            output.println("<HTML><HEAD><TITLE>Binary Data Servlet</TITLE></HEAD>");
            output.println("<BODY>Image doesn't exist...</BODY></HTML>");
            output.close();
        }
    }

}
```

test servlet

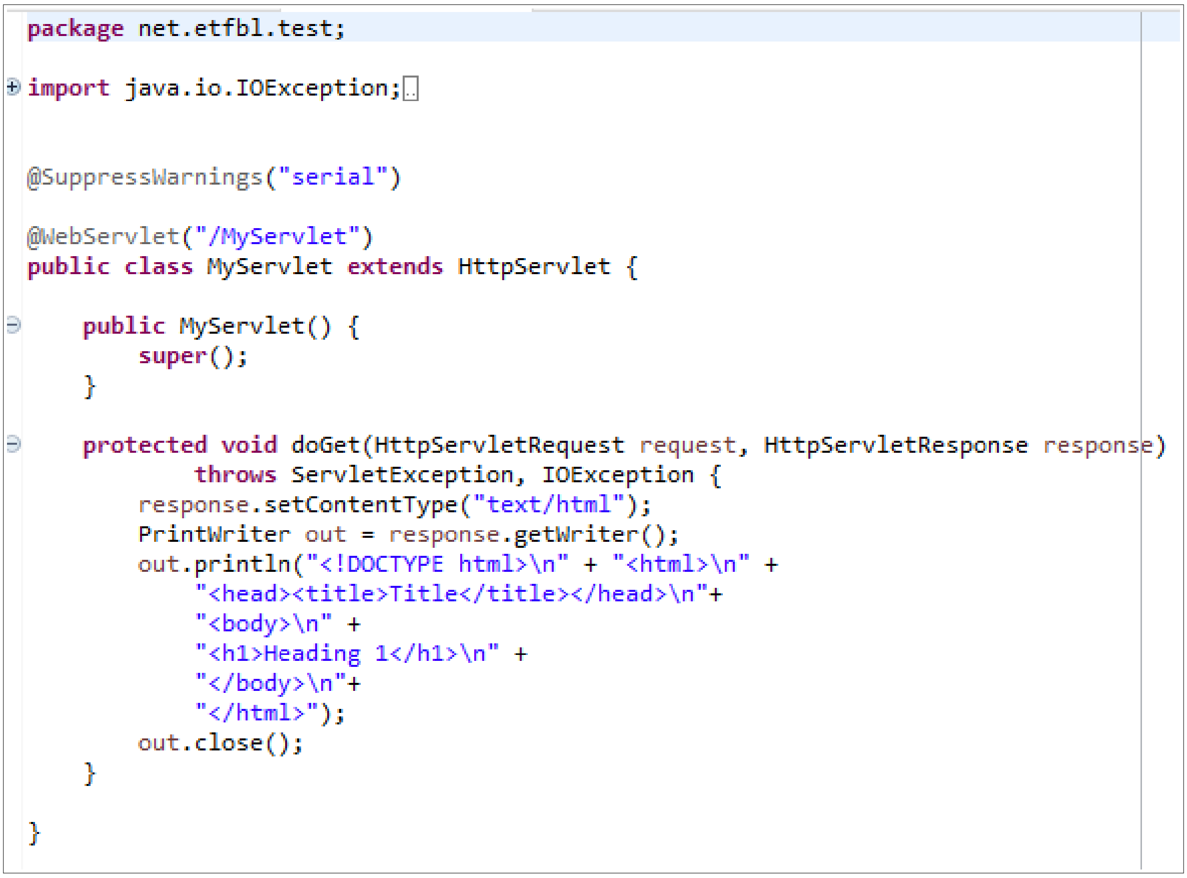


```
src/main/java
  net.etfbl.test
    MyServlet.java
    StringConcatenator.java
    Test.java
```

MyServlet.java

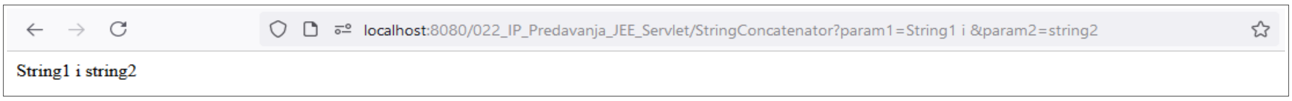


Heading 1

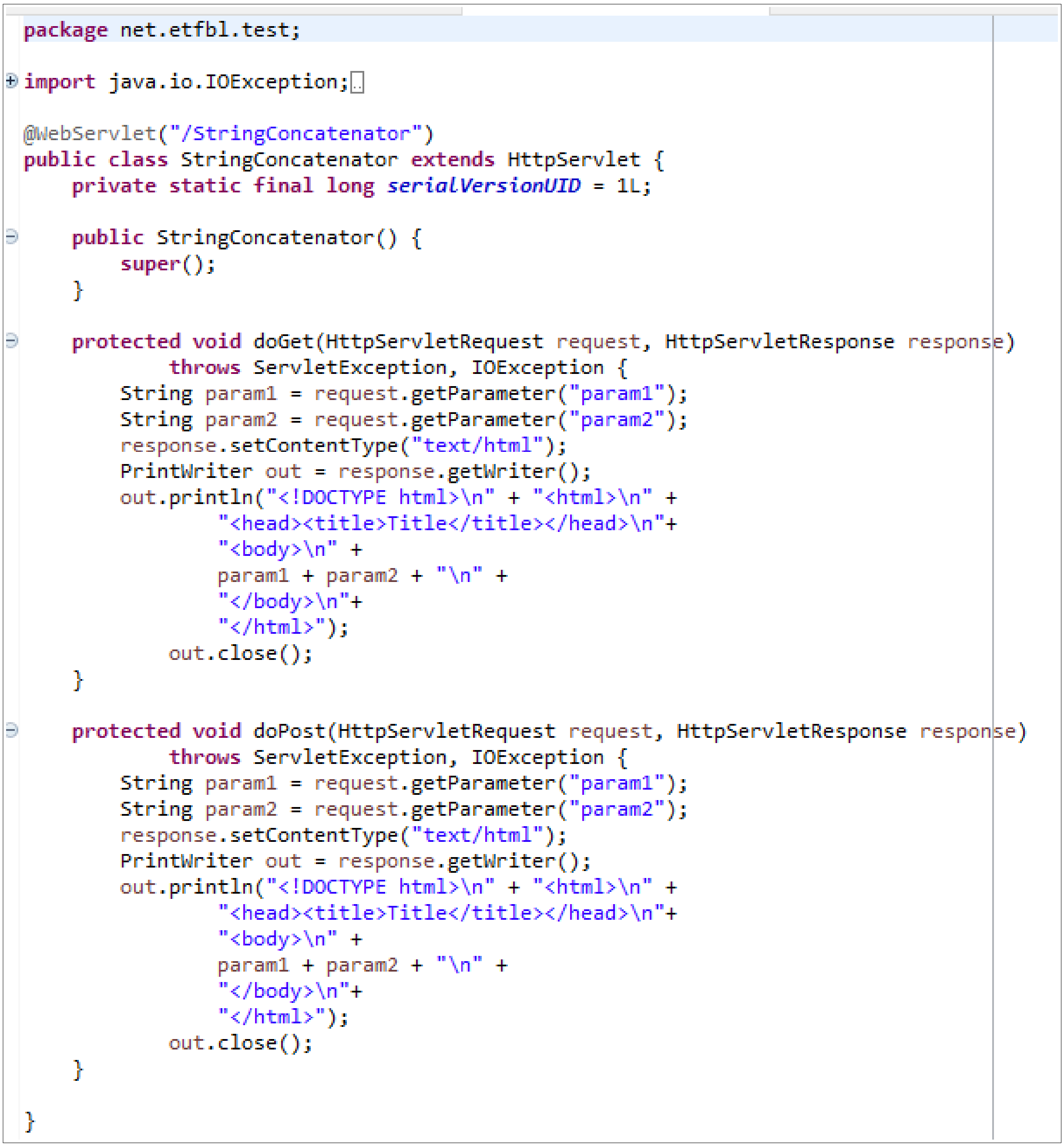


```java
package net.etfbl.test;

import java.io.IOException;


@SuppressWarnings("serial")

@WebServlet("/MyServlet")
public class MyServlet extends HttpServlet {

    public MyServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html>\n" + "<html>\n" +
            "<head><title>Title</title></head>\n"+
            "<body>\n" +
            "<h1>Heading 1</h1>\n" +
            "</body>\n"+
            "</html>");
        out.close();
    }

}
```

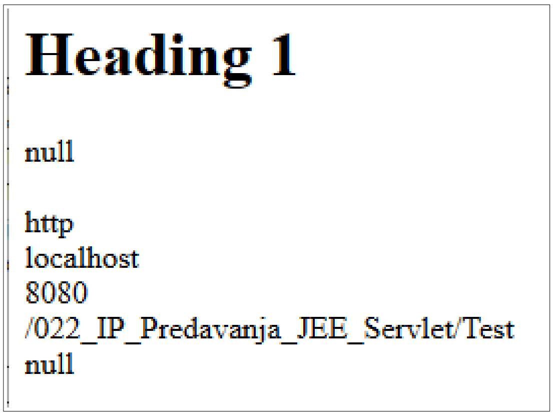
StringConcatenator.java


localhost:8080/022_IP_Predavanja_JEE_Servlet/StringConcatenator?param1=String1 i &param2=string2

String1 i string2


```java
package net.etfbl.test;

import java.io.IOException;

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

    public StringConcatenator() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String param1 = request.getParameter("param1");
        String param2 = request.getParameter("param2");
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html>\n" + "<html>\n" +
                "<head><title>Title</title></head>\n"+
                "<body>\n" +
                param1 + param2 + "\n" +
                "</body>\n"+
                "</html>");
            out.close();
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String param1 = request.getParameter("param1");
        String param2 = request.getParameter("param2");
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html>\n" + "<html>\n" +
                "<head><title>Title</title></head>\n"+
                "<body>\n" +
                param1 + param2 + "\n" +
                "</body>\n"+
                "</html>");
            out.close();
    }

}
```

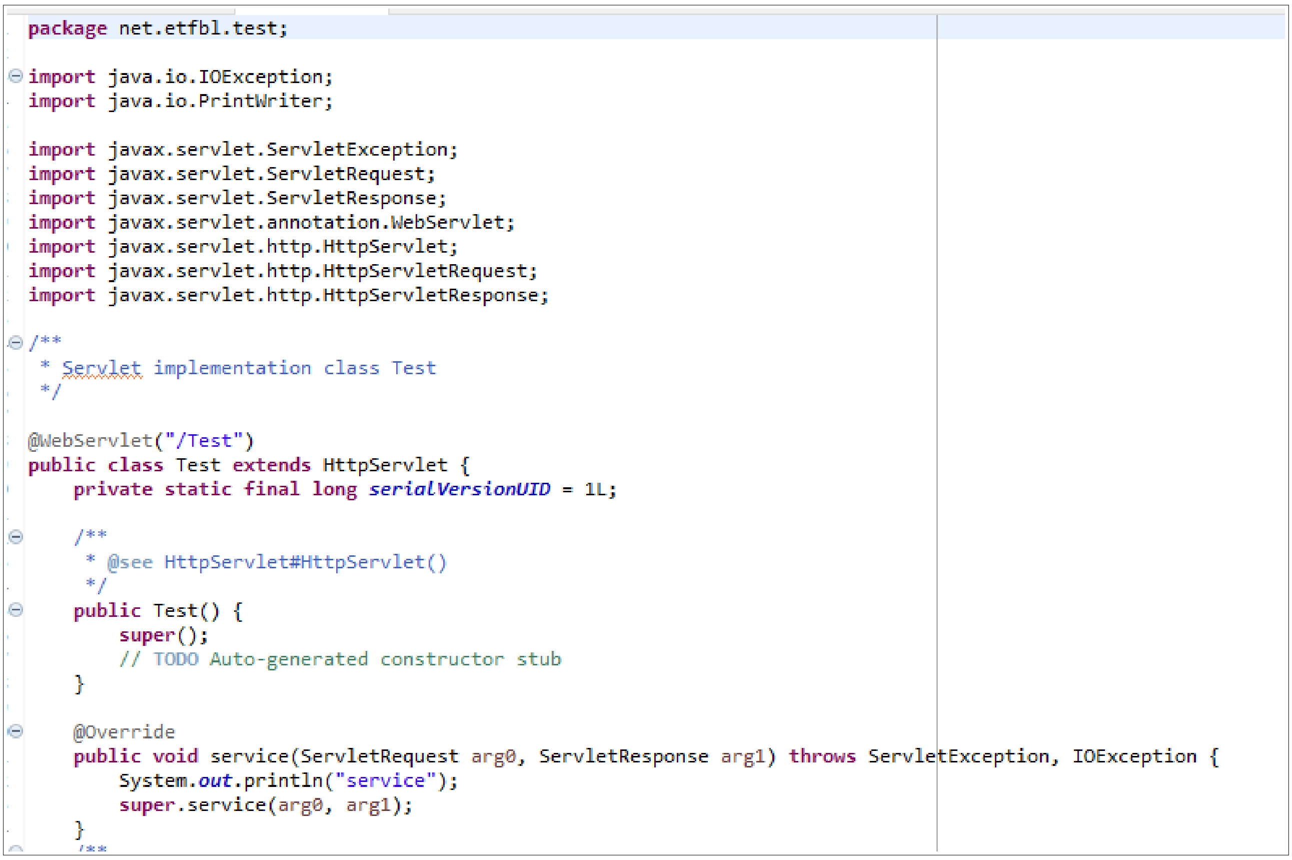
Test.java



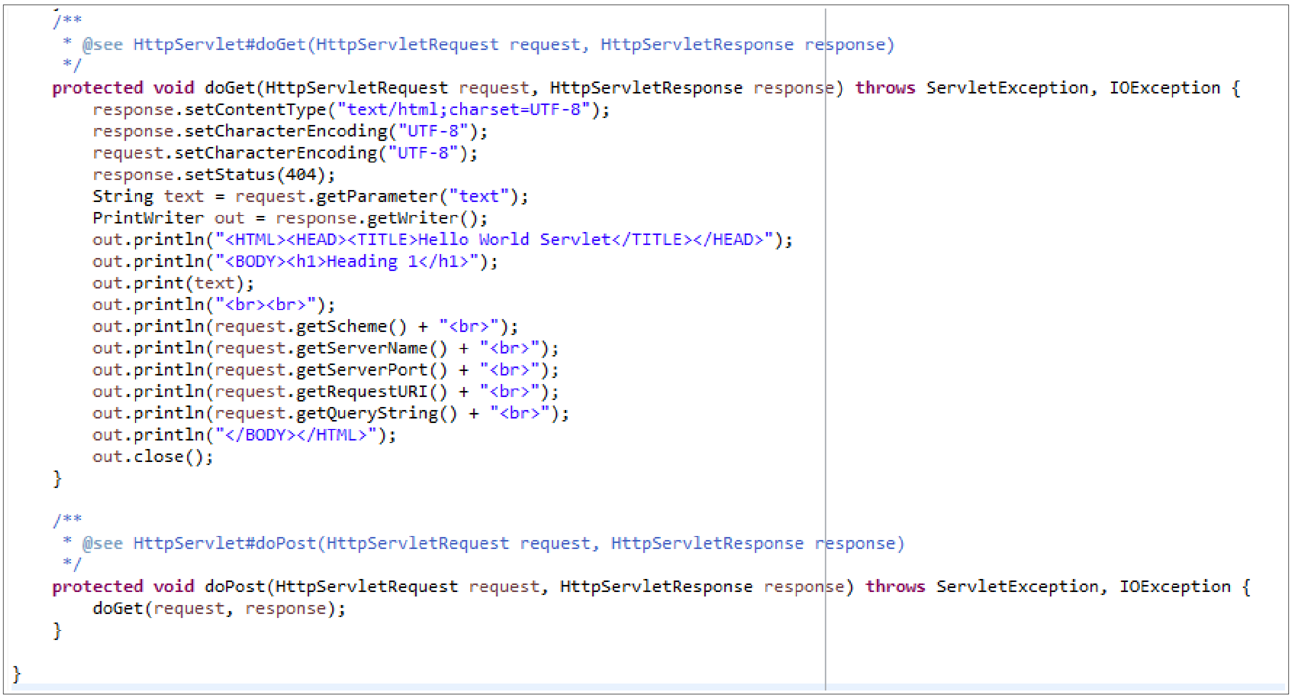
Heading 1

null

http
localhost
8080
/022_IP_Predavanja_JEE_Servlet/Test
null

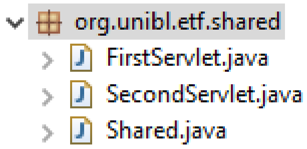


```java
package net.etfbl.test;

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

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

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Test() {
        super();
        // TODO Auto-generated constructor stub
    }

    @Override
    public void service(ServletRequest arg0, ServletResponse arg1) throws ServletException, IOException {
        System.out.println("service");
        super.service(arg0, arg1);
    }
    /**
```

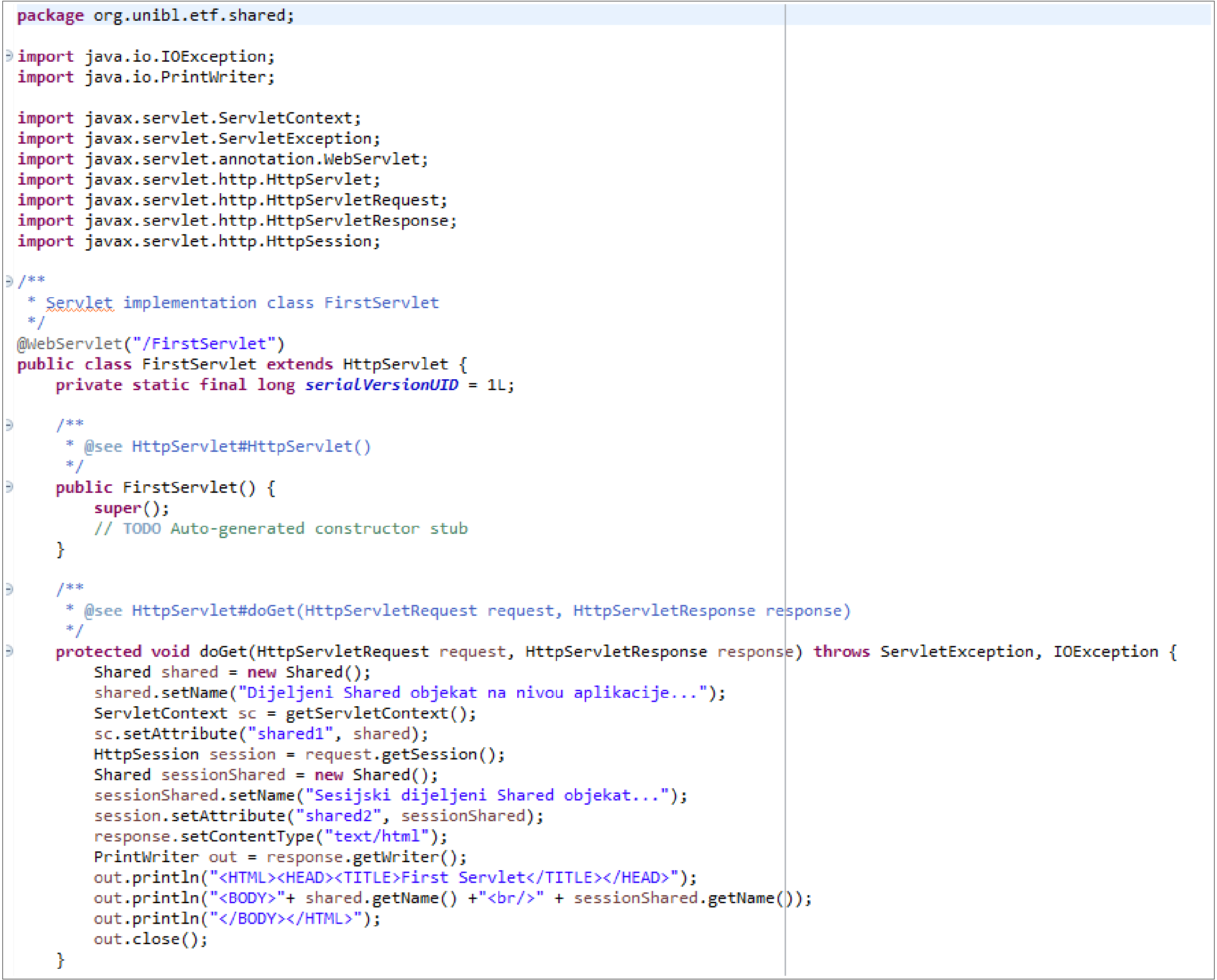


```
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        response.setCharacterEncoding("UTF-8");
        request.setCharacterEncoding("UTF-8");
        response.setStatus(404);
        String text = request.getParameter("text");
        PrintWriter out = response.getWriter();
        out.println("<HTML><HEAD><TITLE>Hello World Servlet</TITLE></HEAD>");
        out.println("<BODY><h1>Heading 1</h1>");
        out.print(text);
        out.println("<br><br>");
        out.println(request.getScheme() + "<br>");
        out.println(request.getServerName() + "<br>");
        out.println(request.getServerPort() + "<br>");
        out.println(request.getRequestURI() + "<br>");
        out.println(request.getQueryString() + "<br>");
        out.println("</BODY></HTML>");
        out.close();
    }

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

}
```

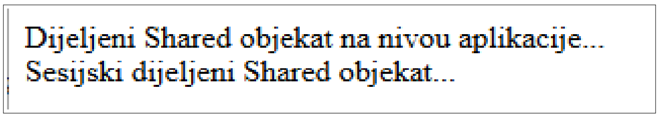
shared servlet



FirstServlet.java

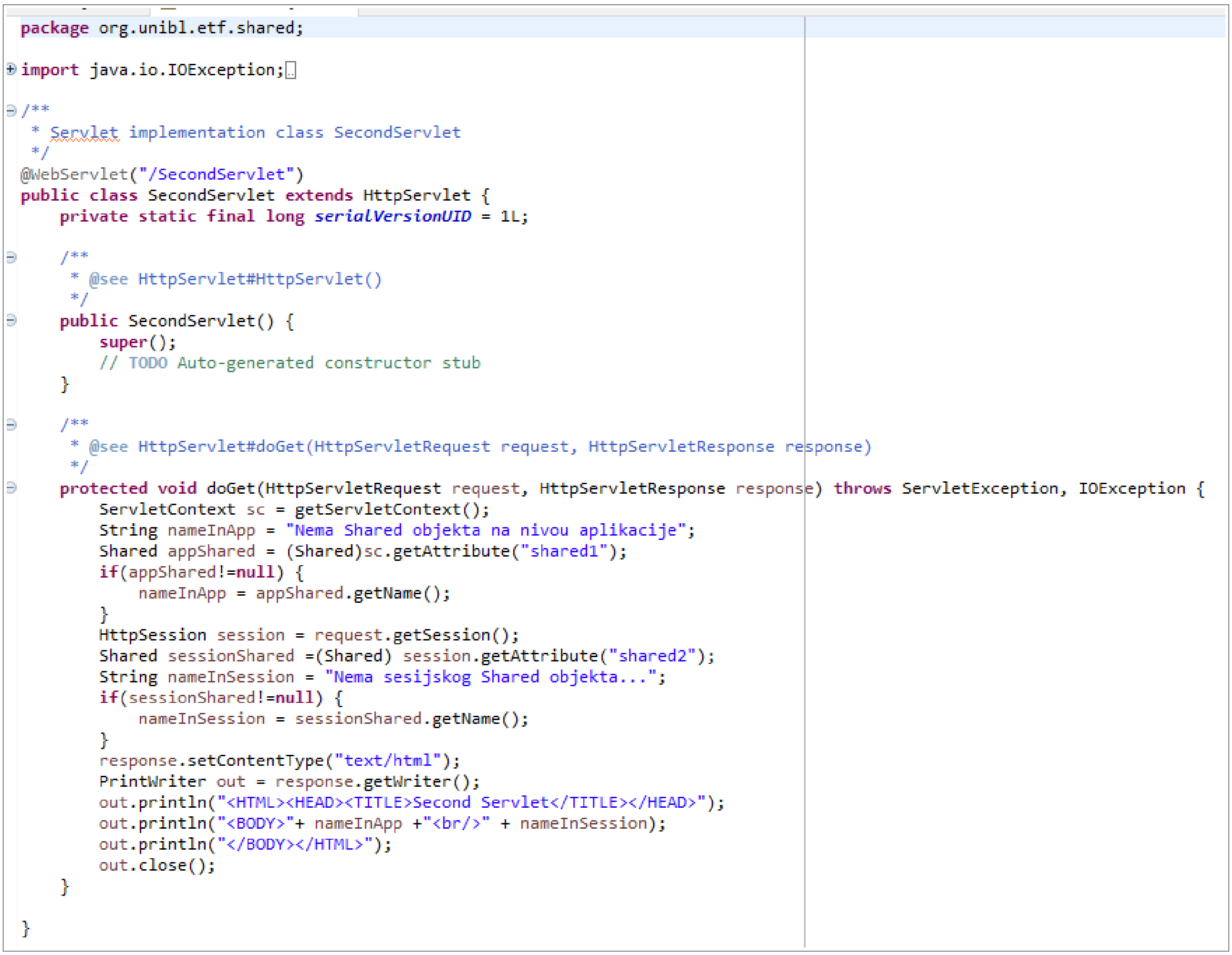


