CÔNG NGHỆ JAVA

Nguyễn Hữu Thể

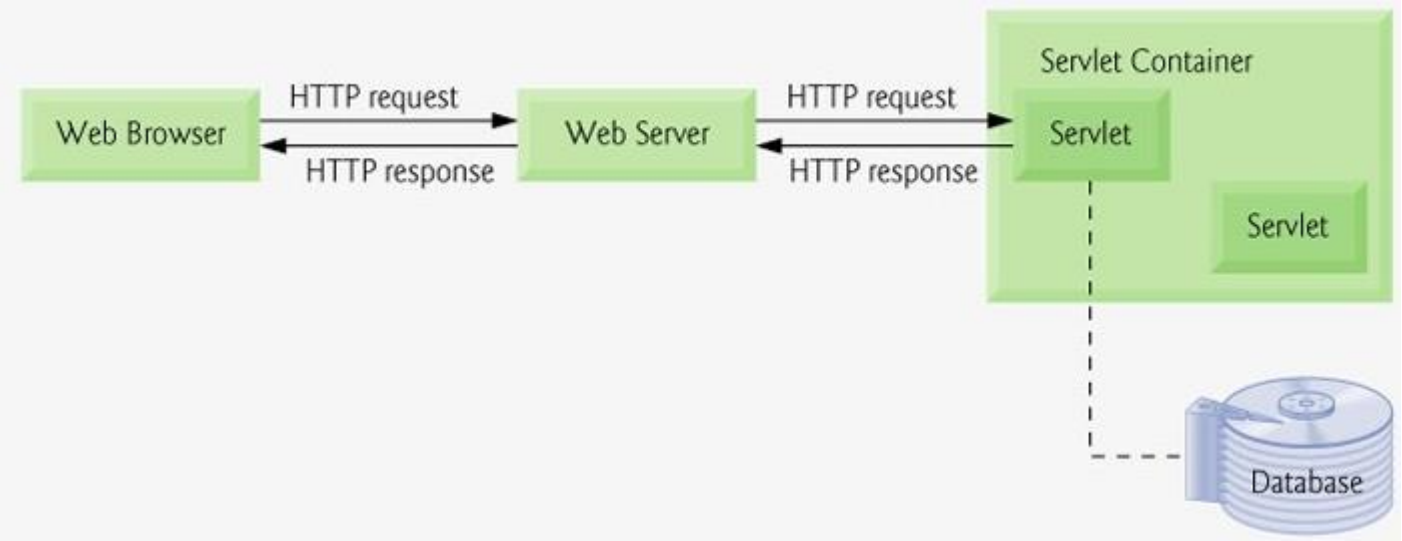
Bài 2: Servlet

Nội dung

- Servlet and Architecture
- Interface Servlet and the Servlet Life Cycle
- HttpServlet Class
- HttpServletRequest Interface
- HttpServletResponse Interface
- Project Servlet in Eclipse
- HTTP get Requests
- HTTP get Requests Containing Data
- HTTP post Requests
- Redirecting Requests to Other Resources
- Welcome Files

Servlet and Architecture

- Servlets: chạy trên **Web server** hoặc **Application server**.
- Tầng trung gian giữa HTTP **Client** với các **Database** hoặc các ứng dụng trên HTTP server.
- **Ưu điểm Servlets:**
  - Hiệu năng tốt. Tính bảo mật cao (dựa trên Java). Độc lập trên nền tảng.
  - Thư viện Java, giao tiếp với Applet, Database hoặc phần mềm khác.

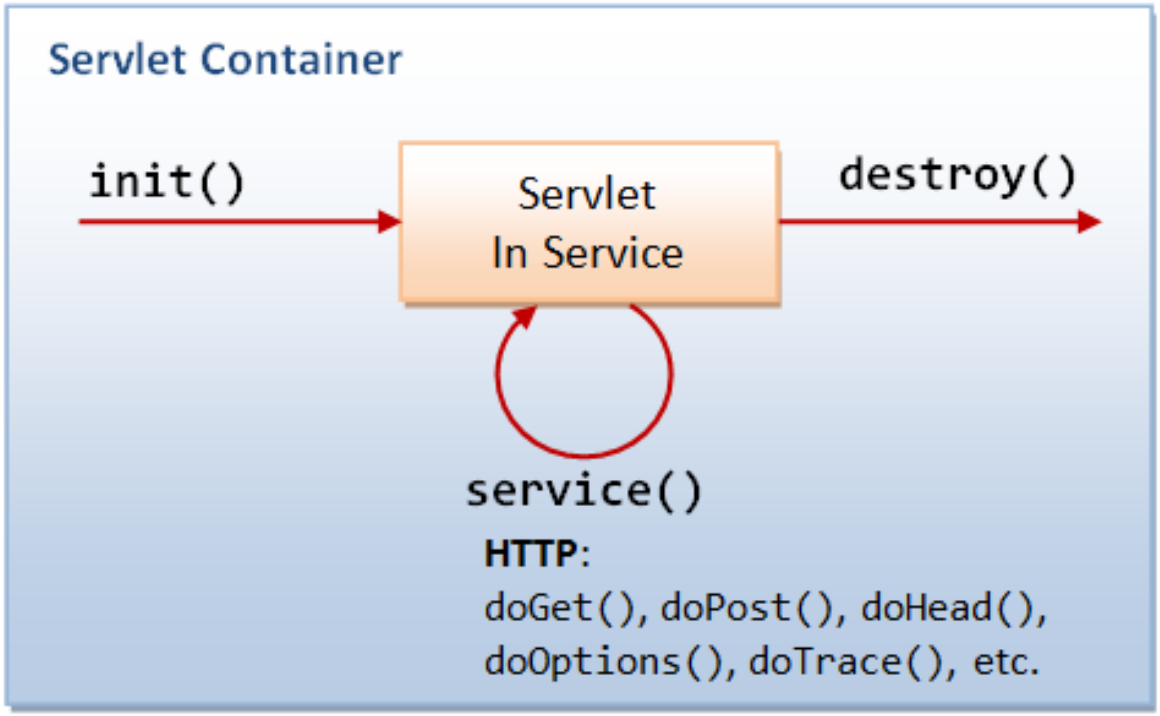


Servlet Package

- Java Servlets sử dụng các lớp Java => Run bởi **Web Server**.
- Các gói thư viện hỗ trợ:
  - javax.servlet
  - javax.servlet.http
- Servlet được biên dịch giống như các lớp khác trong Java.
- Các interface và các lớp trong API Servlet
  - Servlet,
  - GenericServlet,
  - HttpServlet,
  - ServletRequest,
  - ServletResponse, …

Servlet Life Cycle

1. Servlet được khởi tạo bằng cách gọi phương thức **init()**
2. Phương thức **service()** được gọi để xử lý yêu cầu của client.
3. Servlet được hủy bằng phương thức **destroy()**
4. Cuối cùng, servlet được thu thập bởi bộ sưu tập rác của JVM.



Interface Servlet

Method	Description
public void init(ServletConfig config)	initializes the servlet. It is the life cycle method of servlet and invoked by the web container only once.
public void service(ServletRequest request,ServletResponse response)	provides response for the incoming request. It is invoked at each request by the web container.
public void destroy()	is invoked only once and indicates that servlet is being destroyed.
public ServletConfig getServletConfig()	returns the object of ServletConfig.
public String getServletInfo()	returns information about servlet such as writer, copyright, version etc.

Method Detail – init()

public void init(ServletConfig config) throws ServletException

- Phương thức **init()** được gọi chỉ một lần để khởi tạo servlet.
- Khi gọi servlet, một thể hiện duy nhất của mỗi servlet sẽ được tạo ra, với mỗi yêu cầu của người dùng tạo ra một luồng mới được trao cho doGet hoặc doPost.
- Parameters:
  - **config** - a ServletConfig object containing the servlet's configuration and initialization parameters

Method Detail – service()

public void service(ServletRequest req, ServletResponse res) throws ServletException, java.io.IOException

- Phương thức chính để thực hiện nhiệm vụ. Web server gọi phương thức service() để xử lý các yêu cầu đến từ client và trả về kết quả.
- Máy chủ tạo ra 1 Thread mới khi nhận được 1 yêu cầu cho 1 servlet, và gọi phương thức service().
- Phương thức service() kiểm tra kiểu yêu cầu HTTP (GET, POST, PUT, DELETE, v.v.) và gọi các phương thức doGet, doPost, doPut, doDelete,...
- Parameters:
  - req - the ServletRequest object that contains the client's request
  - res - the ServletResponse object that contains the servlet's response

Method Detail – getServletConfig()

public ServletConfig getServletConfig()

- Returns a ServletConfig object, which contains initialization and startup parameters for this servlet.