```
package org.unibl.etf.shared;

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

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

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public FirstServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        Shared shared = new Shared();
        shared.setName("Dijeljeni Shared objekat na nivou aplikacije...");
        ServletContext sc = getServletContext();
        sc.setAttribute("shared1", shared);
        HttpSession session = request.getSession();
        Shared sessionShared = new Shared();
        sessionShared.setName("Sesijski dijeljeni Shared objekat...");
        session.setAttribute("shared2", sessionShared);
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<HTML><HEAD><TITLE>First Servlet</TITLE></HEAD>");
        out.println("<BODY>"+ shared.getName() +"<br/>" + sessionShared.getName());
        out.println("</BODY></HTML>");
        out.close();
    }
```

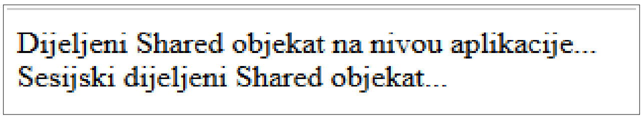


```
Dijeljeni Shared objekat na nivou aplikacije...
Sesijski dijeljeni Shared objekat...
```

SecondServlet.java

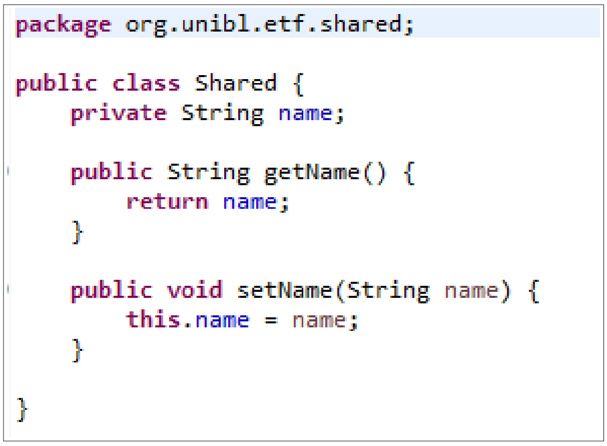


```java
package org.unibl.etf.shared;

import java.io.IOException;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public SecondServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ServletContext sc = getServletContext();
        String nameInApp = "Nema Shared objekta na nivou aplikacije";
        Shared appShared = (Shared)sc.getAttribute("shared1");
        if(appShared!=null) {
            nameInApp = appShared.getName();
        }
        HttpSession session = request.getSession();
        Shared sessionShared =(Shared) session.getAttribute("shared2");
        String nameInSession = "Nema sesijskog Shared objekta...";
        if(sessionShared!=null) {
            nameInSession = sessionShared.getName();
        }
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<HTML><HEAD><TITLE>Second Servlet</TITLE></HEAD>");
        out.println("<BODY>"+ nameInApp +"<br/>" + nameInSession);
        out.println("</BODY></HTML>");
        out.close();
    }

}
```



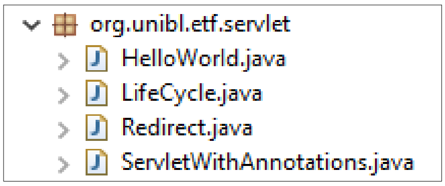
Dijeljeni Shared objekat na nivou aplikacije...
Sesijski dijeljeni Shared objekat...

Shared.java



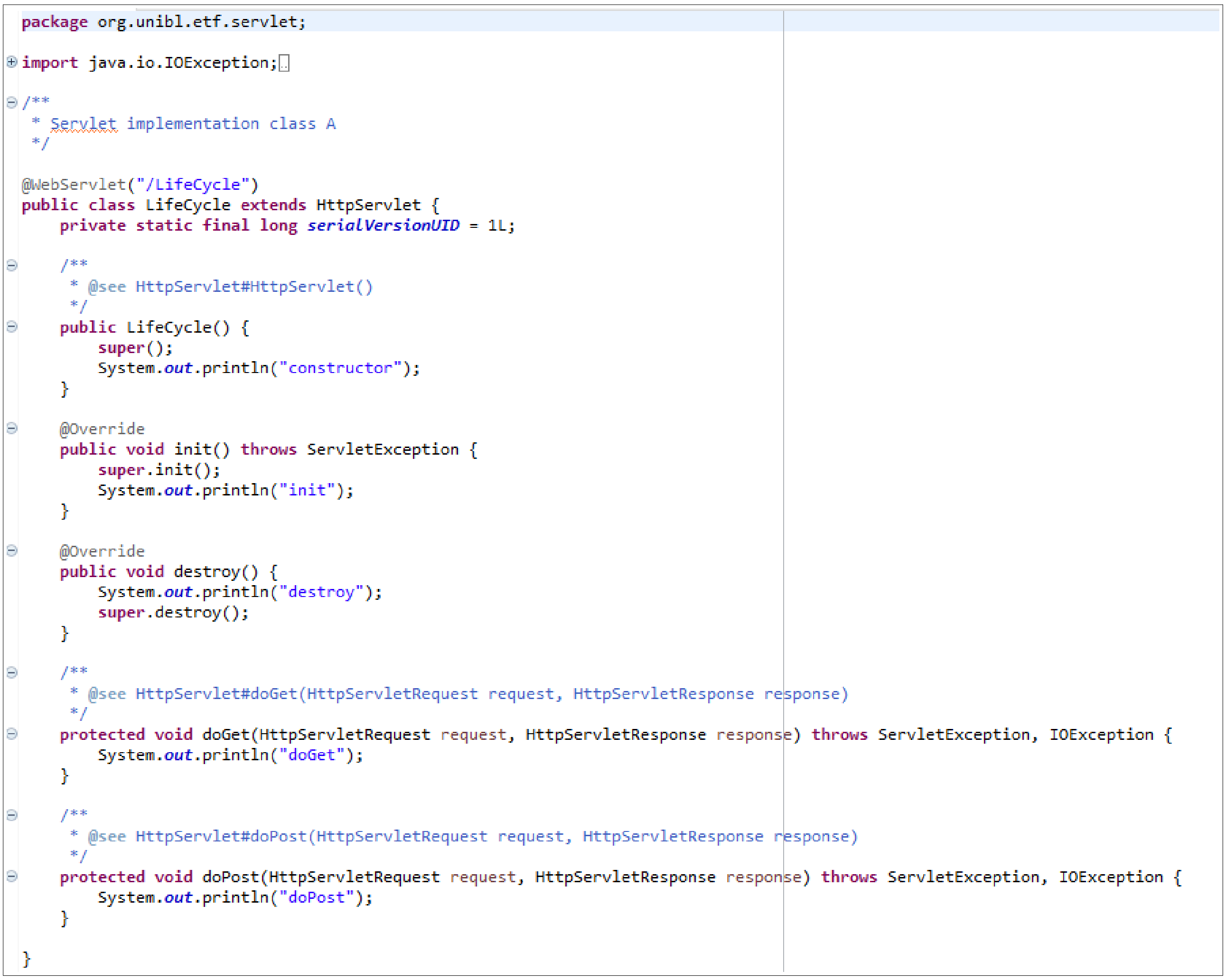
```java
package org.unibl.etf.shared;

public class Shared {
    private String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

}
```

servlet

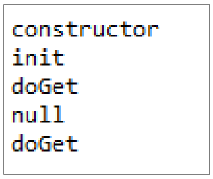


```
org.unibl.etf.servlet
    HelloWorld.java
    LifeCycle.java
    Redirect.java
    ServletWithAnnotations.java
```

LifeCycleServlet

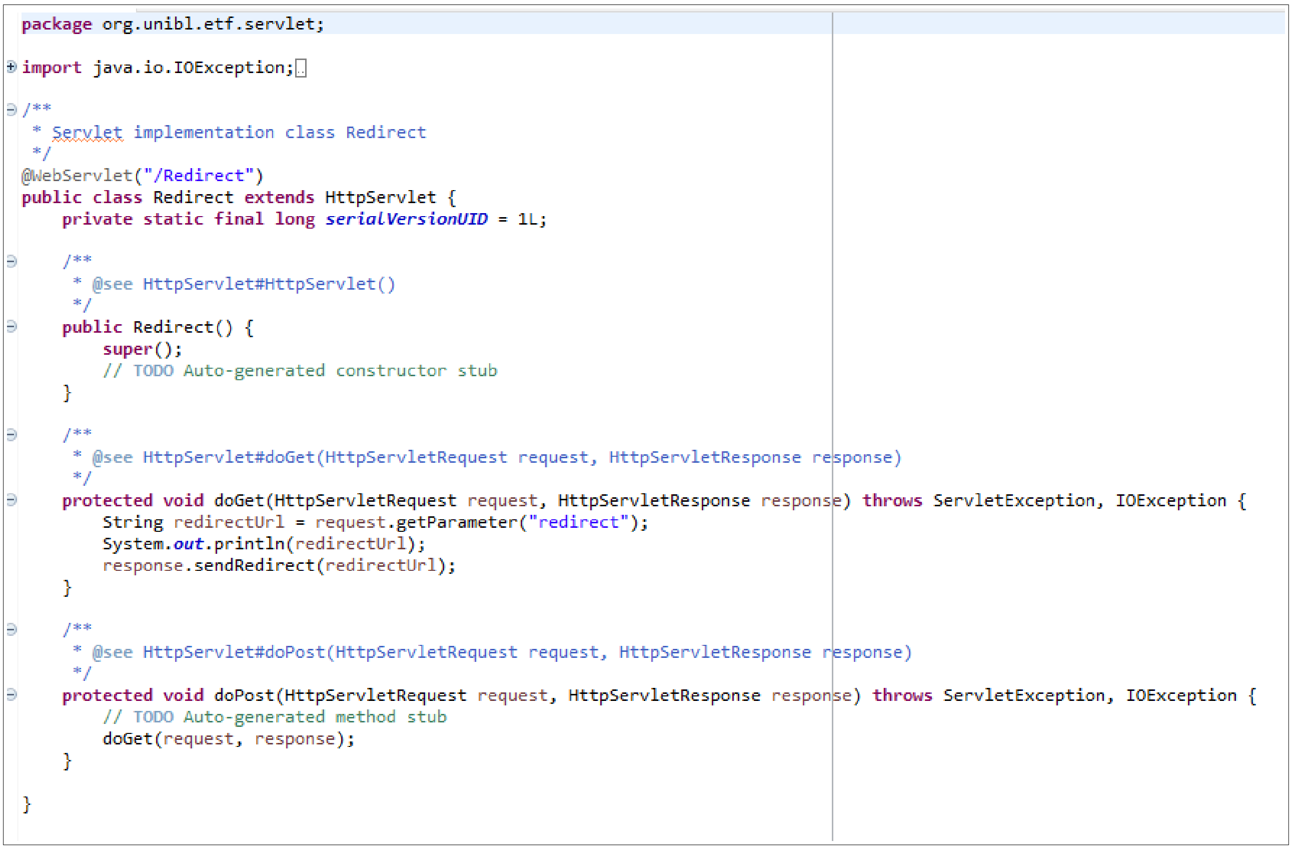


```
package org.unibl.etf.servlet;

import java.io.IOException;

/**
 * Servlet implementation class A
 */

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LifeCycle() {
        super();
        System.out.println("constructor");
    }

    @Override
    public void init() throws ServletException {
        super.init();
        System.out.println("init");
    }

    @Override
    public void destroy() {
        System.out.println("destroy");
        super.destroy();
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        System.out.println("doGet");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        System.out.println("doPost");
    }

}
```



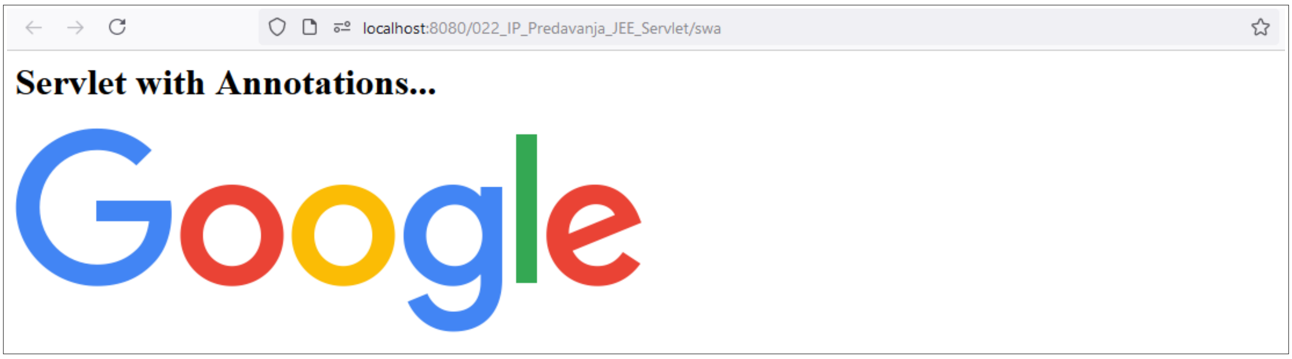
```
constructor
init
doGet
null
doGet
```

Redirect.java

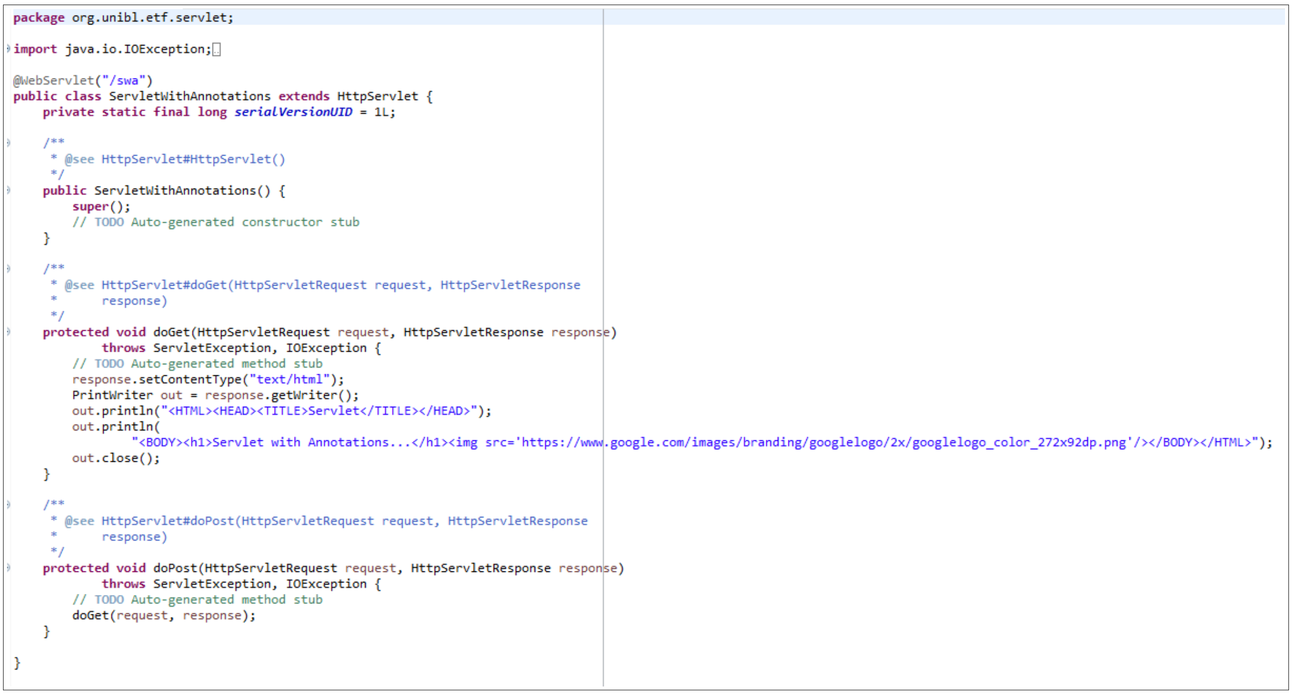


```
package org.unibl.etf.servlet;

import java.io.IOException;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Redirect() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String redirectUrl = request.getParameter("redirect");
        System.out.println(redirectUrl);
        response.sendRedirect(redirectUrl);
    }

    /**
     * @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);
    }

}
```

ServletWithAnnotations.java

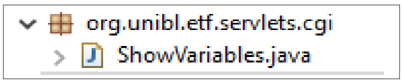


Servlet with Annotations...

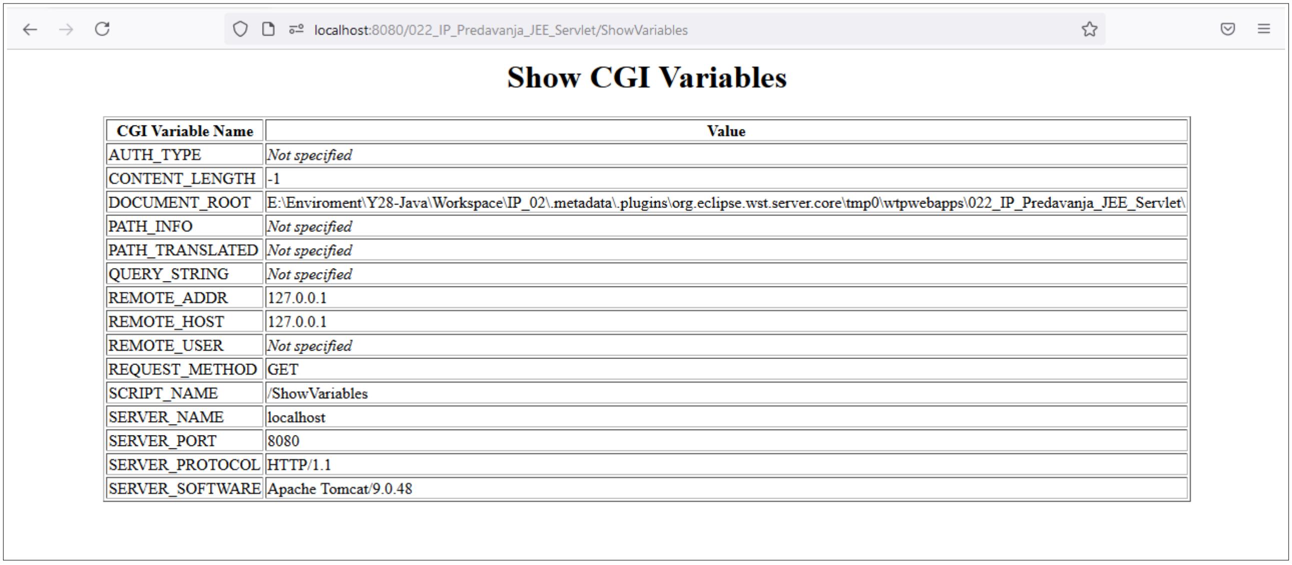


```java
package org.unibl.etf.servlet;

import java.io.IOException;□

@WebServlet("/swa")
public class ServletWithAnnotations extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ServletWithAnnotations() {
        super();
        // 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.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<HTML><HEAD><TITLE>Servlet</TITLE></HEAD>");
        out.println(
                "<BODY><h1>Servlet with Annotations...</h1><img src='https://www.google.com/images/branding/googlelogo/2x/googlelogo_color_272x92dp.png'/></BODY></HTML>");
        out.close();
    }