- Returns:
  - the ServletConfig object that initializes this servlet

Method Detail – getServletInfo()

public java.lang.String getServletInfo()

- Returns **information** about the servlet, such as **author**, version, and copyright.
- The string that this method returns should be **plain text** and not markup of any kind (such as HTML, XML, etc.).

- Returns:
  - a String containing servlet information

Method Detail - destroy

public void destroy()

- Chỉ được gọi một lần ở giai đoạn cuối trong vòng đời Servlet.
- Giúp servlet đóng các kết nối tới Database, dừng thread, thực hiện các hoạt động cleanup.
- Sau khi destroy() được gọi, đối tượng servlet này được đánh dấu cho Garbage Collector.

HttpServlet Class

```
java.lang.Object
        |_extended byjavax.servlet.GenericServlet
                |_extended byjavax.servlet.http.HttpServlet
```

- Servlets kế thừa từ lớp HttpServlet.
- Hai loại HTTP requests phổ biến nhất là **get** và **post**.
- Lớp HttpServlet định nghĩa phương thức **doGet** và **doPost** để respond và requests từ client.
- Được gọi bởi servlet container khi request đến server.

HttpServlet Class

public abstract class HttpServlet extends GenericServlet implements java.io.Serializable

- Cung cấp lớp trừu tượng để tạo một HTTP servlet phù hợp.
- Lớp con của HTTPServlet phải ghi đè ít nhất một trong các phương thức sau:
  - doGet, nếu servlet hỗ trợ HTTP GET requests
  - doPost, cho HTTP POST requests
  - doPut, cho HTTP PUT requests
  - doDelete, cho HTTP DELETE requests
  - init và destroy, để quản lý tài nguyên được giữ
  - getServletInfo: servlet cung cấp thông tin về chính nó

HttpServlet Class

Method Summary	
protected void	doDelete(HttpServletRequest req, HttpServletResponse resp) Called by the server (via the service method) to allow a servlet to handle a DELETE request.
protected void	doGet(HttpServletRequest req, HttpServletResponse resp) Called by the server (via the service method) to allow a servlet to handle a GET request.
protected void	doHead(HttpServletRequest req, HttpServletResponse resp) Receives an HTTP HEAD request from the protected service method and handles the request.
protected void	doOptions(HttpServletRequest req, HttpServletResponse resp) Called by the server (via the service method) to allow a servlet to handle a OPTIONS request.
protected void	doPost(HttpServletRequest req, HttpServletResponse resp) Called by the server (via the service method) to allow a servlet to handle a POST request.

HttpServlet Class

Method Summary	
protected void	doPut(HttpServletRequest req, HttpServletResponse resp) Called by the server (via the service method) to allow a servlet to handle a PUT request.
protected void	doTrace(HttpServletRequest req, HttpServletResponse resp) Called by the server (via the service method) to allow a servlet to handle a TRACE request.
protected long	getLastModified(HttpServletRequest req) Returns the time the HttpServletRequest object was last modified, in milliseconds since midnight January 1, 1970
protected void	service(HttpServletRequest req, HttpServletResponse resp) Receives standard HTTP requests from the public service method and dispatches them to the doXXX methods defined in this class.
void	service(ServletRequest req, ServletResponse res) Dispatches client requests to the protected service method.

HttpServletRequest Interface

- **String getParameter(String name)**
  - Value of a parameter sent to the servlet as part of a get or post request. The name argument represents the parameter name.
- **Enumeration getParameterNames()**
  - Returns the names of all the parameters sent to the servlet as part of a post request.
- **String[] getParameterValues(String name)**
  - For a parameter with multiple values, this method returns an array of strings containing the values for a specified servlet parameter.
- **Cookie[] getCookies()**
  - Returns an array of Cookie objects stored on the client by the server. Cookie objects can be used to uniquely identify clients to the servlet.

HttpServletRequest Interface

- **HttpSession getSession(boolean create)**
  - Returns an HttpSession object associated with the client's current browsing session.
- **String getLocalName()**
  - Gets the host name on which the request was received.
- **String getLocalAddr()**
  - Gets the Internet Protocol (IP) address on which the request was received.
- **int getLocalPort()**
  - Gets the Internet Protocol (IP) port number on which the request was received.

HttpServletResponse Interface

- **void addCookie(Cookie cookie)**
  - Add a Cookie to the header of the response to the client.
- **ServletOutputStream getOutputStream()**
  - Obtains a byte-based output stream for sending binary data to the client.
- **PrintWriter getWriter()**
  - Obtains a character-based output stream for sending text data (usually HTML formatted text) to the client.

HttpServletResponse Interface

- **void setContentType(String type)**
  - Specifies the content type of the response to the browser.
  - For examples, content type "text/html", "image/gif",...

- **String getContentType()**
  - Gets the content type of the response.

Project Servlet trong Eclipse

Java EE - Servlet/src/WelcomeServlet.java - Eclipse

File Edit Source Refactor Navigate Search Project Scripts Run Window Help

Java EE

Database De...

Project Explorer

- Servlet
  - Deployment Descriptor: Servlet
  - Java Resources: src
    - (default package)
      - WelcomeServlet.java
    - Libraries
  - JavaScript Resources
  - build
  - WebContent
    - META-INF
    - WEB-INF
    - WelcomeServlet.html

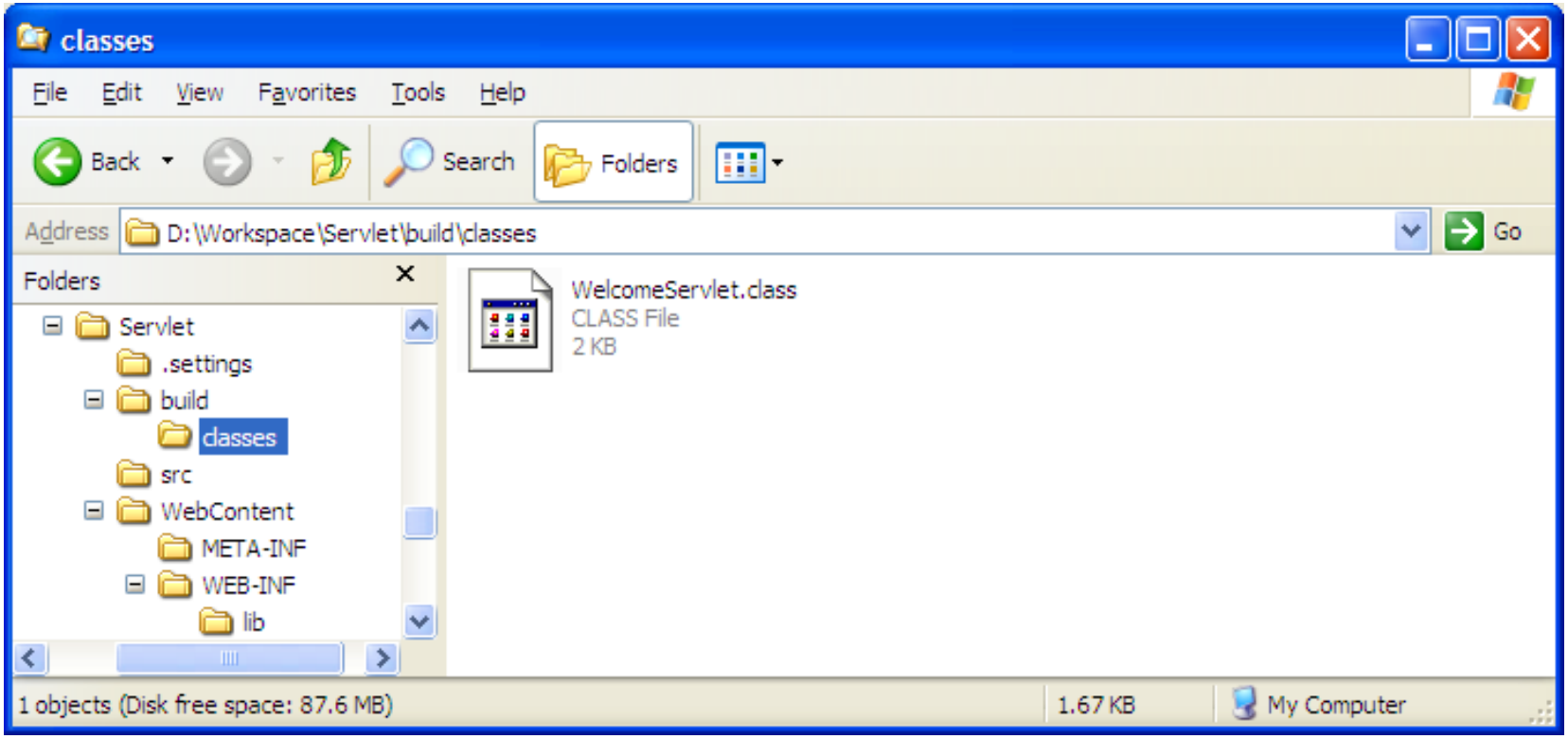
WelcomeServlet.java | A Simple Servlet Exa | web.xml

```
12 public class WelcomeServlet extends HttpServlet {
13     private static final long serialVersionUID = 1L;
14
15     /**
16      * @see HttpServlet#HttpServlet()
17      */
18     public WelcomeServlet() {
19         super();
20         // TODO Auto-generated constructor stub
21     }
22
23     /**
24      * @see HttpServlet#doGet(HttpServletRequest request
```