    /**
     * @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);
    }

}
```

servlet-cgi

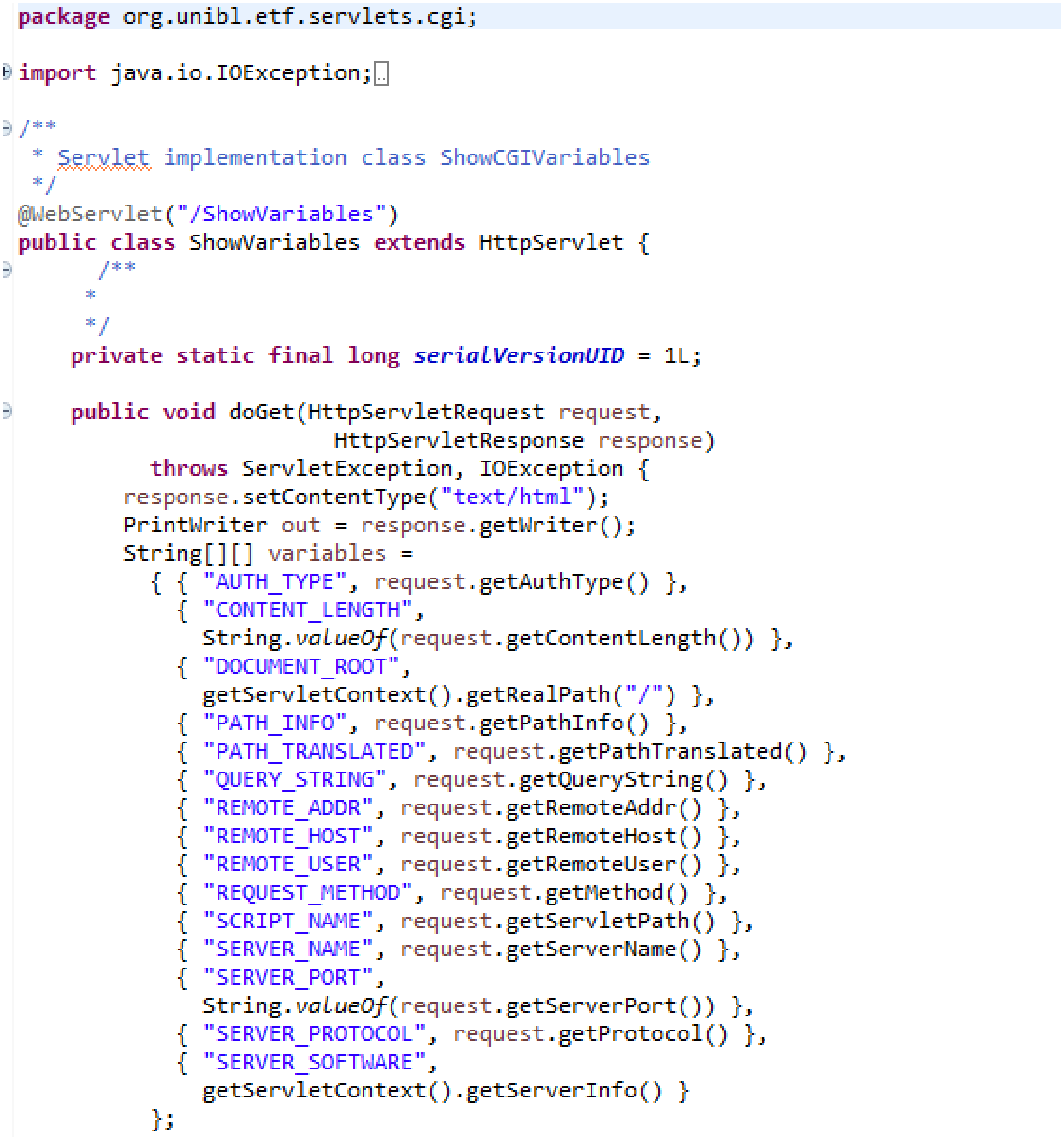


ServletVariables.java

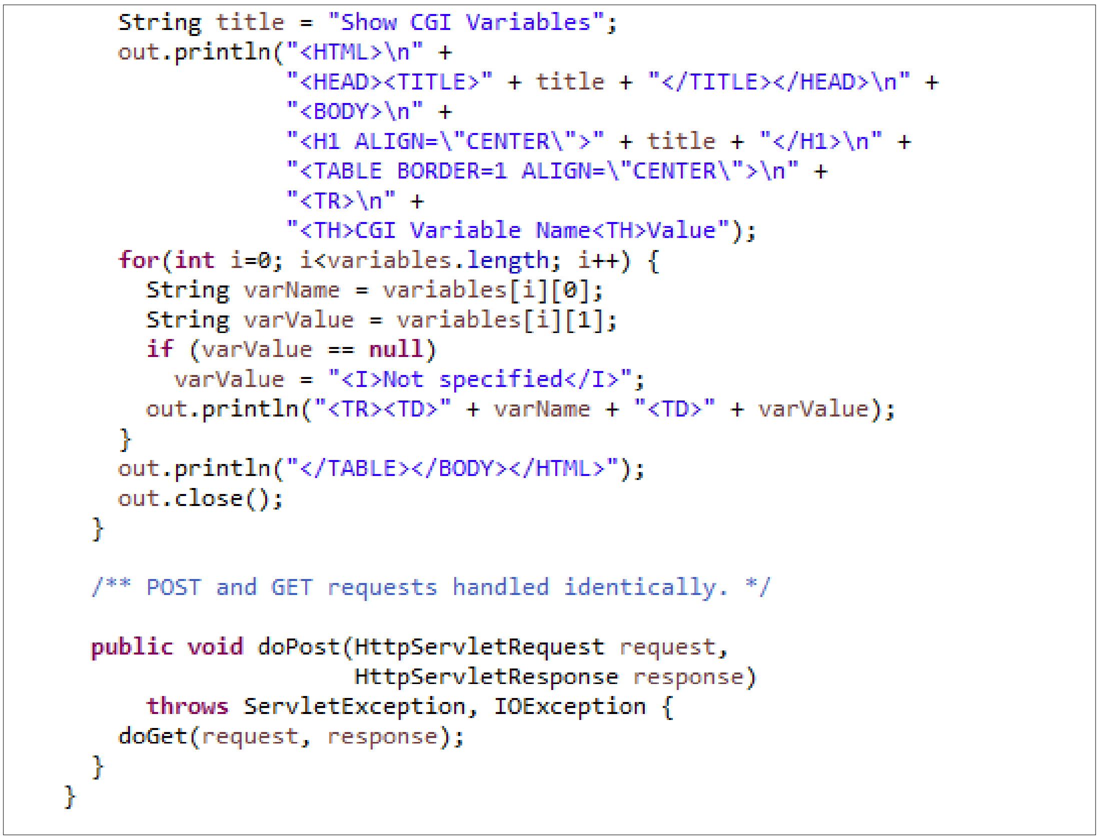


Show CGI Variables

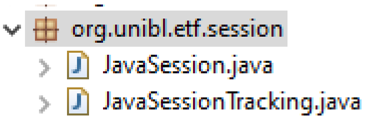
CGI Variable Name	Value
AUTH_TYPE	*Not specified*
CONTENT_LENGTH	-1
DOCUMENT_ROOT	E:\Enviroment\Y28-Java\Workspace\IP_02\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\022_IP_Predavanja_JEE_Servlet\
PATH_INFO	*Not specified*
PATH_TRANSLATED	*Not specified*
QUERY_STRING	*Not specified*
REMOTE_ADDR	127.0.0.1
REMOTE_HOST	127.0.0.1
REMOTE_USER	*Not specified*
REQUEST_METHOD	GET
SCRIPT_NAME	/ShowVariables
SERVER_NAME	localhost
SERVER_PORT	8080
SERVER_PROTOCOL	HTTP/1.1
SERVER_SOFTWARE	Apache Tomcat/9.0.48

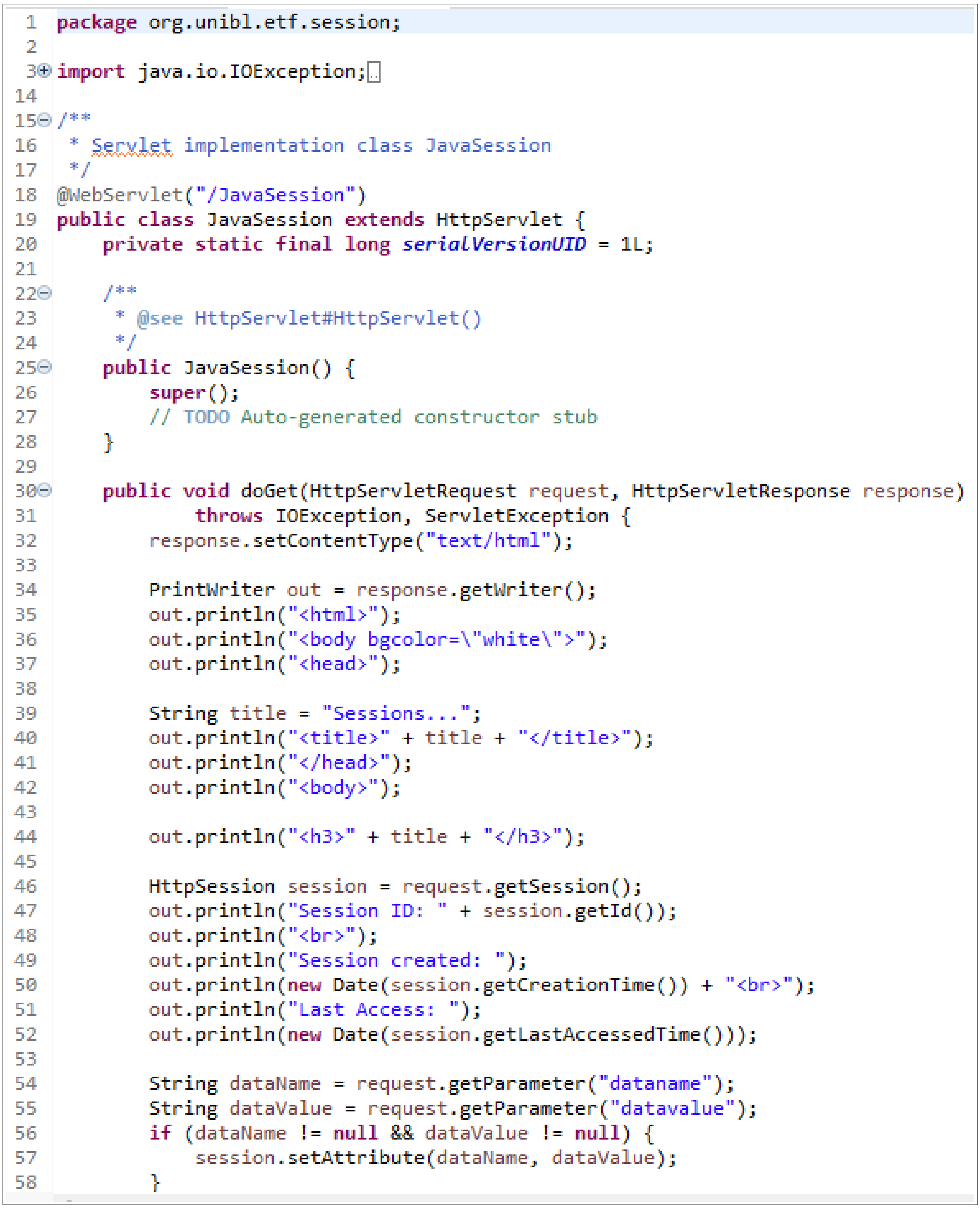


```
package org.unibl.etf.servlets.cgi;

import java.io.IOException;

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

    public void doGet(HttpServletRequest request,
                        HttpServletResponse response)
          throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String[][] variables =
          { { "AUTH_TYPE", request.getAuthType() },
            { "CONTENT_LENGTH",
              String.valueOf(request.getContentLength()) },
            { "DOCUMENT_ROOT",
              getServletContext().getRealPath("/") },
            { "PATH_INFO", request.getPathInfo() },
            { "PATH_TRANSLATED", request.getPathTranslated() },
            { "QUERY_STRING", request.getQueryString() },
            { "REMOTE_ADDR", request.getRemoteAddr() },
            { "REMOTE_HOST", request.getRemoteHost() },
            { "REMOTE_USER", request.getRemoteUser() },
            { "REQUEST_METHOD", request.getMethod() },
            { "SCRIPT_NAME", request.getServletPath() },
            { "SERVER_NAME", request.getServerName() },
            { "SERVER_PORT",
              String.valueOf(request.getServerPort()) },
            { "SERVER_PROTOCOL", request.getProtocol() },
            { "SERVER_SOFTWARE",
              getServletContext().getServerInfo() }
          };
```

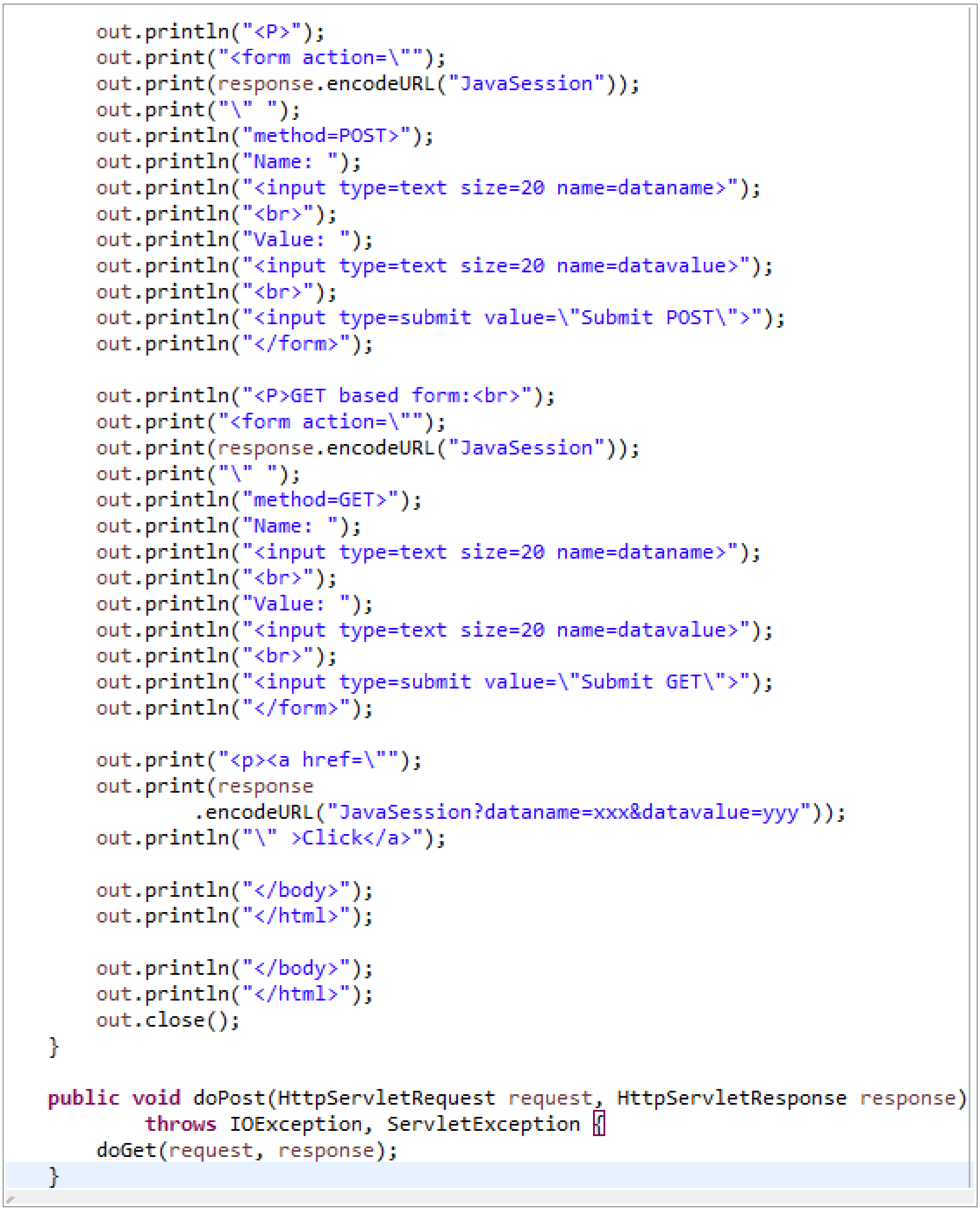


```
    String title = "Show CGI Variables";
    out.println("<HTML>\n" +
                "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
                "<BODY>\n" +
                "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n" +
                "<TABLE BORDER=1 ALIGN=\"CENTER\">\n" +
                "<TR>\n" +
                "<TH>CGI Variable Name<TH>Value");
    for(int i=0; i<variables.length; i++) {
      String varName = variables[i][0];
      String varValue = variables[i][1];
      if (varValue == null)
        varValue = "<I>Not specified</I>";
      out.println("<TR><TD>" + varName + "<TD>" + varValue);
    }
    out.println("</TABLE></BODY></HTML>");
    out.close();
  }

  /** POST and GET requests handled identically. */

  public void doPost(HttpServletRequest request,
                     HttpServletResponse response)
      throws ServletException, IOException {
    doGet(request, response);
  }
}
```

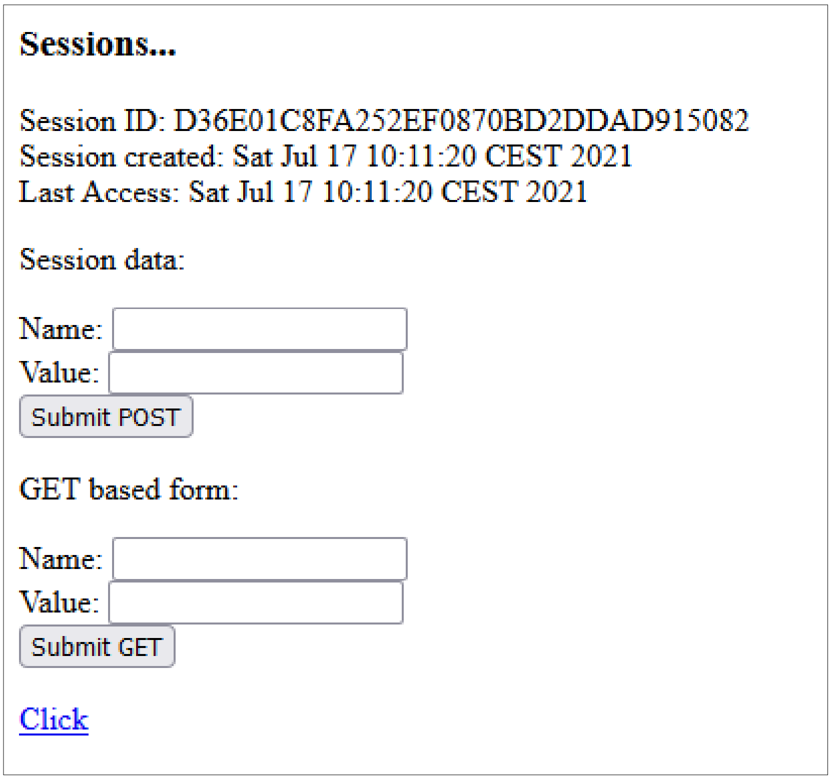




```java
package org.unibl.etf.session;

import java.io.IOException;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public JavaSession() {
        super();
        // TODO Auto-generated constructor stub
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws IOException, ServletException {
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body bgcolor=\"white\">");
        out.println("<head>");

        String title = "Sessions...";
        out.println("<title>" + title + "</title>");
        out.println("</head>");
        out.println("<body>");

        out.println("<h3>" + title + "</h3>");

        HttpSession session = request.getSession();
        out.println("Session ID: " + session.getId());
        out.println("<br>");
        out.println("Session created: ");
        out.println(new Date(session.getCreationTime()) + "<br>");
        out.println("Last Access: ");
        out.println(new Date(session.getLastAccessedTime()));

        String dataName = request.getParameter("dataname");
        String dataValue = request.getParameter("datavalue");
        if (dataName != null && dataValue != null) {
            session.setAttribute(dataName, dataValue);
        }
```



```java
        out.println("<P>");
        out.print("<form action=\"");
        out.print(response.encodeURL("JavaSession"));
        out.print("\" ");
        out.println("method=POST>");
        out.println("Name: ");
        out.println("<input type=text size=20 name=dataname>");
        out.println("<br>");
        out.println("Value: ");
        out.println("<input type=text size=20 name=datavalue>");
        out.println("<br>");
        out.println("<input type=submit value=\"Submit POST\">");
        out.println("</form>");

        out.println("<P>GET based form:<br>");
        out.print("<form action=\"");
        out.print(response.encodeURL("JavaSession"));
        out.print("\" ");
        out.println("method=GET>");
        out.println("Name: ");
        out.println("<input type=text size=20 name=dataname>");
        out.println("<br>");
        out.println("Value: ");
        out.println("<input type=text size=20 name=datavalue>");
        out.println("<br>");
        out.println("<input type=submit value=\"Submit GET\">");
        out.println("</form>");

        out.print("<p><a href=\"");
        out.print(response
                .encodeURL("JavaSession?dataname=xxx&datavalue=yyy"));
        out.println("\" >Click</a>");

        out.println("</body>");
        out.println("</html>");

        out.println("</body>");
        out.println("</html>");
        out.close();
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
            throws IOException, ServletException {
        doGet(request, response);
    }
```



Sessions...

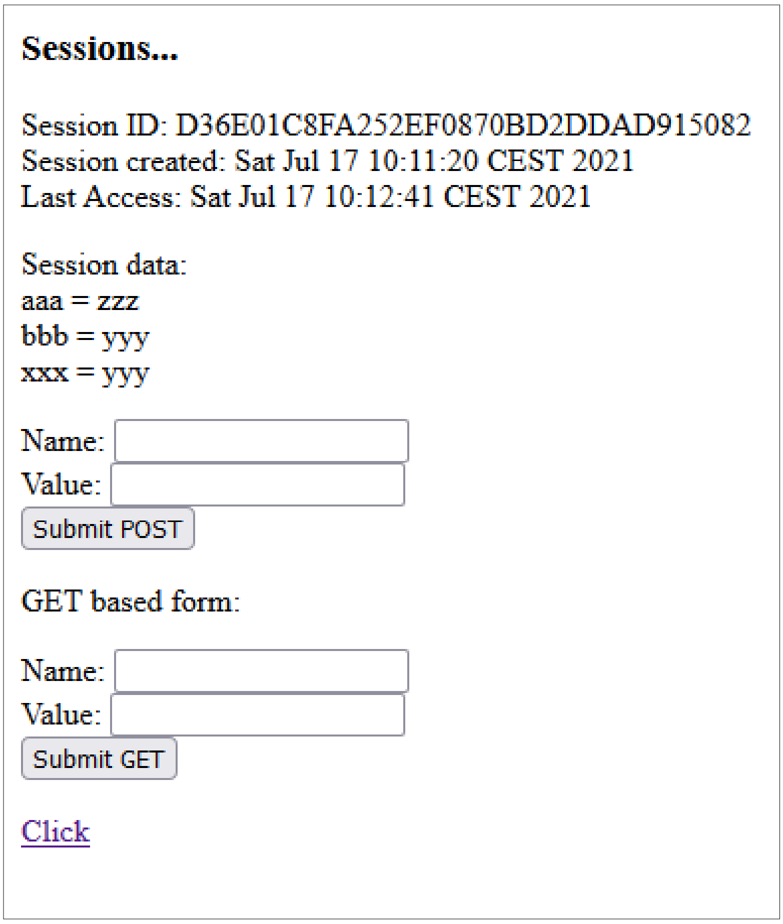
Session ID: D36E01C8FA252EF0870BD2DDAD915082
Session created: Sat Jul 17 10:11:20 CEST 2021
Last Access: Sat Jul 17 10:11:20 CEST 2021

Session data:

Name:
Value:

GET based form:

Name:
Value:



Sessions...

Session ID: D36E01C8FA252EF0870BD2DDAD915082
Session created: Sat Jul 17 10:11:20 CEST 2021
Last Access: Sat Jul 17 10:12:41 CEST 2021

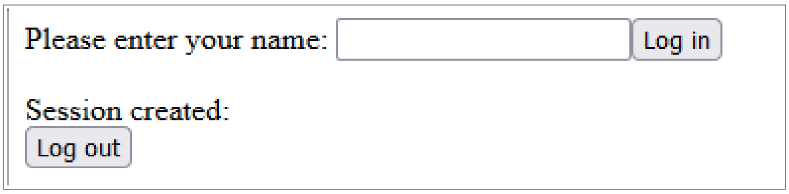
Session data:
aaa = zzz
bbb = yyy
xxx = yyy

Name:
Value:

GET based form:

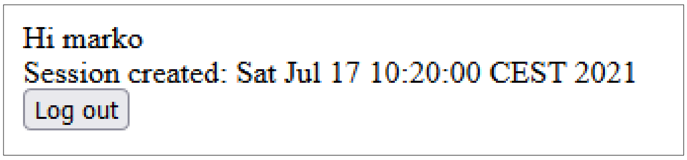
Name:
Value:

JavaSessionTracking.java

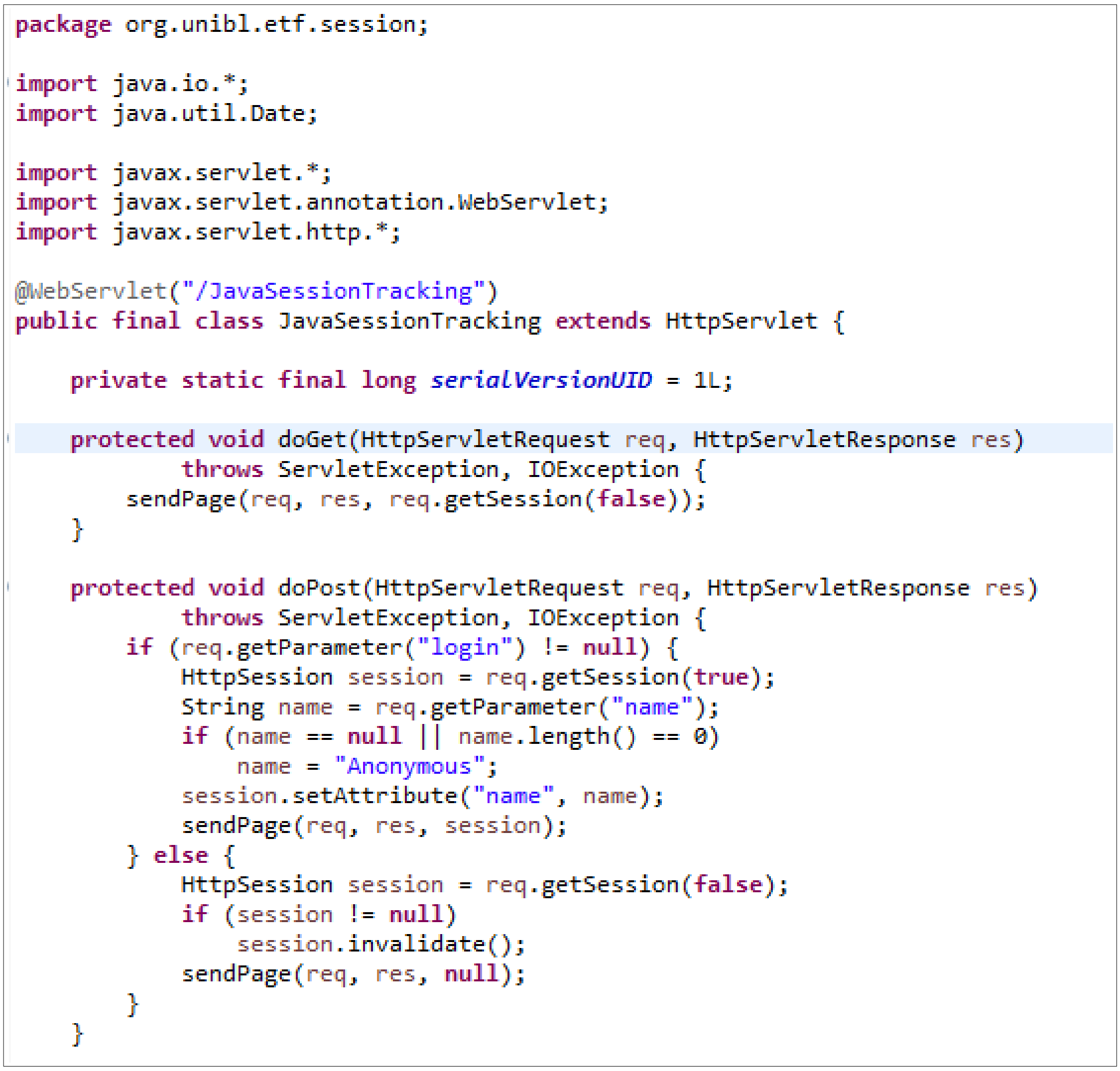


Please enter your name:

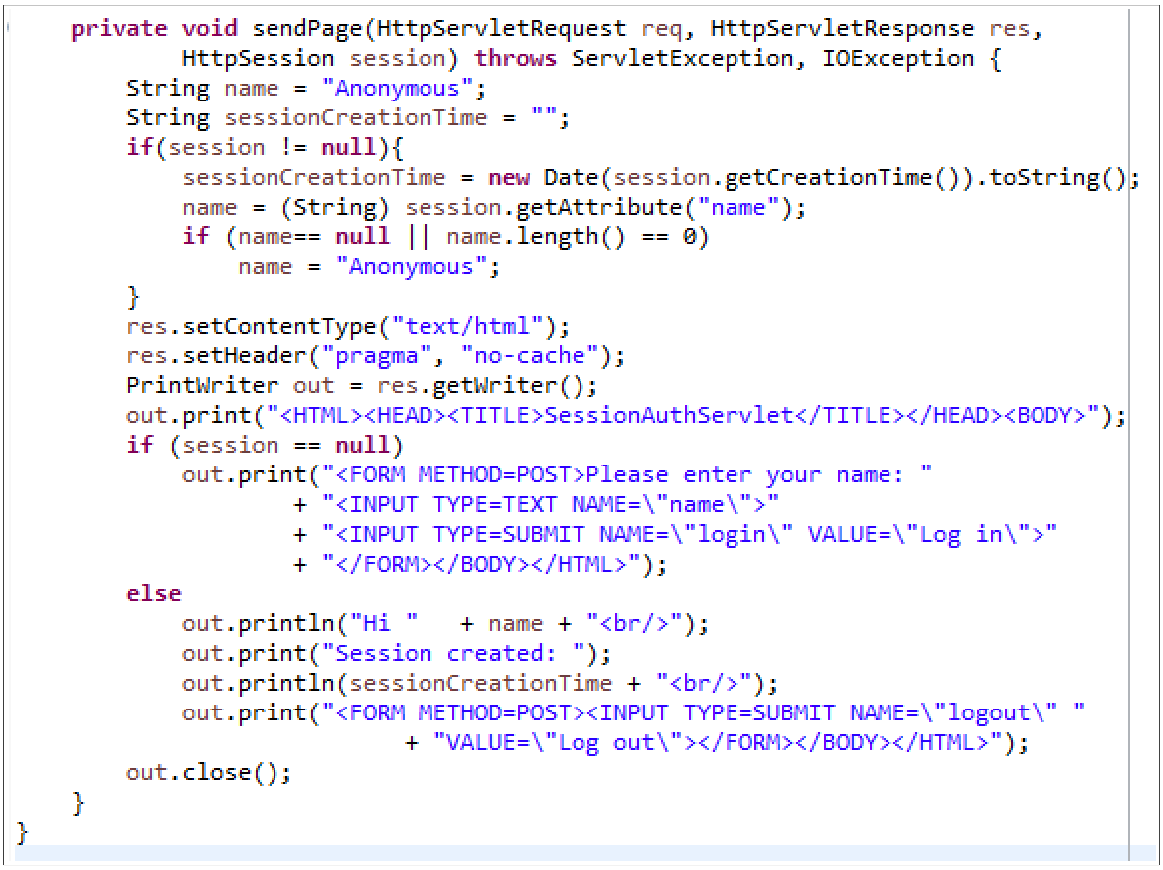
Session created:



Hi marko
Session created: Sat Jul 17 10:20:00 CEST 2021

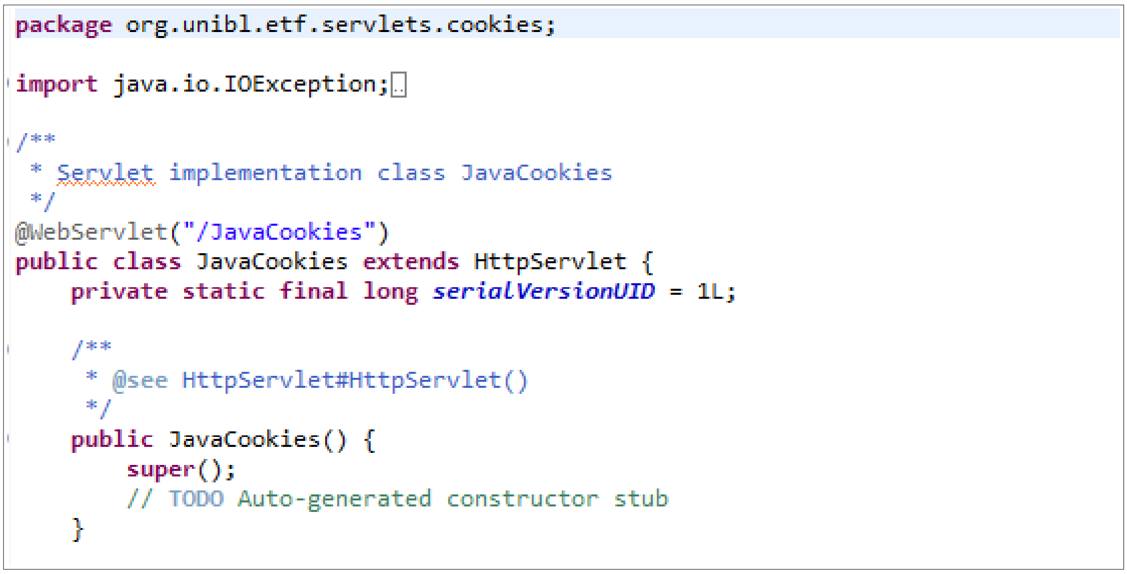


```java
package org.unibl.etf.session;

import java.io.*;
import java.util.Date;

import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

@WebServlet("/JavaSessionTracking")
public final class JavaSessionTracking extends HttpServlet {

    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest req, HttpServletResponse res)
            throws ServletException, IOException {
        sendPage(req, res, req.getSession(false));
    }

    protected void doPost(HttpServletRequest req, HttpServletResponse res)
            throws ServletException, IOException {
        if (req.getParameter("login") != null) {
            HttpSession session = req.getSession(true);
            String name = req.getParameter("name");
            if (name == null || name.length() == 0)
                name = "Anonymous";
            session.setAttribute("name", name);
            sendPage(req, res, session);
        } else {
            HttpSession session = req.getSession(false);
            if (session != null)
                session.invalidate();
            sendPage(req, res, null);
        }
    }
```

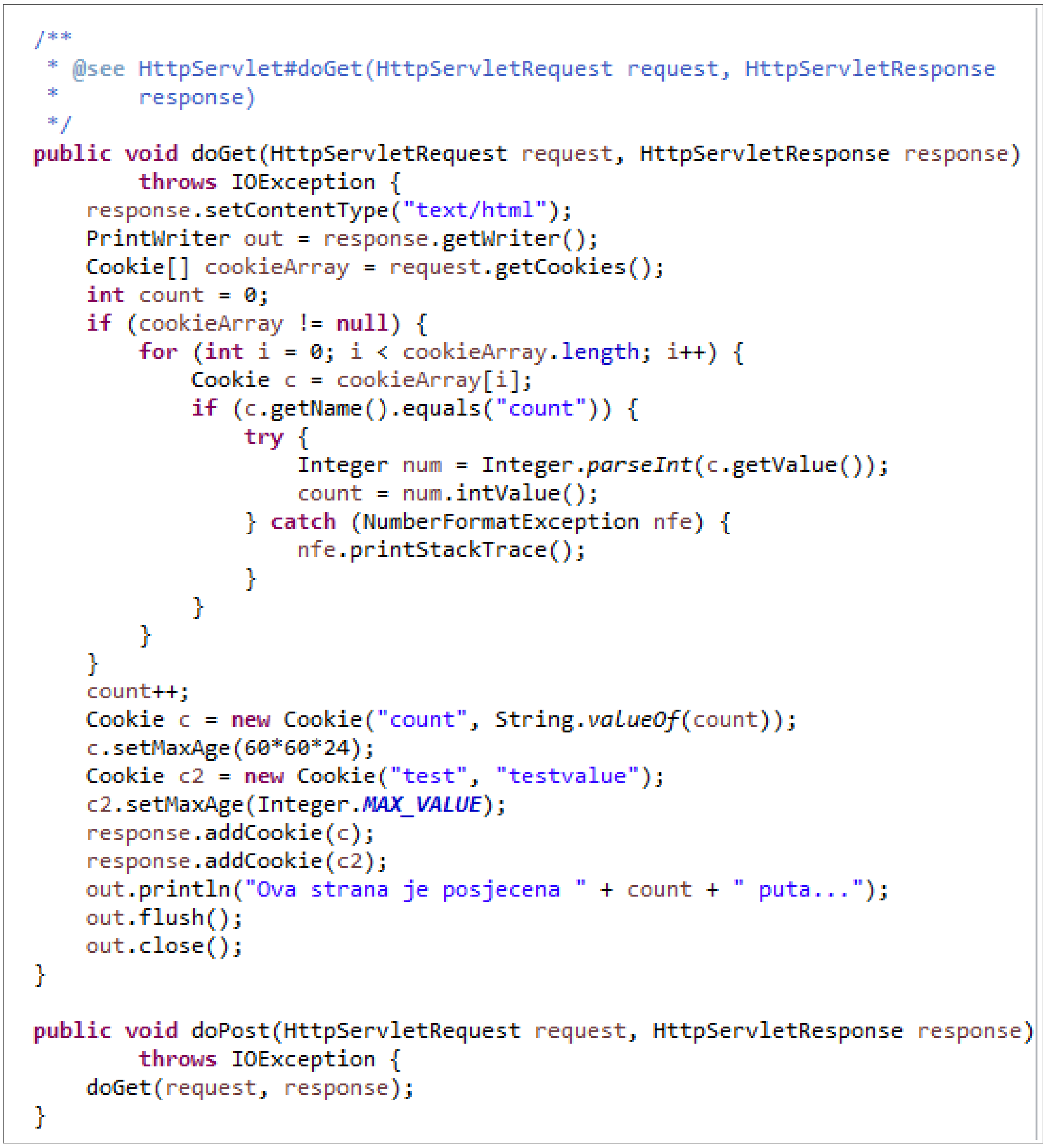


```java
    private void sendPage(HttpServletRequest req, HttpServletResponse res,
            HttpSession session) throws ServletException, IOException {
        String name = "Anonymous";
        String sessionCreationTime = "";
        if(session != null){
            sessionCreationTime = new Date(session.getCreationTime()).toString();
            name = (String) session.getAttribute("name");
            if (name== null || name.length() == 0)
                name = "Anonymous";
        }
        res.setContentType("text/html");
        res.setHeader("pragma", "no-cache");
        PrintWriter out = res.getWriter();
        out.print("<HTML><HEAD><TITLE>SessionAuthServlet</TITLE></HEAD><BODY>");
        if (session == null)
            out.print("<FORM METHOD=POST>Please enter your name: "
                    + "<INPUT TYPE=TEXT NAME=\"name\">"
                    + "<INPUT TYPE=SUBMIT NAME=\"login\" VALUE=\"Log in\">"
                    + "</FORM></BODY></HTML>");
        else
            out.println("Hi "   + name + "<br/>");
            out.print("Session created: ");
            out.println(sessionCreationTime + "<br/>");
            out.print("<FORM METHOD=POST><INPUT TYPE=SUBMIT NAME=\"logout\" "
                            + "VALUE=\"Log out\"></FORM></BODY></HTML>");
        out.close();
    }
}
```

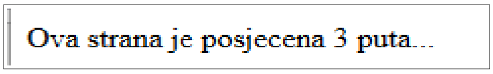
JavaCookies.java



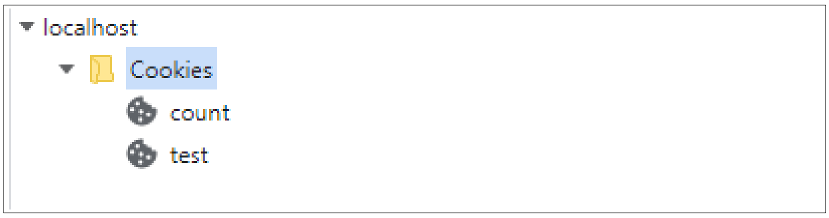
```
package org.unibl.etf.servlets.cookies;

import java.io.IOException;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public JavaCookies() {
        super();
        // TODO Auto-generated constructor stub
    }
```



```
  /**
   * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
   *      response)
   */
  public void doGet(HttpServletRequest request, HttpServletResponse response)
          throws IOException {
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      Cookie[] cookieArray = request.getCookies();
      int count = 0;
      if (cookieArray != null) {
          for (int i = 0; i < cookieArray.length; i++) {
              Cookie c = cookieArray[i];
              if (c.getName().equals("count")) {
                  try {
                      Integer num = Integer.parseInt(c.getValue());
                      count = num.intValue();
                  } catch (NumberFormatException nfe) {
                      nfe.printStackTrace();
                  }
              }
          }
      }
      count++;
      Cookie c = new Cookie("count", String.valueOf(count));
      c.setMaxAge(60*60*24);
      Cookie c2 = new Cookie("test", "testvalue");
      c2.setMaxAge(Integer.MAX_VALUE);
      response.addCookie(c);
      response.addCookie(c2);
      out.println("Ova strana je posjecena " + count + " puta...");
      out.flush();
      out.close();
  }

  public void doPost(HttpServletRequest request, HttpServletResponse response)
          throws IOException {
      doGet(request, response);
  }
```

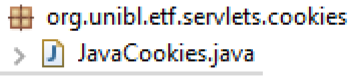


Ova strana je posjecena 3 puta...



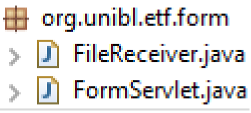
```
▾ localhost
    ▾ Cookies
          count
          test
```

cookies – servlet

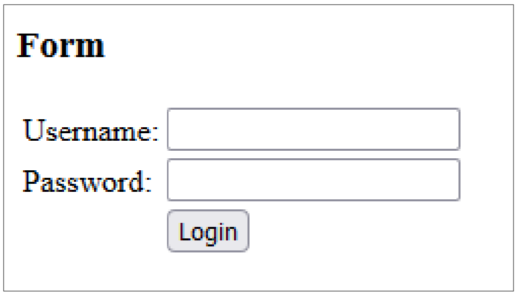


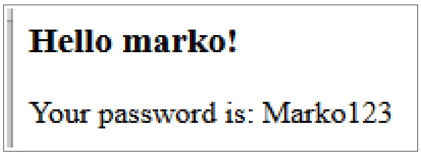
```
org.unibl.etf.servlets.cookies
 > JavaCookies.java
```

form – servlet

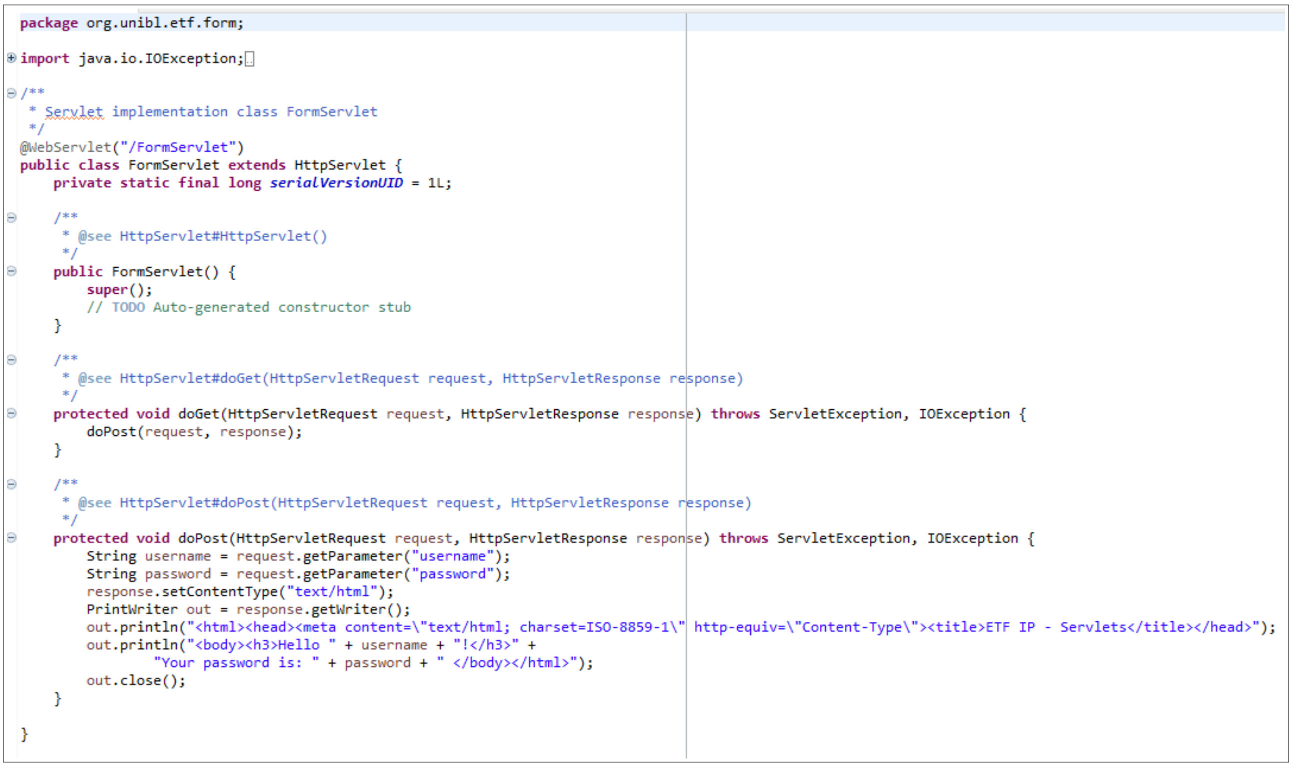


FormsServlet.java

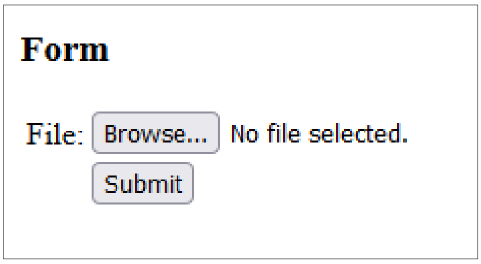

Form

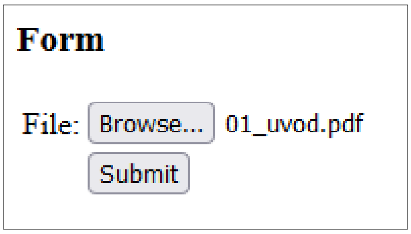

Hello marko!

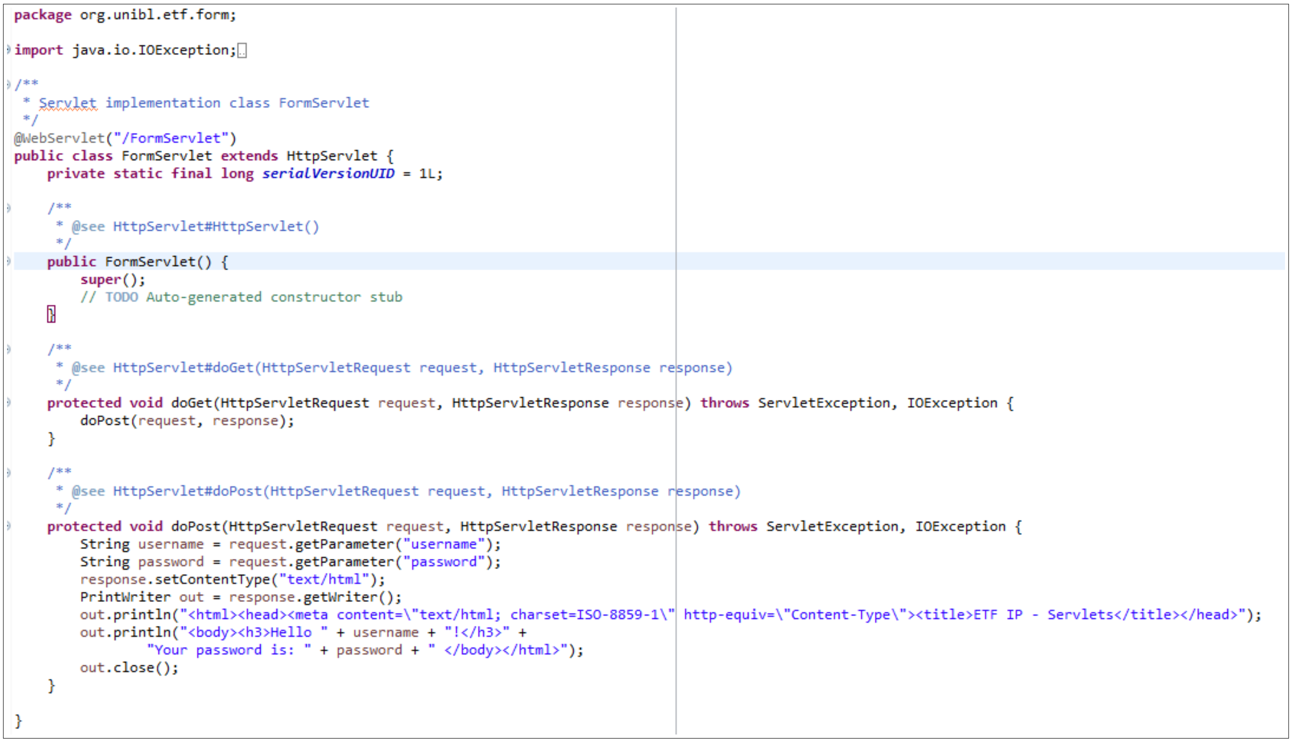
Your password is: Marko123


```
package org.unibl.etf.form;

import java.io.IOException;□

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public FormServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

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

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><head><meta content=\"text/html; charset=ISO-8859-1\" http-equiv=\"Content-Type\"><title>ETF IP - Servlets</title></head>");
        out.println("<body><h3>Hello " + username + "!</h3>" +
                "Your password is: " + password + " </body></html>");
        out.close();
    }

}
```

FileReceiver.java

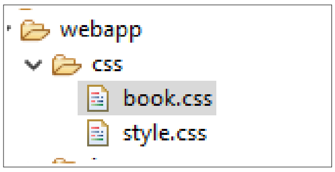






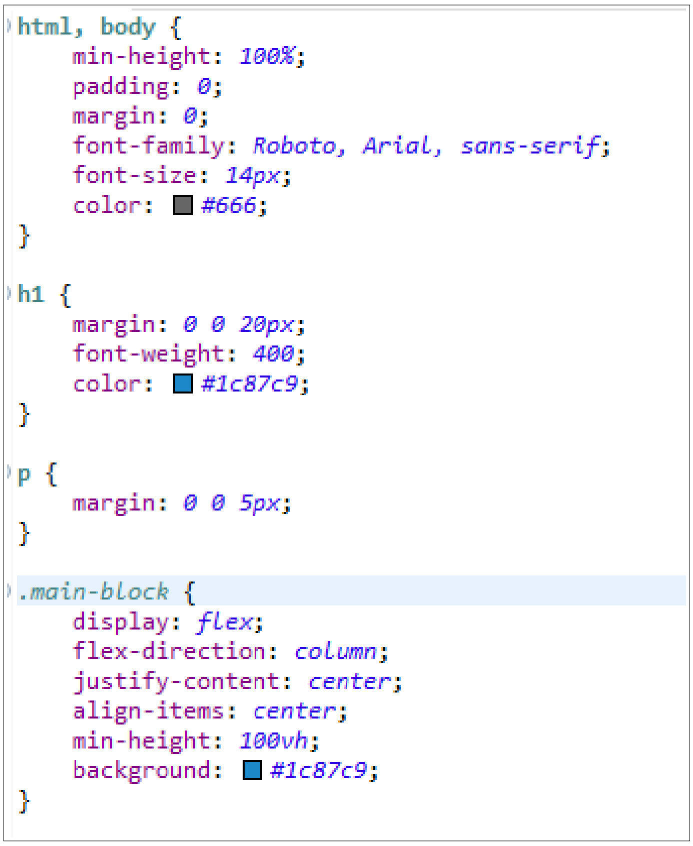
```java
package org.unibl.etf.form;

import java.io.IOException;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public FormServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