Servlet

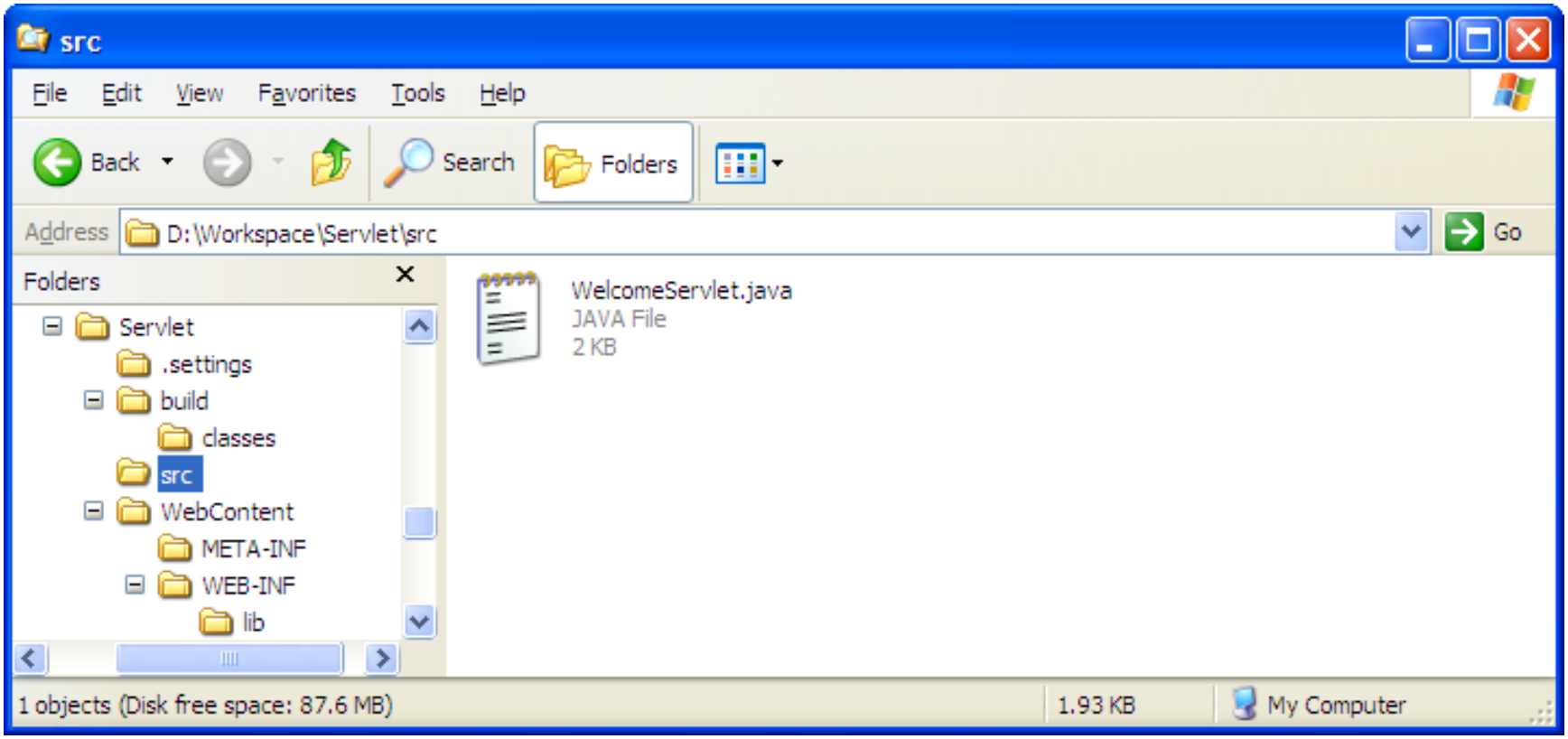
Project Servlet trong Eclipse

– File **WelcomeServlet.java** sau khi biên dịch thành file **WelcomeServlet.class** được lưu trong thư mục **build/class/**