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

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><head><meta content=\"text/html; charset=ISO-8859-1\" http-equiv=\"Content-Type\"><title>ETF IP - Servlets</title></head>");
        out.println("<body><h3>Hello " + username + "!</h3>" +
                "Your password is: " + password + " </body></html>");
        out.close();
    }

}
```

web content – css

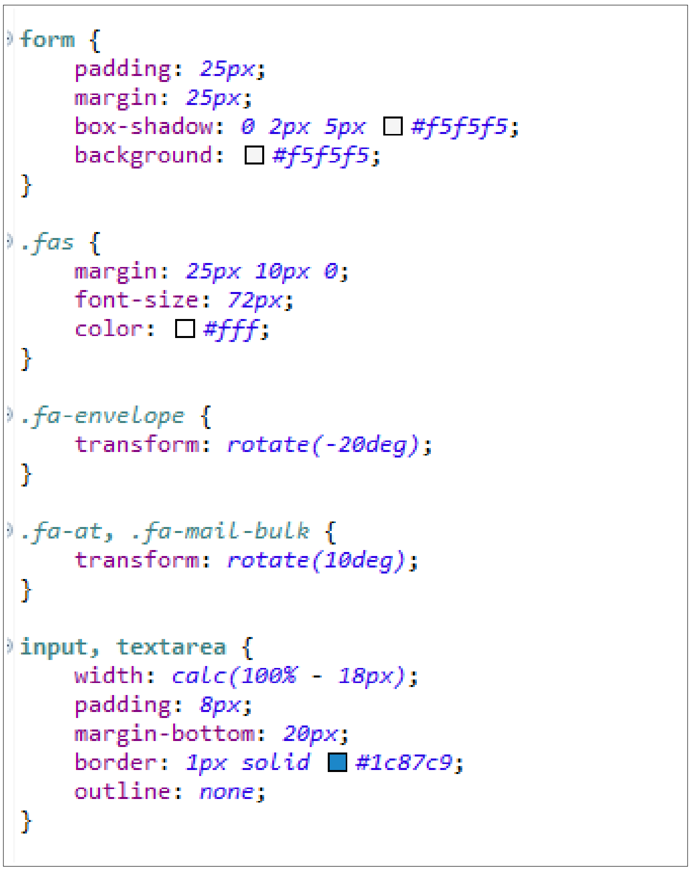


```
webapp
  css
    book.css
    style.css
```

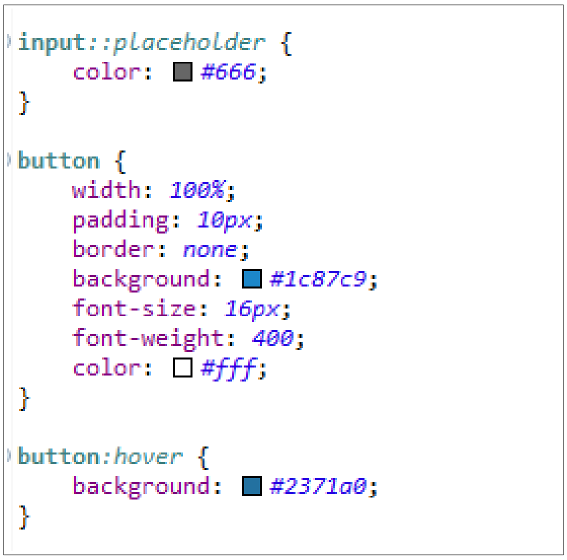
book.css



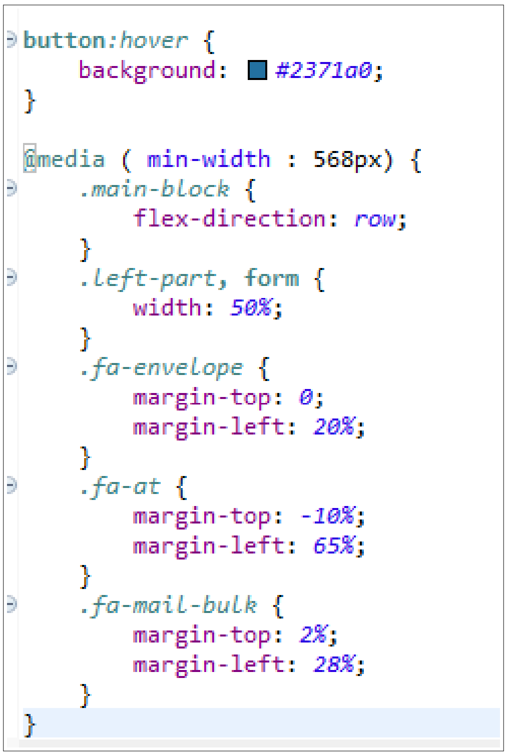
```
html, body {
    min-height: 100%;
    padding: 0;
    margin: 0;
    font-family: Roboto, Arial, sans-serif;
    font-size: 14px;
    color: #666;
}

h1 {
    margin: 0 0 20px;
    font-weight: 400;
    color: #1c87c9;
}

p {
    margin: 0 0 5px;
}

.main-block {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background: #1c87c9;
}
```



```
form {
    padding: 25px;
    margin: 25px;
    box-shadow: 0 2px 5px #f5f5f5;
    background: #f5f5f5;
}

.fas {
    margin: 25px 10px 0;
    font-size: 72px;
    color: #fff;
}

.fa-envelope {
    transform: rotate(-20deg);
}

.fa-at, .fa-mail-bulk {
    transform: rotate(10deg);
}

input, textarea {
    width: calc(100% - 18px);
    padding: 8px;
    margin-bottom: 20px;
    border: 1px solid #1c87c9;
    outline: none;
}
```

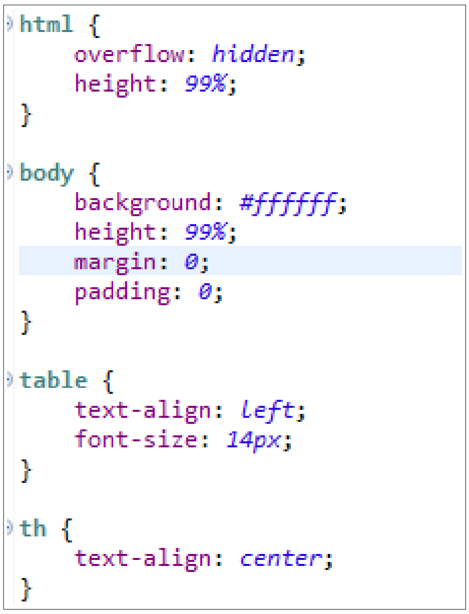


```css
input::placeholder {
    color: #666;
}

button {
    width: 100%;
    padding: 10px;
    border: none;
    background: #1c87c9;
    font-size: 16px;
    font-weight: 400;
    color: #fff;
}

button:hover {
    background: #2371a0;
}
```



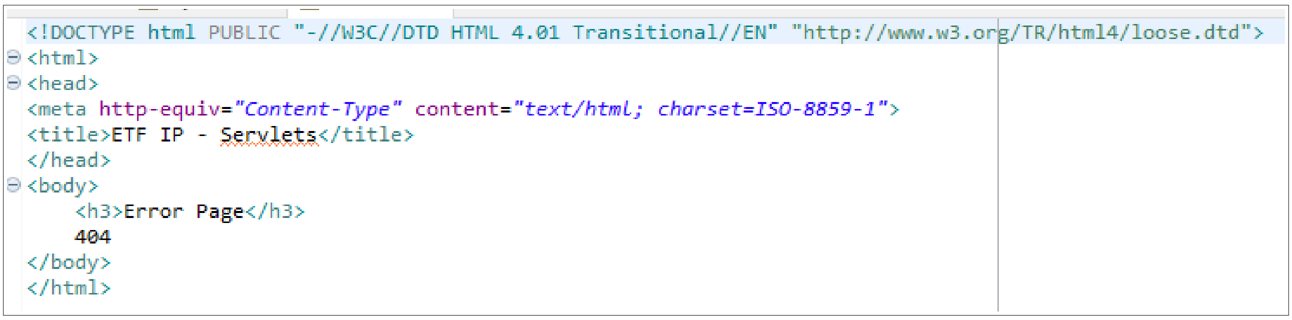
```css
button:hover {
    background: #2371a0;
}

@media ( min-width : 568px) {
    .main-block {
        flex-direction: row;
    }
    .left-part, form {
        width: 50%;
    }
    .fa-envelope {
        margin-top: 0;
        margin-left: 20%;
    }
    .fa-at {
        margin-top: -10%;
        margin-left: 65%;
    }
    .fa-mail-bulk {
        margin-top: 2%;
        margin-left: 28%;
    }
}
```

style.css



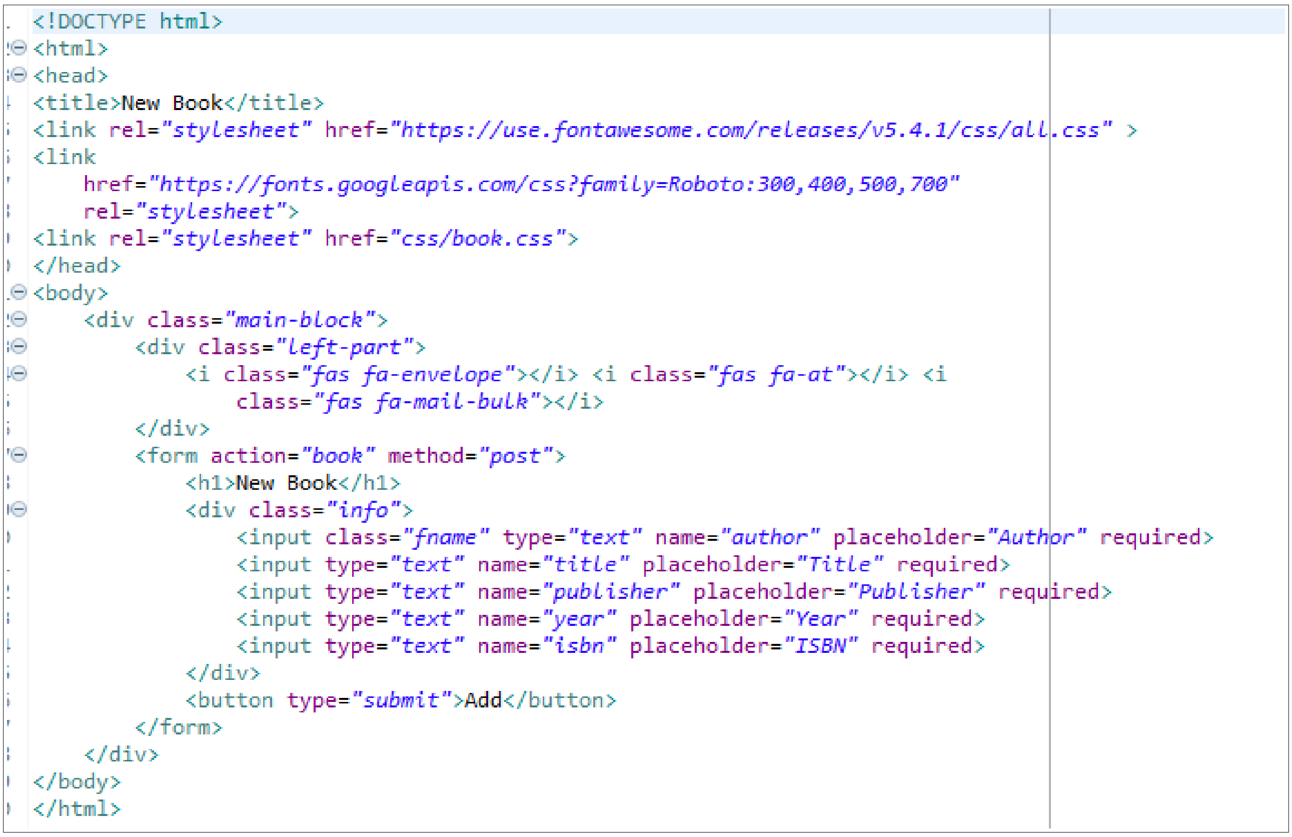
```
html {
    overflow: hidden;
    height: 99%;
}

body {
    background: #ffffff;
    height: 99%;
    margin: 0;
    padding: 0;
}

table {
    text-align: left;
    font-size: 14px;
}

th {
    text-align: center;
}
```

404.html



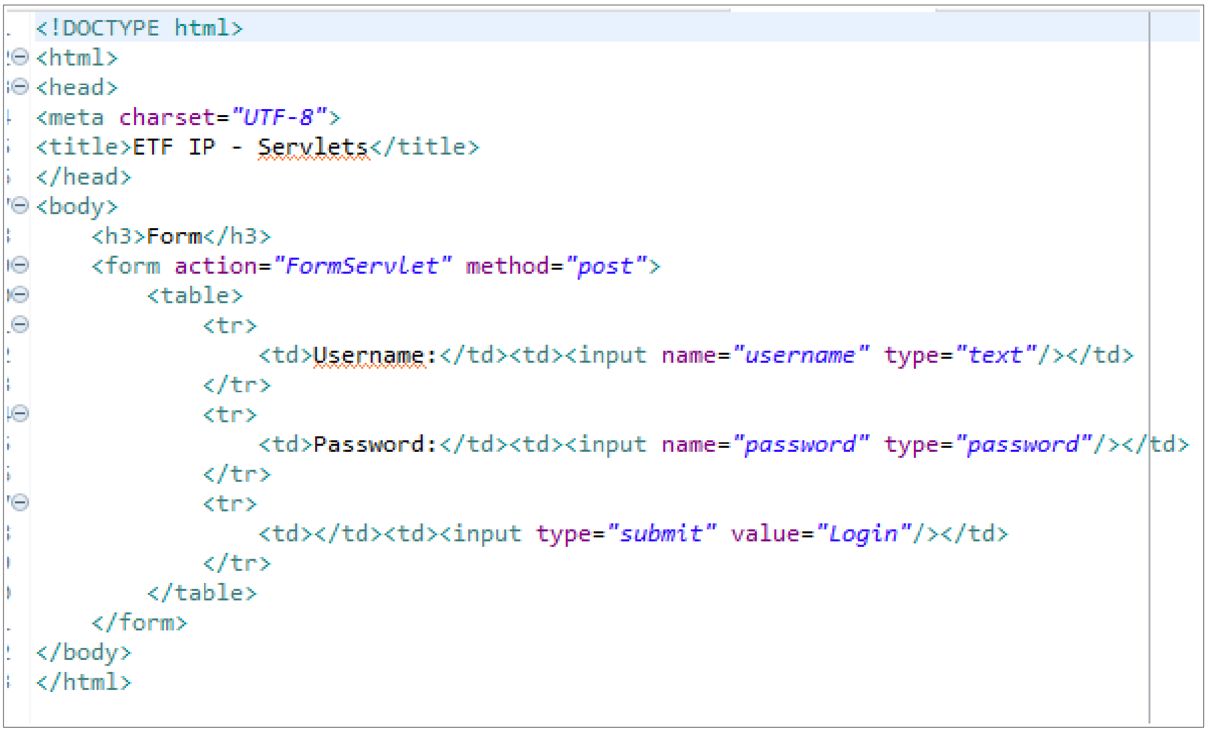
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>ETF IP - Servlets</title>
</head>
<body>
    <h3>Error Page</h3>
    404
</body>
</html>
```

book.html



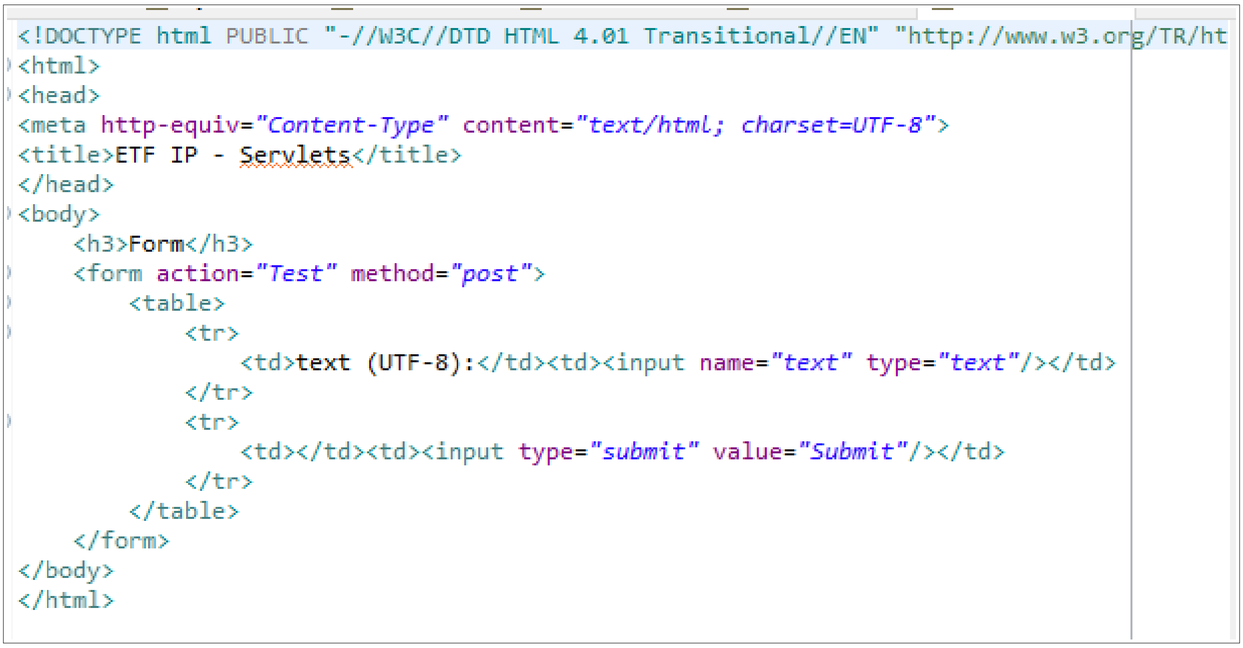
```
<!DOCTYPE html>
<html>
<head>
<title>New Book</title>
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.4.1/css/all.css" >
<link
    href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700"
    rel="stylesheet">
<link rel="stylesheet" href="css/book.css">
</head>
<body>
    <div class="main-block">
        <div class="left-part">
            <i class="fas fa-envelope"></i> <i class="fas fa-at"></i> <i
                class="fas fa-mail-bulk"></i>
        </div>
        <form action="book" method="post">
            <h1>New Book</h1>
            <div class="info">
                <input class="fname" type="text" name="author" placeholder="Author" required>
                <input type="text" name="title" placeholder="Title" required>
                <input type="text" name="publisher" placeholder="Publisher" required>
                <input type="text" name="year" placeholder="Year" required>
                <input type="text" name="isbn" placeholder="ISBN" required>
            </div>
            <button type="submit">Add</button>
        </form>
    </div>
</body>
</html>
```

form.html



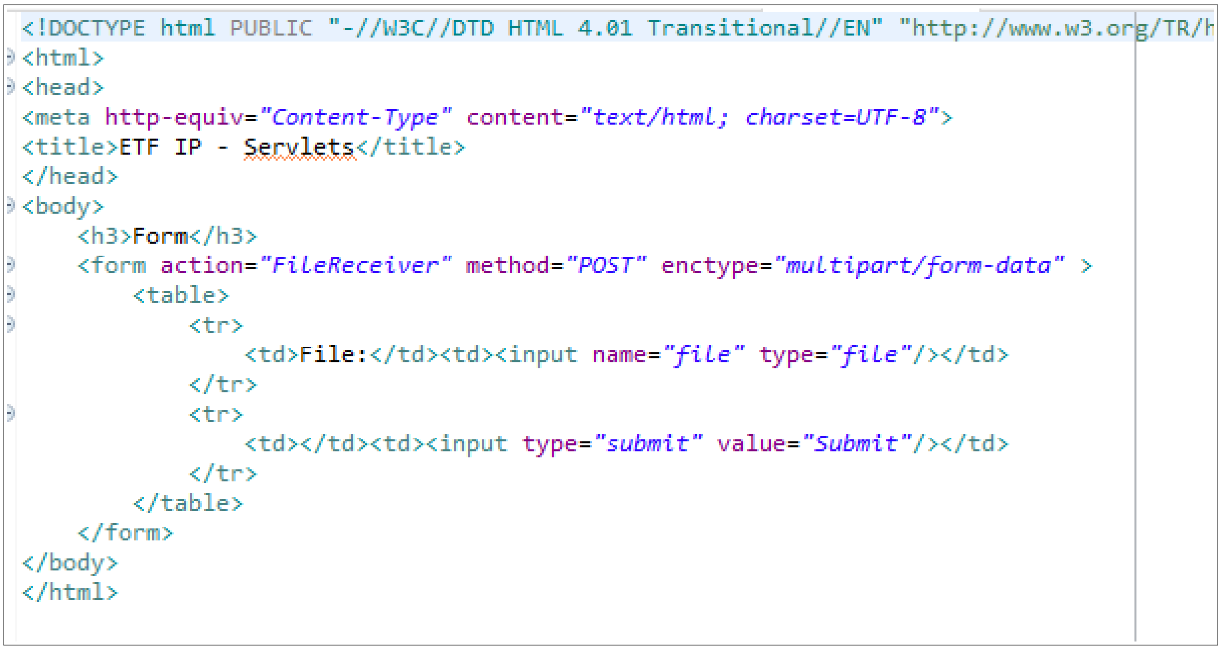
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>ETF IP - Servlets</title>
</head>
<body>
    <h3>Form</h3>
    <form action="FormServlet" method="post">
        <table>
            <tr>
                <td>Username:</td><td><input name="username" type="text"/></td>
            </tr>
            <tr>
                <td>Password:</td><td><input name="password" type="password"/></td>
            </tr>
            <tr>
                <td></td><td><input type="submit" value="Login"/></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

form2.html



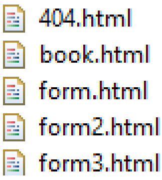
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/ht
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>ETF IP - Servlets</title>
</head>
<body>
    <h3>Form</h3>
    <form action="Test" method="post">
        <table>
            <tr>
                <td>text (UTF-8):</td><td><input name="text" type="text"/></td>
            </tr>
            <tr>
                <td></td><td><input type="submit" value="Submit"/></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

form3.html



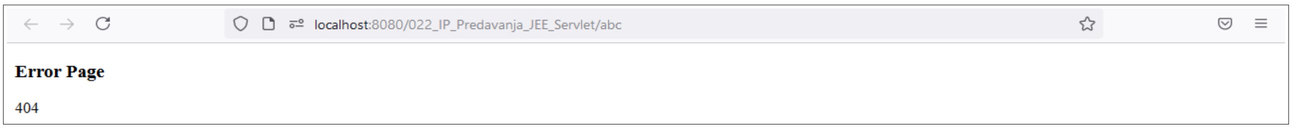
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/h
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>ETF IP - Servlets</title>
</head>
<body>
    <h3>Form</h3>
    <form action="FileReceiver" method="POST" enctype="multipart/form-data" >
        <table>
            <tr>
                <td>File:</td><td><input name="file" type="file"/></td>
            </tr>
            <tr>
                <td></td><td><input type="submit" value="Submit"/></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

web content -html



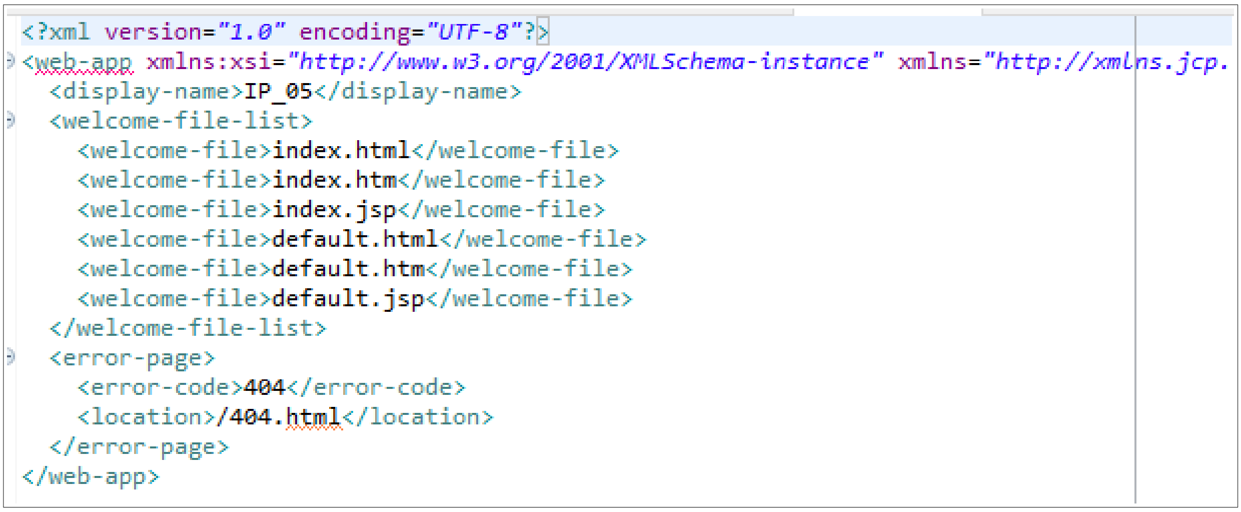
404.html
book.html
form.html
form2.html
form3.html

error-page



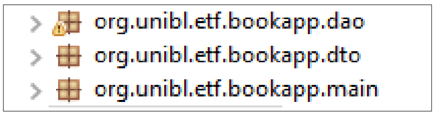
Error Page

404



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.
  <display-name>IP_05</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <error-page>
    <error-code>404</error-code>
    <location>/404.html</location>
  </error-page>
</web-app>
```

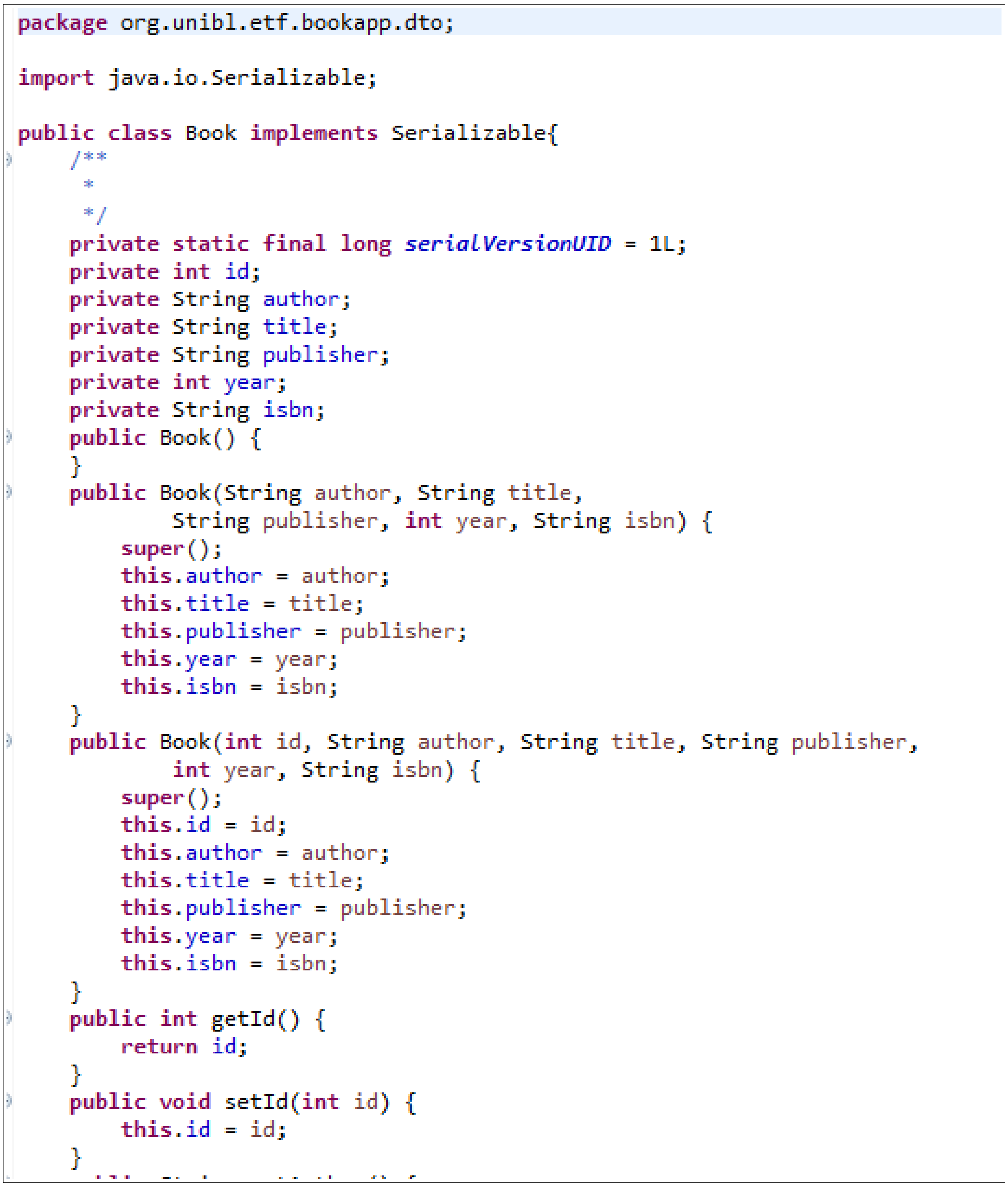
BOOK API



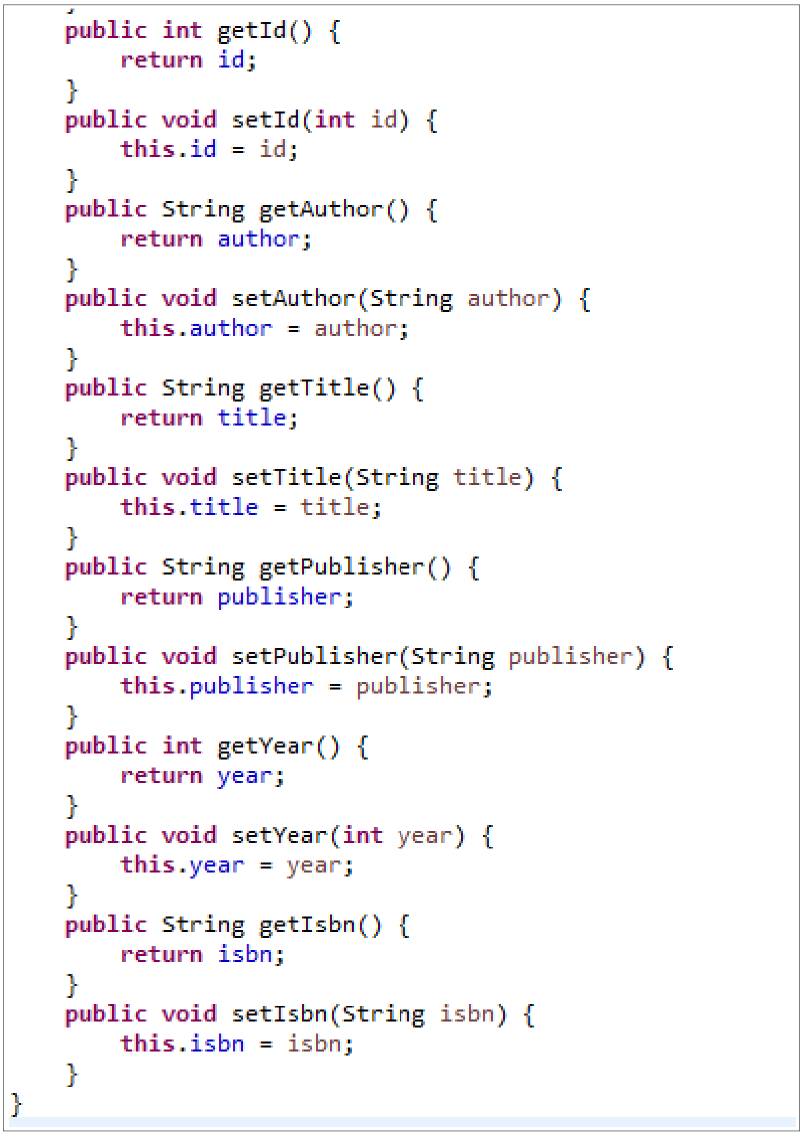
org.unibl.etf.bookapp.dao
org.unibl.etf.bookapp.dto
org.unibl.etf.bookapp.main

bookapp – dto

BookDTO.java

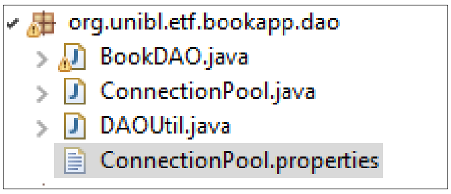


```java
package org.unibl.etf.bookapp.dto;

import java.io.Serializable;

public class Book implements Serializable{
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private int id;
    private String author;
    private String title;
    private String publisher;
    private int year;
    private String isbn;
    public Book() {
    }
    public Book(String author, String title,
            String publisher, int year, String isbn) {
        super();
        this.author = author;
        this.title = title;
        this.publisher = publisher;
        this.year = year;
        this.isbn = isbn;
    }
    public Book(int id, String author, String title, String publisher,
            int year, String isbn) {
        super();
        this.id = id;
        this.author = author;
        this.title = title;
        this.publisher = publisher;
        this.year = year;
        this.isbn = isbn;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
```



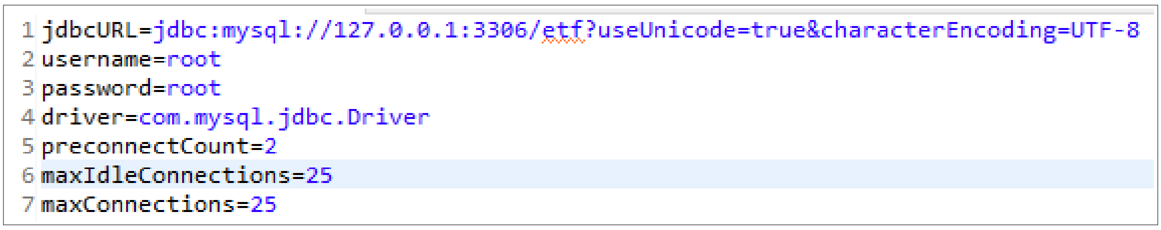
```java
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public int getYear() {
        return year;
    }
    public void setYear(int year) {
        this.year = year;
    }
    public String getIsbn() {
        return isbn;
    }
    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }
}
```

book-dao

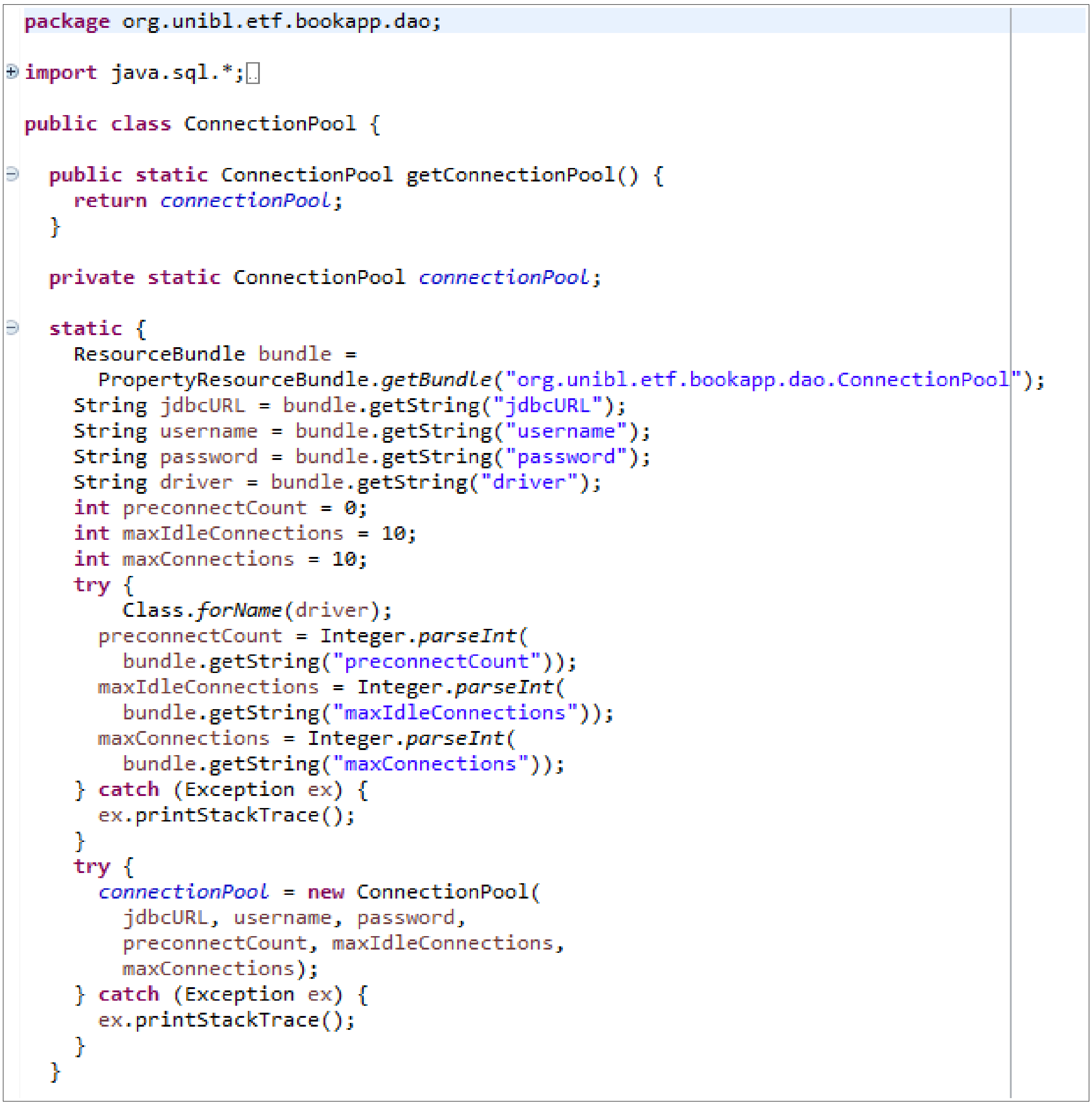


```
org.unibl.etf.bookapp.dao
  > BookDAO.java
  > ConnectionPool.java
  > DAOUtil.java
    ConnectionPool.properties
```

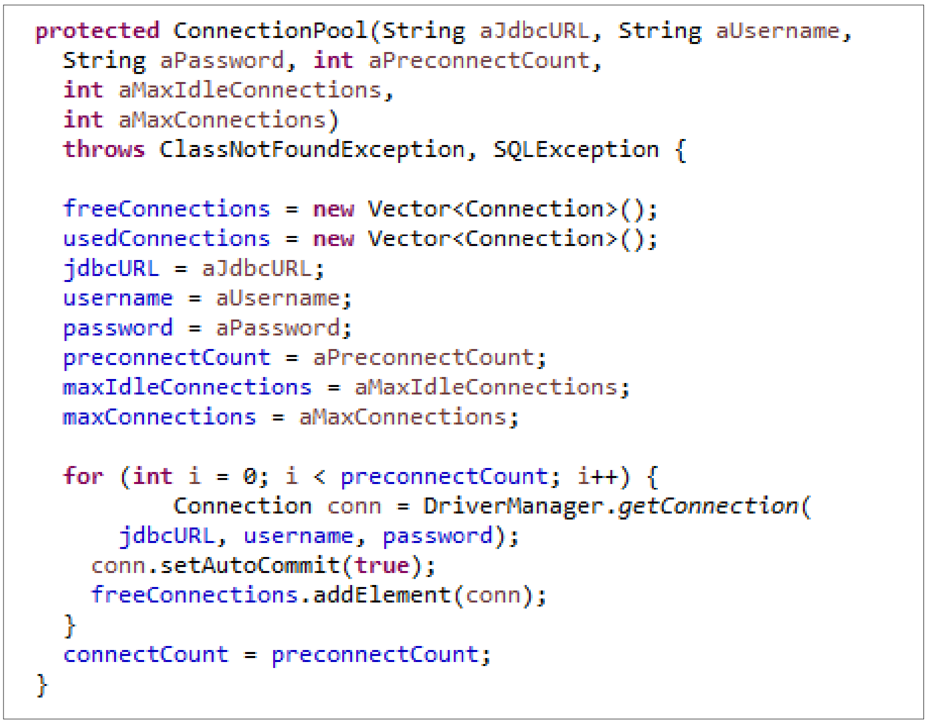
ConnectionPool.properties



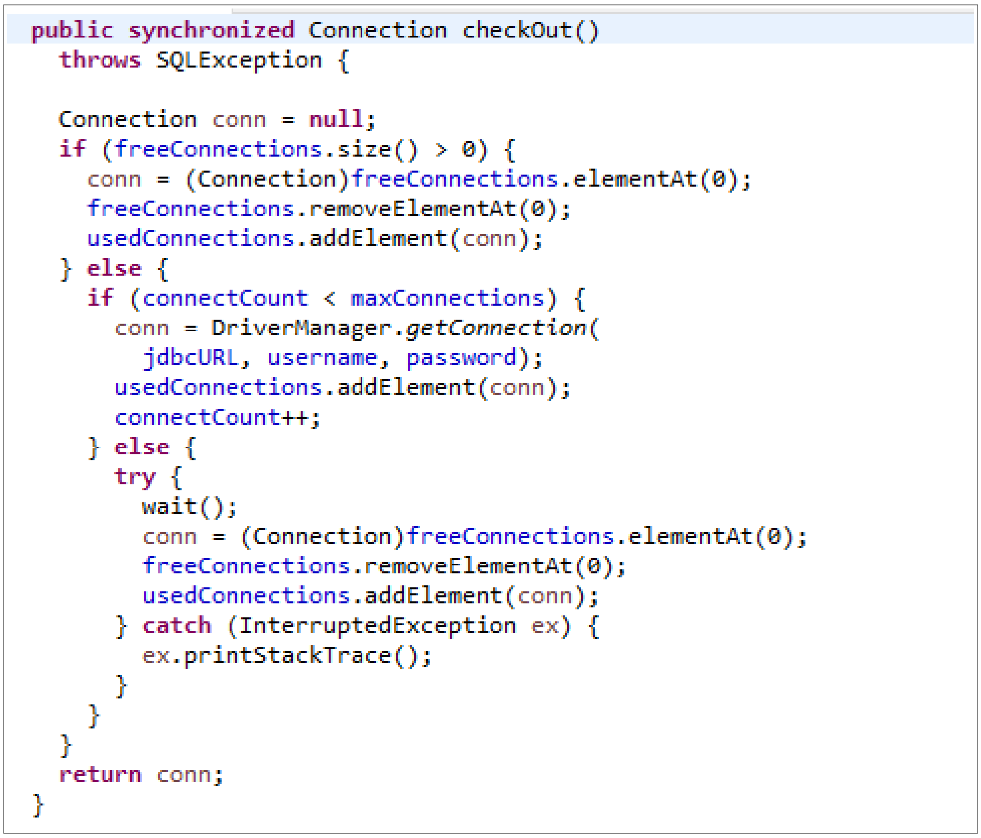
```properties
jdbcURL=jdbc:mysql://127.0.0.1:3306/etf?useUnicode=true&characterEncoding=UTF-8
username=root
password=root
driver=com.mysql.jdbc.Driver
preconnectCount=2
maxIdleConnections=25
maxConnections=25
```



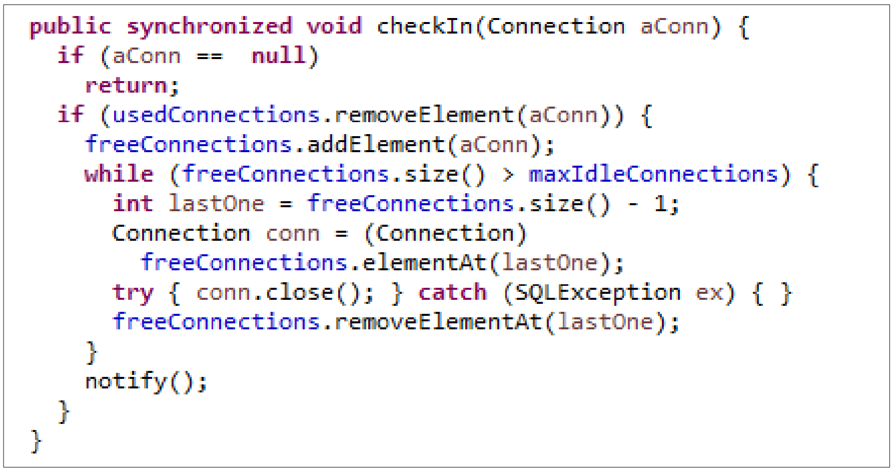
```java
package org.unibl.etf.bookapp.dao;

import java.sql.*;

public class ConnectionPool {

  public static ConnectionPool getConnectionPool() {
    return connectionPool;
  }

  private static ConnectionPool connectionPool;

  static {
    ResourceBundle bundle =
      PropertyResourceBundle.getBundle("org.unibl.etf.bookapp.dao.ConnectionPool");
    String jdbcURL = bundle.getString("jdbcURL");
    String username = bundle.getString("username");
    String password = bundle.getString("password");
    String driver = bundle.getString("driver");
    int preconnectCount = 0;
    int maxIdleConnections = 10;
    int maxConnections = 10;
    try {
        Class.forName(driver);
      preconnectCount = Integer.parseInt(
        bundle.getString("preconnectCount"));
      maxIdleConnections = Integer.parseInt(
        bundle.getString("maxIdleConnections"));
      maxConnections = Integer.parseInt(
        bundle.getString("maxConnections"));
    } catch (Exception ex) {
      ex.printStackTrace();
    }
    try {
      connectionPool = new ConnectionPool(
        jdbcURL, username, password,
        preconnectCount, maxIdleConnections,
        maxConnections);
    } catch (Exception ex) {
      ex.printStackTrace();
    }
  }
```



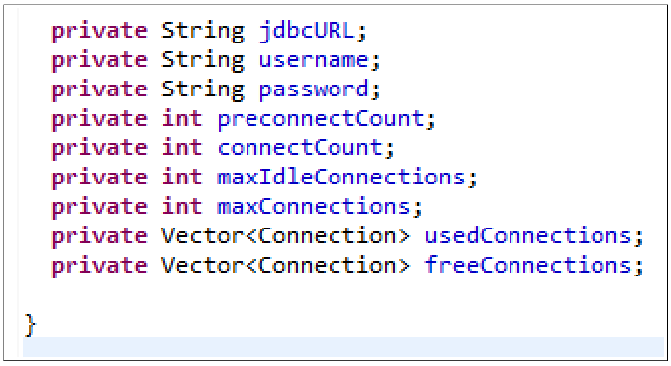
```java
protected ConnectionPool(String aJdbcURL, String aUsername,
  String aPassword, int aPreconnectCount,
  int aMaxIdleConnections,
  int aMaxConnections)
  throws ClassNotFoundException, SQLException {

  freeConnections = new Vector<Connection>();
  usedConnections = new Vector<Connection>();
  jdbcURL = aJdbcURL;
  username = aUsername;
  password = aPassword;
  preconnectCount = aPreconnectCount;
  maxIdleConnections = aMaxIdleConnections;
  maxConnections = aMaxConnections;

  for (int i = 0; i < preconnectCount; i++) {
          Connection conn = DriverManager.getConnection(
      jdbcURL, username, password);
    conn.setAutoCommit(true);
    freeConnections.addElement(conn);
  }
  connectCount = preconnectCount;
}
```