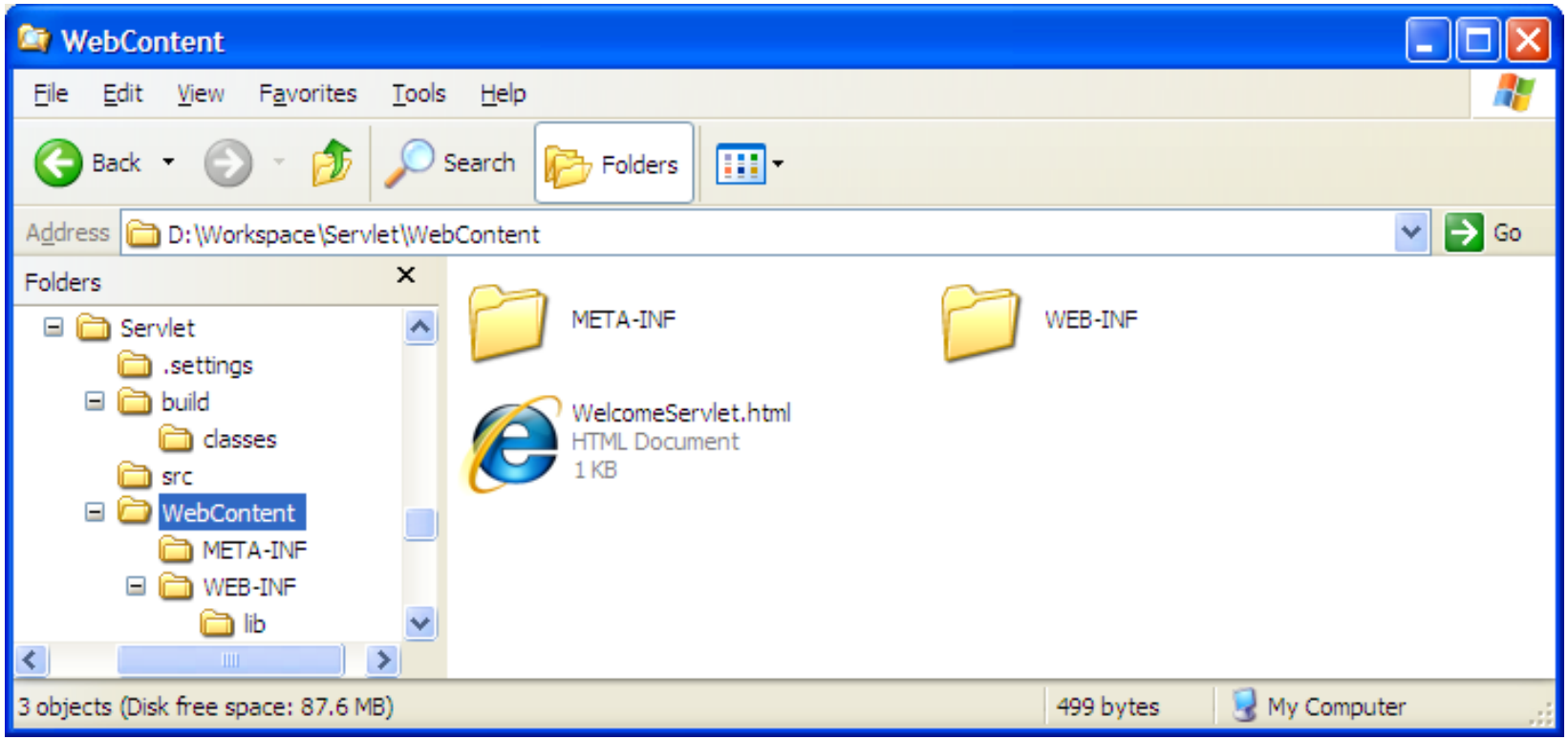
Project Servlet trong Eclipse

– Class java lưu trong thư mục **src**