```
public synchronized Connection checkOut()
  throws SQLException {

  Connection conn = null;
  if (freeConnections.size() > 0) {
    conn = (Connection)freeConnections.elementAt(0);
    freeConnections.removeElementAt(0);
    usedConnections.addElement(conn);
  } else {
    if (connectCount < maxConnections) {
      conn = DriverManager.getConnection(
        jdbcURL, username, password);
      usedConnections.addElement(conn);
      connectCount++;
    } else {
      try {
        wait();
        conn = (Connection)freeConnections.elementAt(0);
        freeConnections.removeElementAt(0);
        usedConnections.addElement(conn);
      } catch (InterruptedException ex) {
        ex.printStackTrace();
      }
    }
  }
  return conn;
}
```

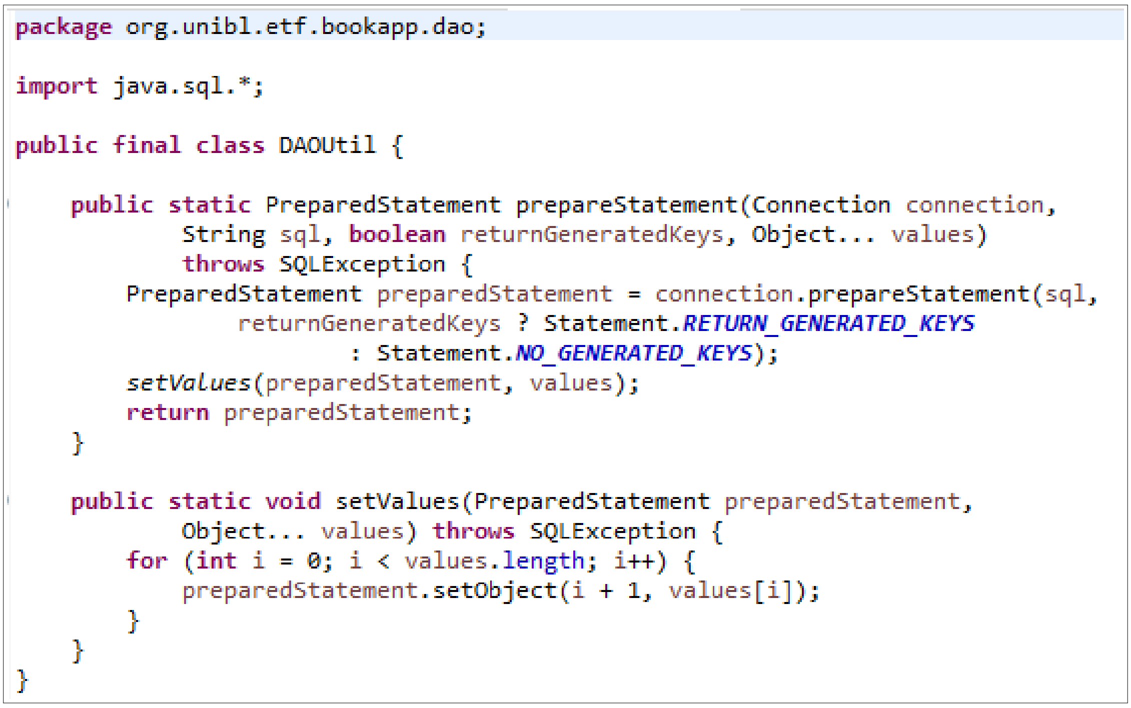


```
public synchronized void checkIn(Connection aConn) {
  if (aConn ==  null)
    return;
  if (usedConnections.removeElement(aConn)) {
    freeConnections.addElement(aConn);
    while (freeConnections.size() > maxIdleConnections) {
      int lastOne = freeConnections.size() - 1;
      Connection conn = (Connection)
        freeConnections.elementAt(lastOne);
      try { conn.close(); } catch (SQLException ex) { }
      freeConnections.removeElementAt(lastOne);
    }
    notify();
  }
}
```



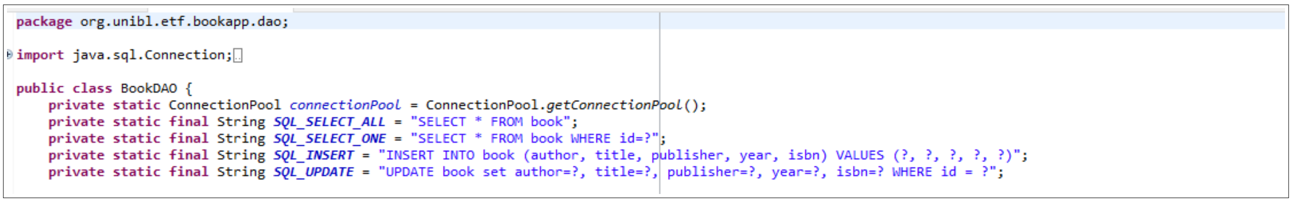
```
  private String jdbcURL;
  private String username;
  private String password;
  private int preconnectCount;
  private int connectCount;
  private int maxIdleConnections;
  private int maxConnections;
  private Vector<Connection> usedConnections;
  private Vector<Connection> freeConnections;

}
```

DAOUtil.java

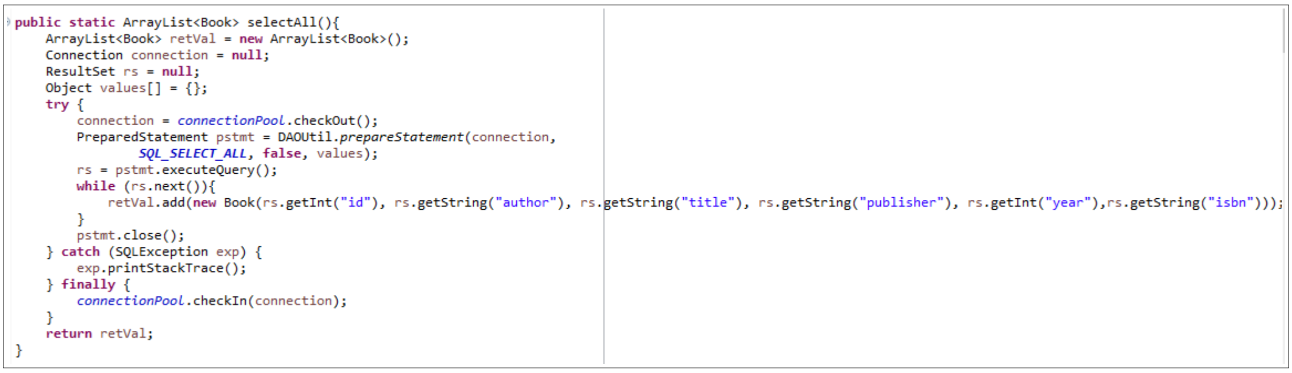


```java
package org.unibl.etf.bookapp.dao;

import java.sql.*;

public final class DAOUtil {

    public static PreparedStatement prepareStatement(Connection connection,
            String sql, boolean returnGeneratedKeys, Object... values)
            throws SQLException {
        PreparedStatement preparedStatement = connection.prepareStatement(sql,
                returnGeneratedKeys ? Statement.RETURN_GENERATED_KEYS
                        : Statement.NO_GENERATED_KEYS);
        setValues(preparedStatement, values);
        return preparedStatement;
    }

    public static void setValues(PreparedStatement preparedStatement,
            Object... values) throws SQLException {
        for (int i = 0; i < values.length; i++) {
            preparedStatement.setObject(i + 1, values[i]);
        }
    }
}
```

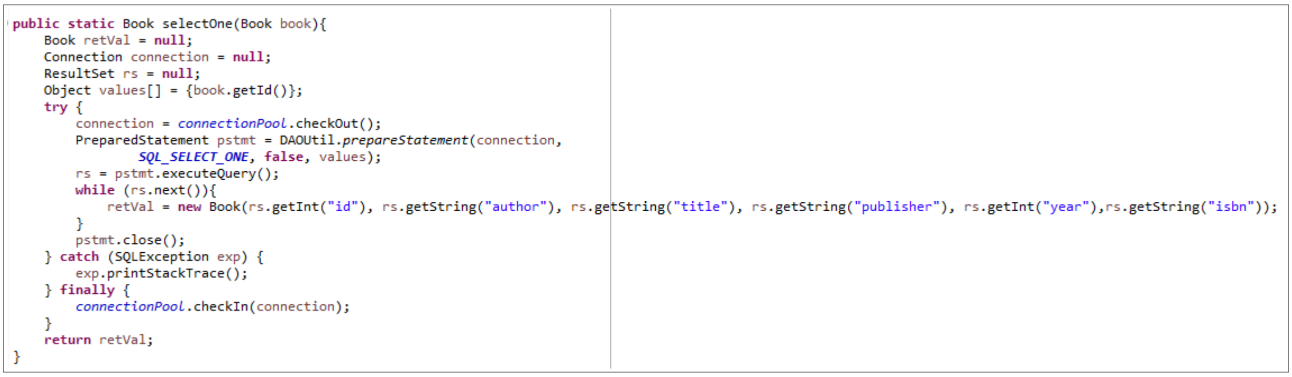
BookDAO.java



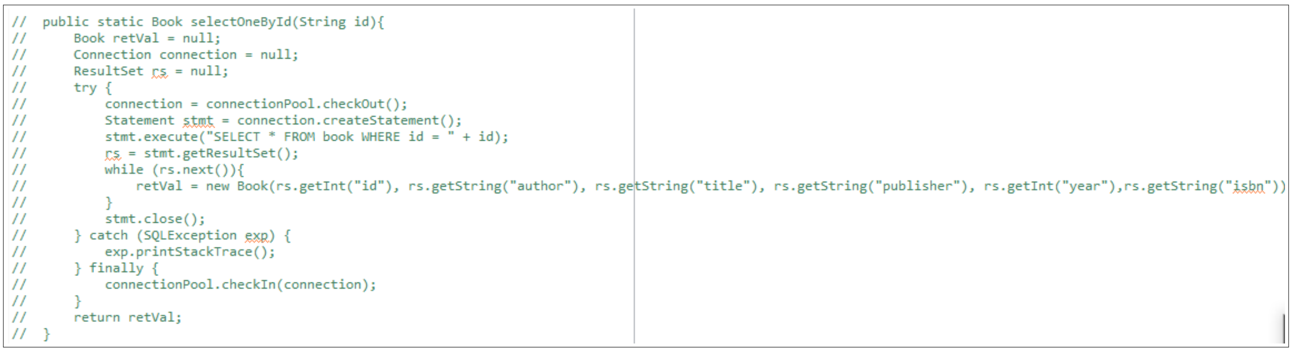
```java
package org.unibl.etf.bookapp.dao;

import java.sql.Connection;

public class BookDAO {
    private static ConnectionPool connectionPool = ConnectionPool.getConnectionPool();
    private static final String SQL_SELECT_ALL = "SELECT * FROM book";
    private static final String SQL_SELECT_ONE = "SELECT * FROM book WHERE id=?";
    private static final String SQL_INSERT = "INSERT INTO book (author, title, publisher, year, isbn) VALUES (?, ?, ?, ?, ?)";
    private static final String SQL_UPDATE = "UPDATE book set author=?, title=?, publisher=?, year=?, isbn=? WHERE id = ?";
```



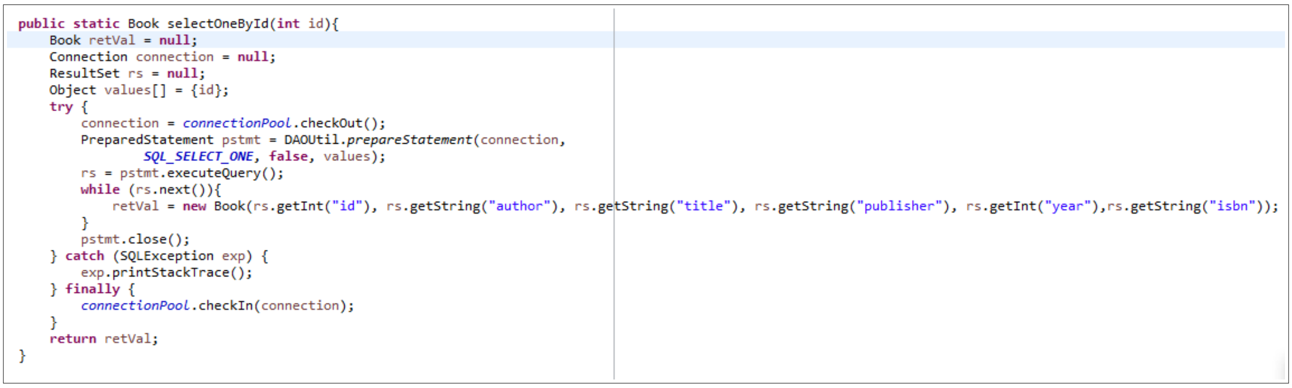
```java
public static ArrayList<Book> selectAll(){
    ArrayList<Book> retVal = new ArrayList<Book>();
    Connection connection = null;
    ResultSet rs = null;
    Object values[] = {};
    try {
        connection = connectionPool.checkOut();
        PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
                SQL_SELECT_ALL, false, values);
        rs = pstmt.executeQuery();
        while (rs.next()){
            retVal.add(new Book(rs.getInt("id"), rs.getString("author"), rs.getString("title"), rs.getString("publisher"), rs.getInt("year"),rs.getString("isbn")));
        }
        pstmt.close();
    } catch (SQLException exp) {
        exp.printStackTrace();
    } finally {
        connectionPool.checkIn(connection);
    }
    return retVal;
}
```



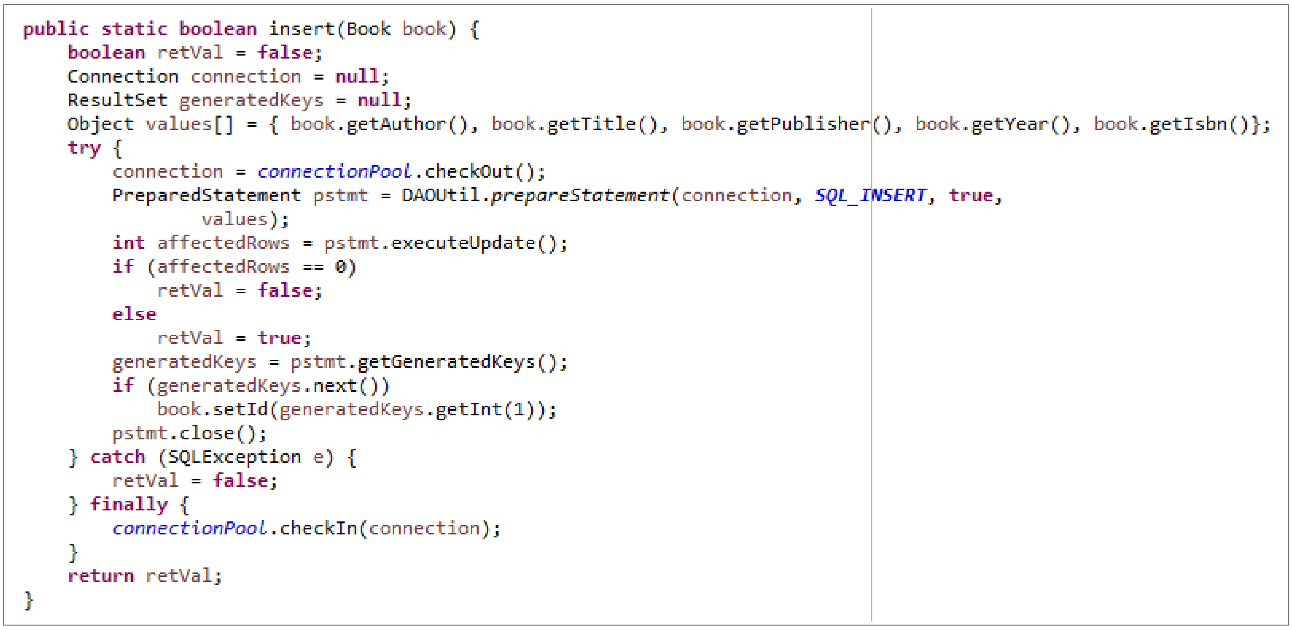
```
public static Book selectOne(Book book){
    Book retVal = null;
    Connection connection = null;
    ResultSet rs = null;
    Object values[] = {book.getId()};
    try {
        connection = connectionPool.checkOut();
        PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
                SQL_SELECT_ONE, false, values);
        rs = pstmt.executeQuery();
        while (rs.next()){
            retVal = new Book(rs.getInt("id"), rs.getString("author"), rs.getString("title"), rs.getString("publisher"), rs.getInt("year"),rs.getString("isbn"));
        }
        pstmt.close();
    } catch (SQLException exp) {
        exp.printStackTrace();
    } finally {
        connectionPool.checkIn(connection);
    }
    return retVal;
}
```



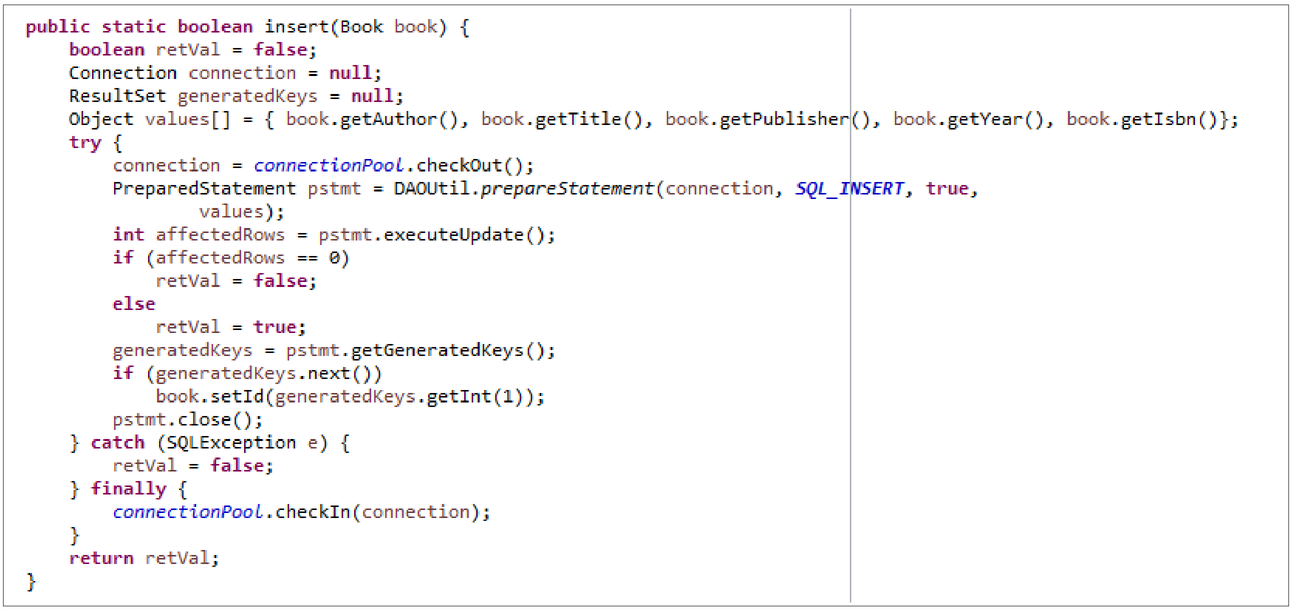
```
//  public static Book selectOneById(String id){
//      Book retVal = null;
//      Connection connection = null;
//      ResultSet rs = null;
//      try {
//          connection = connectionPool.checkOut();
//          Statement stmt = connection.createStatement();
//          stmt.execute("SELECT * FROM book WHERE id = " + id);
//          rs = stmt.getResultSet();
//          while (rs.next()){
//              retVal = new Book(rs.getInt("id"), rs.getString("author"), rs.getString("title"), rs.getString("publisher"), rs.getInt("year"),rs.getString("isbn"))
//          }
//          stmt.close();
//      } catch (SQLException exp) {
//          exp.printStackTrace();
//      } finally {
//          connectionPool.checkIn(connection);
//      }
//      return retVal;
//  }
```



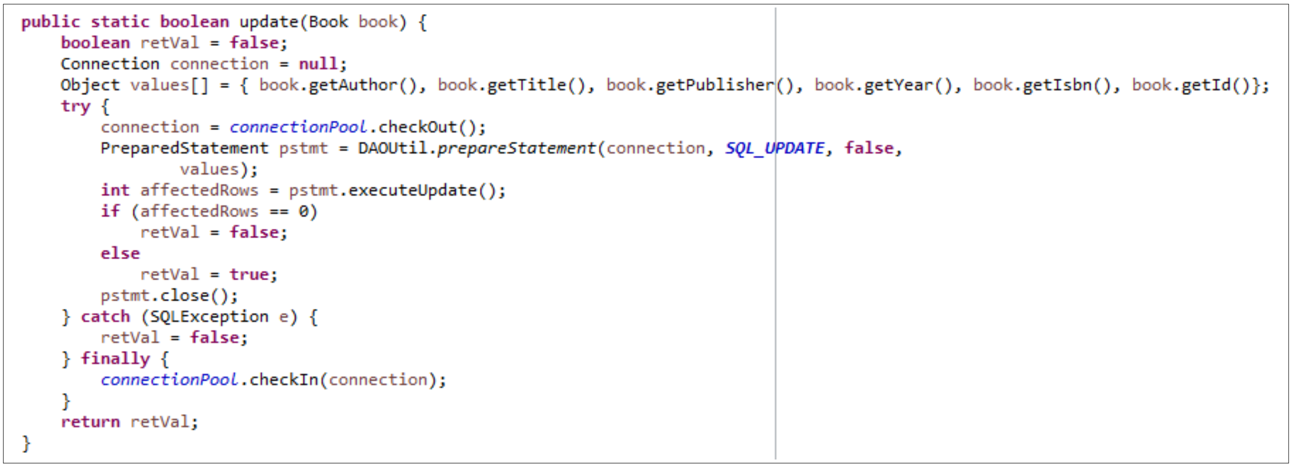
```
public static Book selectOneById(int id){
    Book retVal = null;
    Connection connection = null;
    ResultSet rs = null;
    Object values[] = {id};
    try {
        connection = connectionPool.checkOut();
        PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
                SQL_SELECT_ONE, false, values);
        rs = pstmt.executeQuery();
        while (rs.next()){
            retVal = new Book(rs.getInt("id"), rs.getString("author"), rs.getString("title"), rs.getString("publisher"), rs.getInt("year"),rs.getString("isbn"));
        }
        pstmt.close();
    } catch (SQLException exp) {
        exp.printStackTrace();
    } finally {
        connectionPool.checkIn(connection);
    }
    return retVal;
}
```



```
public static boolean insert(Book book) {
    boolean retVal = false;
    Connection connection = null;
    ResultSet generatedKeys = null;
    Object values[] = { book.getAuthor(), book.getTitle(), book.getPublisher(), book.getYear(), book.getIsbn()};
    try {
        connection = connectionPool.checkOut();
        PreparedStatement pstmt = DAOUtil.prepareStatement(connection, SQL_INSERT, true,
                values);
        int affectedRows = pstmt.executeUpdate();
        if (affectedRows == 0)
            retVal = false;
        else
            retVal = true;
        generatedKeys = pstmt.getGeneratedKeys();
        if (generatedKeys.next())
            book.setId(generatedKeys.getInt(1));
        pstmt.close();
    } catch (SQLException e) {
        retVal = false;
    } finally {
        connectionPool.checkIn(connection);
    }
    return retVal;
}
```

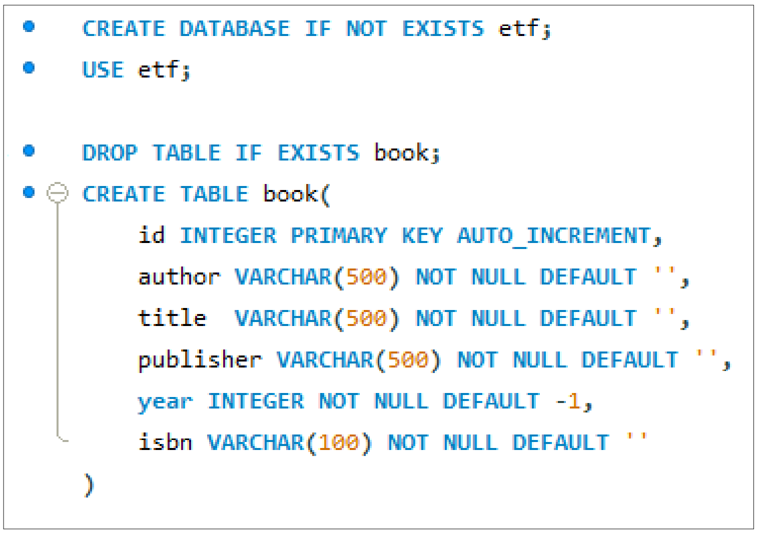


```java
public static boolean insert(Book book) {
    boolean retVal = false;
    Connection connection = null;
    ResultSet generatedKeys = null;
    Object values[] = { book.getAuthor(), book.getTitle(), book.getPublisher(), book.getYear(), book.getIsbn()};
    try {
        connection = connectionPool.checkOut();
        PreparedStatement pstmt = DAOUtil.prepareStatement(connection, SQL_INSERT, true,
                values);
        int affectedRows = pstmt.executeUpdate();
        if (affectedRows == 0)
            retVal = false;
        else
            retVal = true;
        generatedKeys = pstmt.getGeneratedKeys();
        if (generatedKeys.next())
            book.setId(generatedKeys.getInt(1));
        pstmt.close();
    } catch (SQLException e) {
        retVal = false;
    } finally {
        connectionPool.checkIn(connection);
    }
    return retVal;
}
```



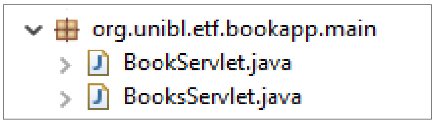
```java
public static boolean update(Book book) {
    boolean retVal = false;
    Connection connection = null;
    Object values[] = { book.getAuthor(), book.getTitle(), book.getPublisher(), book.getYear(), book.getIsbn(), book.getId()};
    try {
        connection = connectionPool.checkOut();
        PreparedStatement pstmt = DAOUtil.prepareStatement(connection, SQL_UPDATE, false,
                values);
        int affectedRows = pstmt.executeUpdate();
        if (affectedRows == 0)
            retVal = false;
        else
            retVal = true;
        pstmt.close();
    } catch (SQLException e) {
        retVal = false;
    } finally {
        connectionPool.checkIn(connection);
    }
    return retVal;
}
```

<u>database-mysql</u>



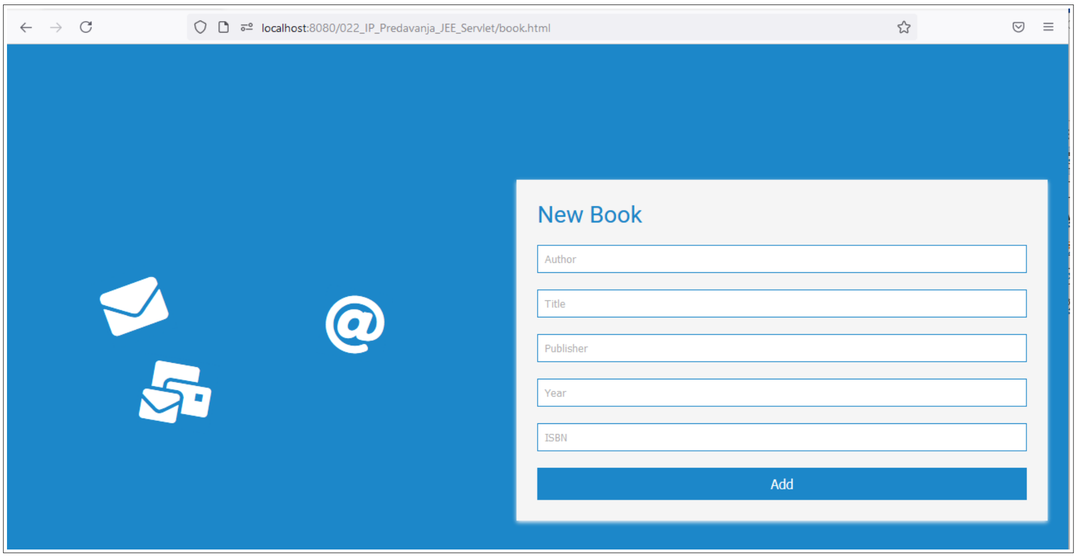
```sql
CREATE DATABASE IF NOT EXISTS etf;
USE etf;

DROP TABLE IF EXISTS book;
CREATE TABLE book(
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
    author VARCHAR(500) NOT NULL DEFAULT '',
    title  VARCHAR(500) NOT NULL DEFAULT '',
    publisher VARCHAR(500) NOT NULL DEFAULT '',
    year INTEGER NOT NULL DEFAULT -1,
    isbn VARCHAR(100) NOT NULL DEFAULT ''
)
```

book-servlet

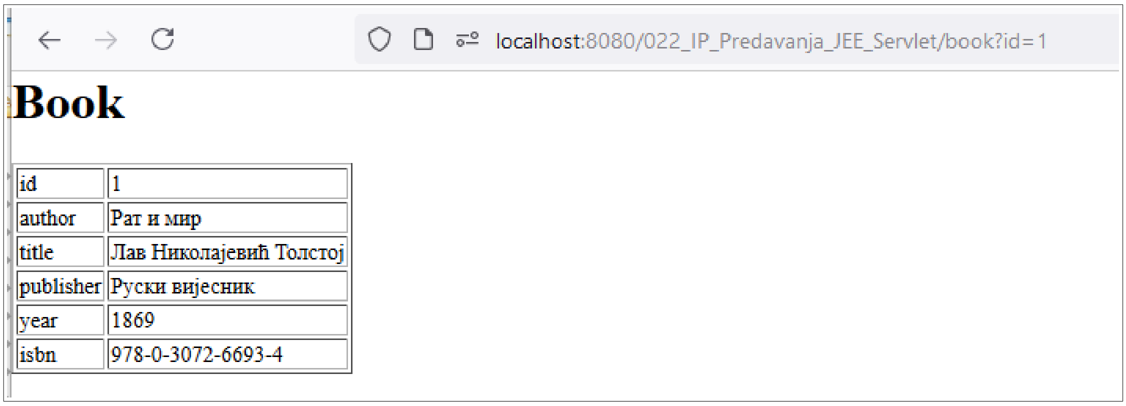


org.unibl.etf.bookapp.main
- BookServlet.java
- BooksServlet.java

BookServlet.java



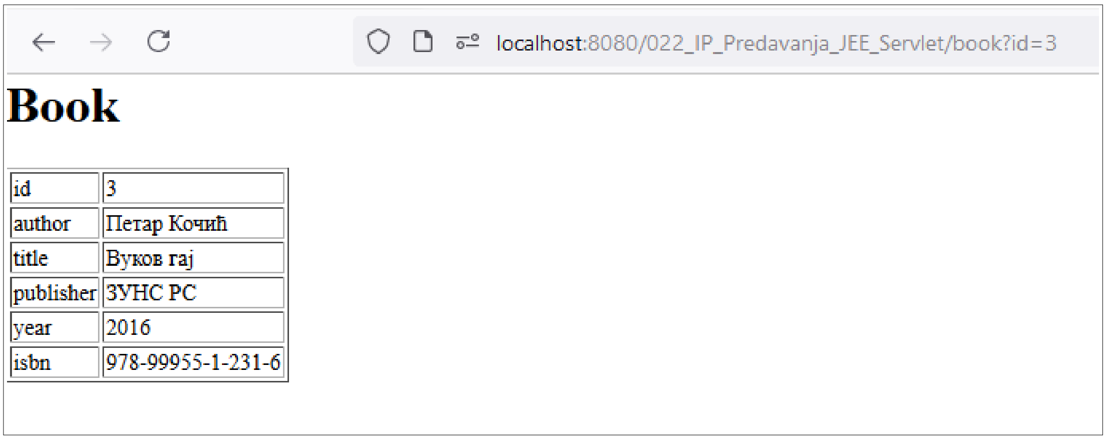
localhost:8080/022_IP_Predavanja_JEE_Servlet/book.html



localhost:8080/022_IP_Predavanja_JEE_Servlet/book?id=1

Book

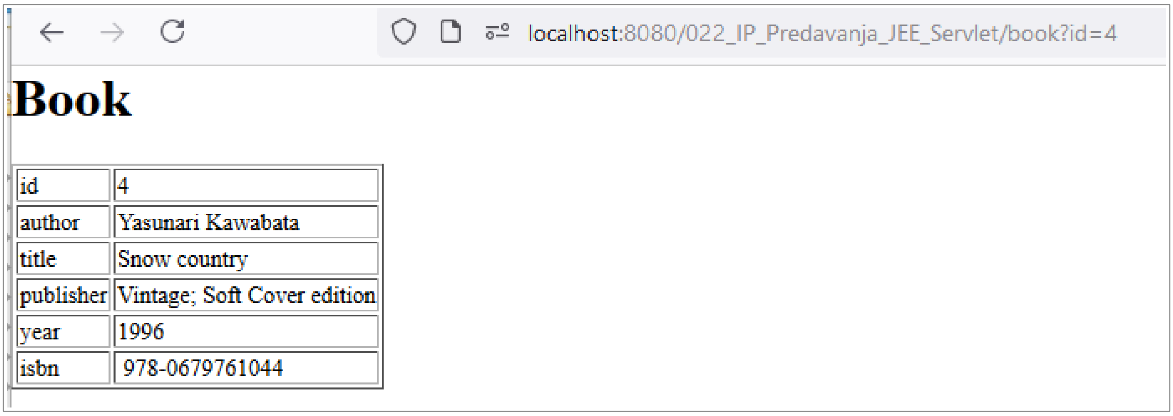
id	1
author	Рат и мир
title	Лав Николајевић Толстој
publisher	Руски вјесник
year	1869
isbn	978-0-3072-6693-4



localhost:8080/022_IP_Predavanja_JEE_Servlet/book?id=3

Book

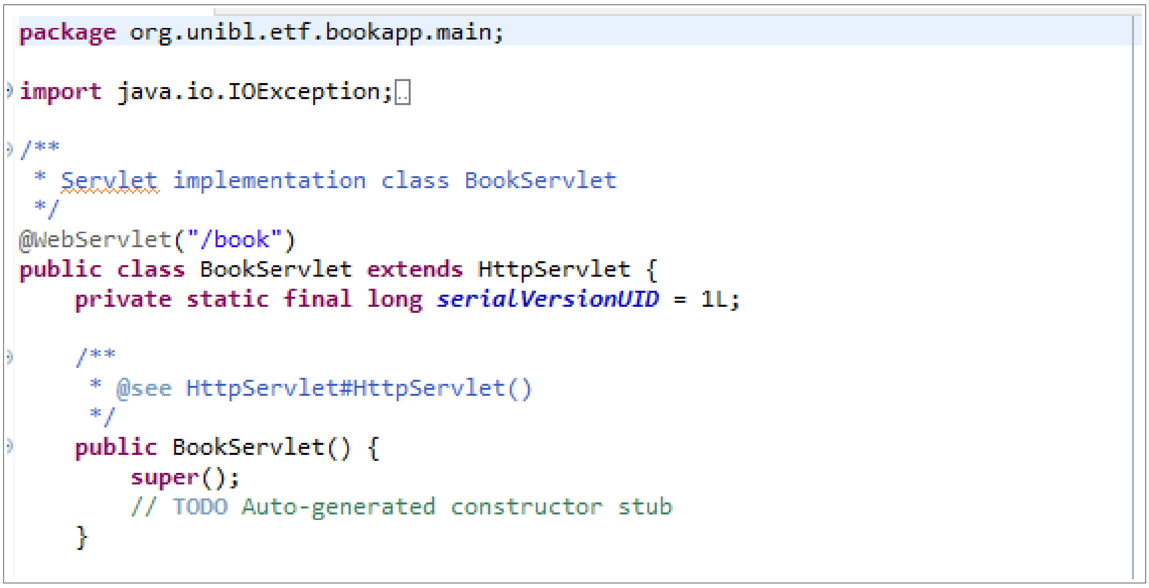
id	3
author	Петар Кочић
title	Буков рај
publisher	ЗУНС РС
year	2016
isbn	978-99955-1-231-6



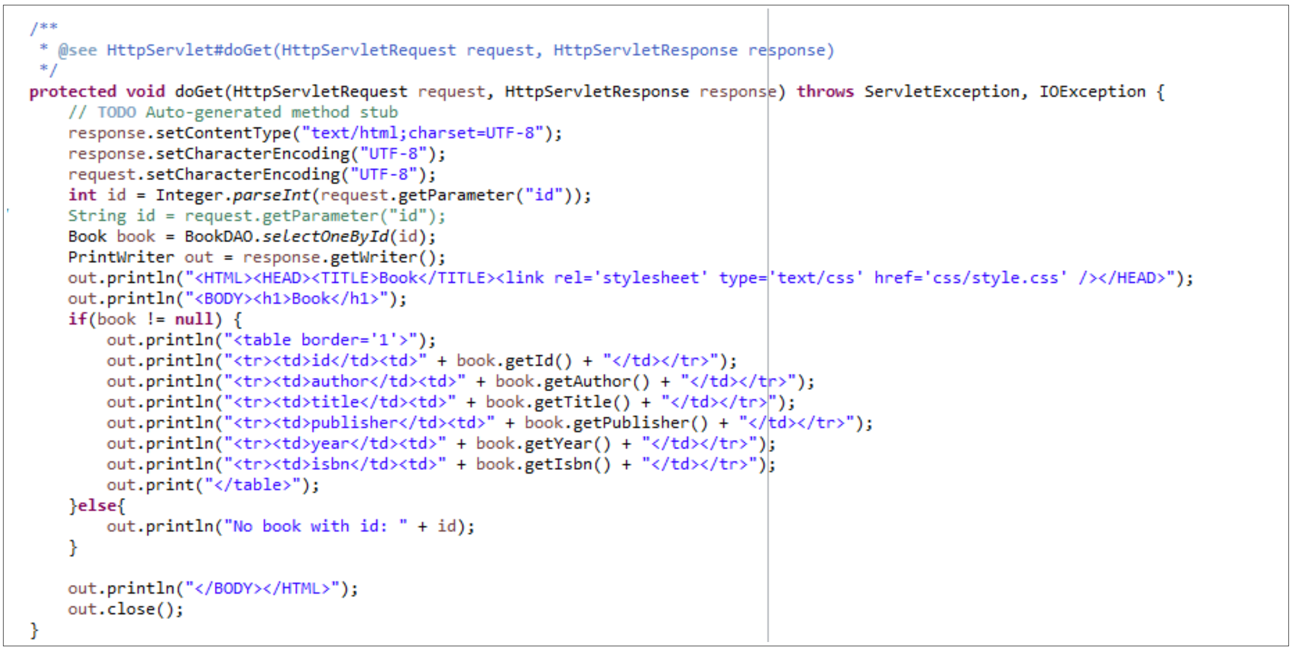
localhost:8080/022_IP_Predavanja_JEE_Servlet/book?id=4

Book

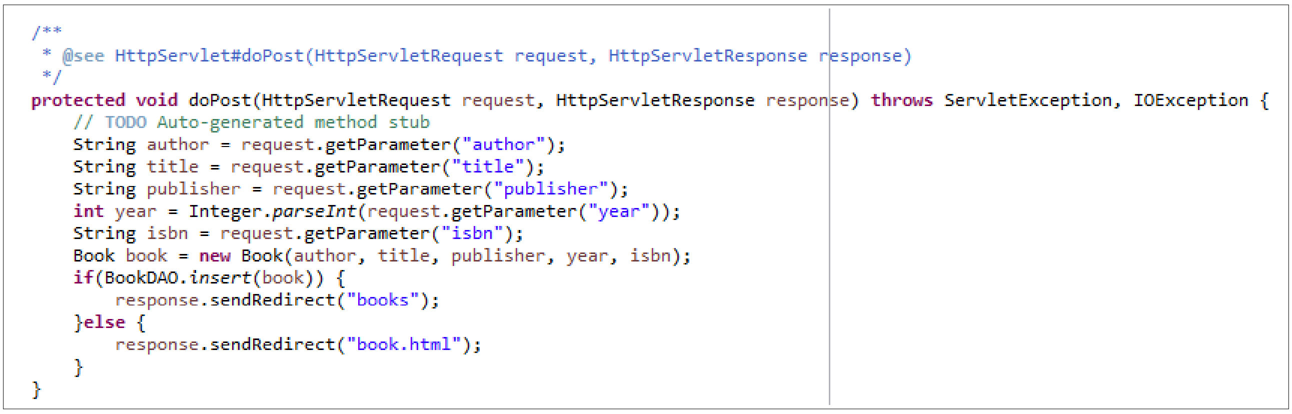
id	4
author	Yasunari Kawabata
title	Snow country
publisher	Vintage; Soft Cover edition
year	1996
isbn	978-0679761044



```java
package org.unibl.etf.bookapp.main;

import java.io.IOException;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public BookServlet() {
        super();
        // TODO Auto-generated constructor stub
    }
```

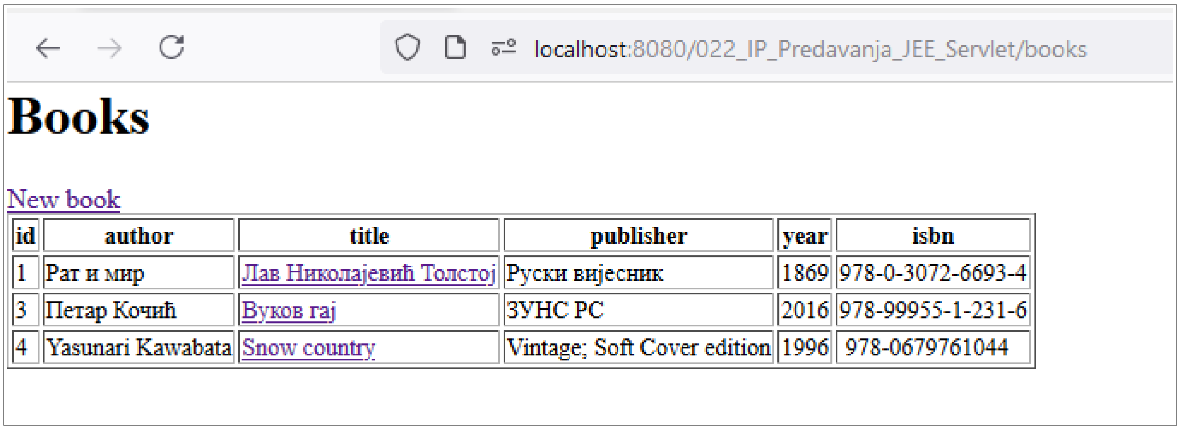


```java
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    response.setContentType("text/html;charset=UTF-8");
    response.setCharacterEncoding("UTF-8");
    request.setCharacterEncoding("UTF-8");
    int id = Integer.parseInt(request.getParameter("id"));
    String id = request.getParameter("id");
    Book book = BookDAO.selectOneById(id);
    PrintWriter out = response.getWriter();
    out.println("<HTML><HEAD><TITLE>Book</TITLE><link rel='stylesheet' type='text/css' href='css/style.css' /></HEAD>");
    out.println("<BODY><h1>Book</h1>");
    if(book != null) {
        out.println("<table border='1'>");
        out.println("<tr><td>id</td><td>" + book.getId() + "</td></tr>");
        out.println("<tr><td>author</td><td>" + book.getAuthor() + "</td></tr>");
        out.println("<tr><td>title</td><td>" + book.getTitle() + "</td></tr>");
        out.println("<tr><td>publisher</td><td>" + book.getPublisher() + "</td></tr>");
        out.println("<tr><td>year</td><td>" + book.getYear() + "</td></tr>");
        out.println("<tr><td>isbn</td><td>" + book.getIsbn() + "</td></tr>");
        out.print("</table>");
    }else{
        out.println("No book with id: " + id);
    }

    out.println("</BODY></HTML>");
    out.close();
}
```



```java
/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    String author = request.getParameter("author");
    String title = request.getParameter("title");
    String publisher = request.getParameter("publisher");
    int year = Integer.parseInt(request.getParameter("year"));
    String isbn = request.getParameter("isbn");
    Book book = new Book(author, title, publisher, year, isbn);
    if(BookDAO.insert(book)) {
        response.sendRedirect("books");
    }else {
        response.sendRedirect("book.html");
    }
}
```

BooksSevlet.jsp

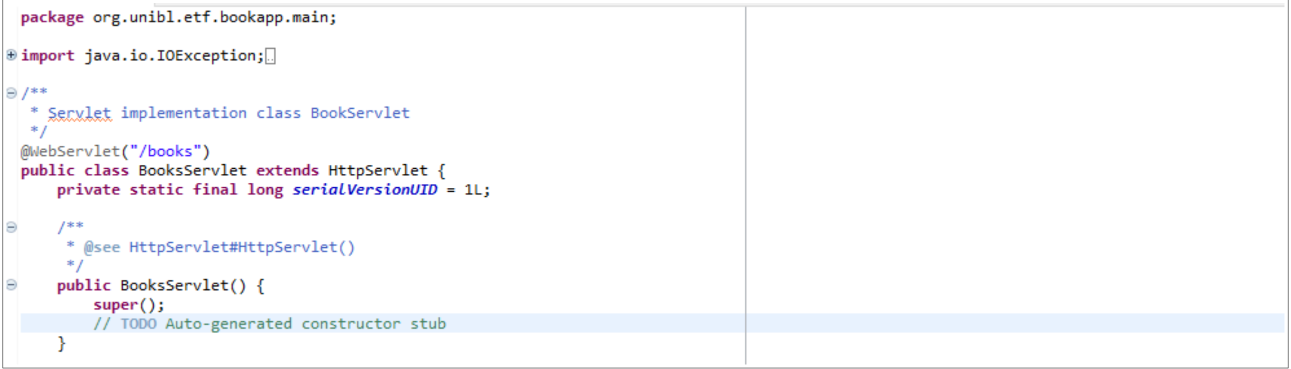


localhost:8080/022_IP_Predavanja_JEE_Servlet/books

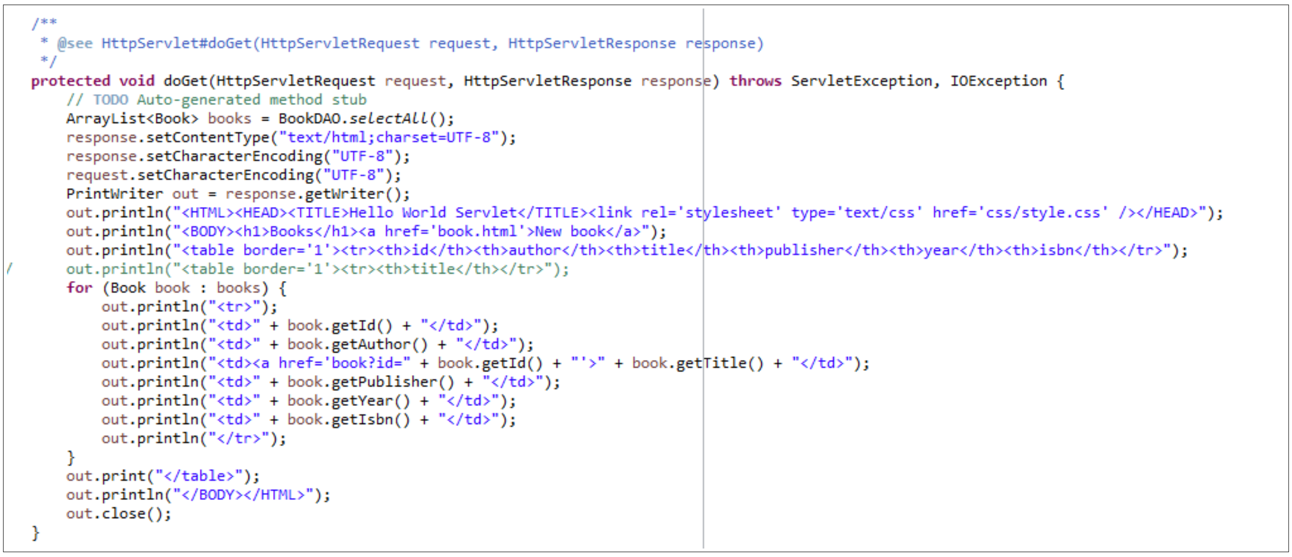
Books

New book

id	author	title	publisher	year	isbn
1	Рат и мир	Лав Николајевић Толстој	Руски вијесник	1869	978-0-3072-6693-4
3	Петар Кочић	Вуков raj	ЗУНС РС	2016	978-99955-1-231-6
4	Yasunari Kawabata	Snow country	Vintage; Soft Cover edition	1996	978-0679761044



```java
package org.unibl.etf.bookapp.main;

import java.io.IOException;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public BooksServlet() {
        super();
        // 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
        ArrayList<Book> books = BookDAO.selectAll();
        response.setContentType("text/html;charset=UTF-8");
        response.setCharacterEncoding("UTF-8");
        request.setCharacterEncoding("UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<HTML><HEAD><TITLE>Hello World Servlet</TITLE><link rel='stylesheet' type='text/css' href='css/style.css' /></HEAD>");
        out.println("<BODY><h1>Books</h1><a href='book.html'>New book</a>");
        out.println("<table border='1'><tr><th>id</th><th>author</th><th>title</th><th>publisher</th><th>year</th><th>isbn</th></tr>");
/       out.println("<table border='1'><tr><th>title</th></tr>");
        for (Book book : books) {
            out.println("<tr>");
            out.println("<td>" + book.getId() + "</td>");
            out.println("<td>" + book.getAuthor() + "</td>");
            out.println("<td><a href='book?id=" + book.getId() + "'>" + book.getTitle() + "</td>");
            out.println("<td>" + book.getPublisher() + "</td>");
            out.println("<td>" + book.getYear() + "</td>");
            out.println("<td>" + book.getIsbn() + "</td>");
            out.println("</tr>");
        }
        out.print("</table>");
        out.println("</BODY></HTML>");
        out.close();
    }
```