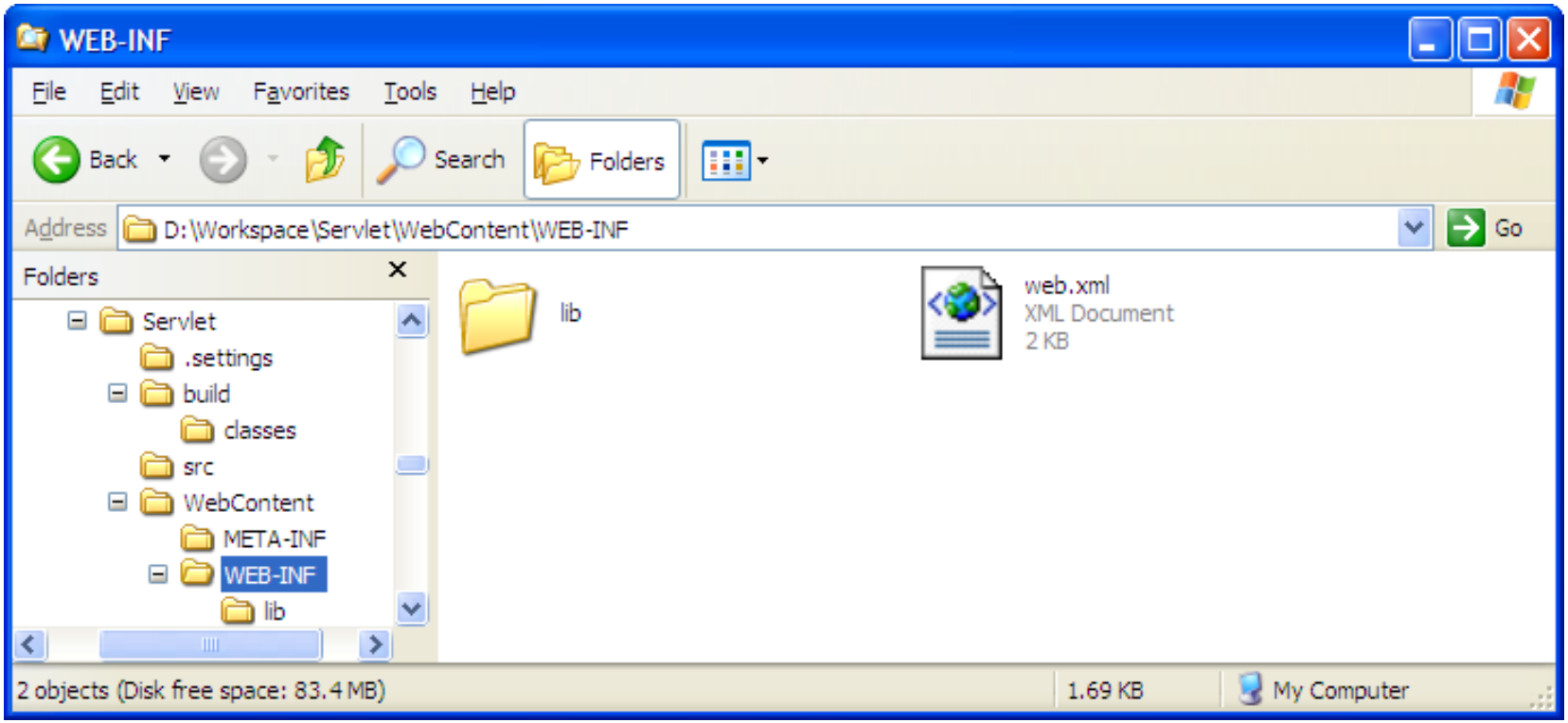
Project Servlet trong Eclipse

– Trang html, jsp lưu trong thư mục **WebContent**



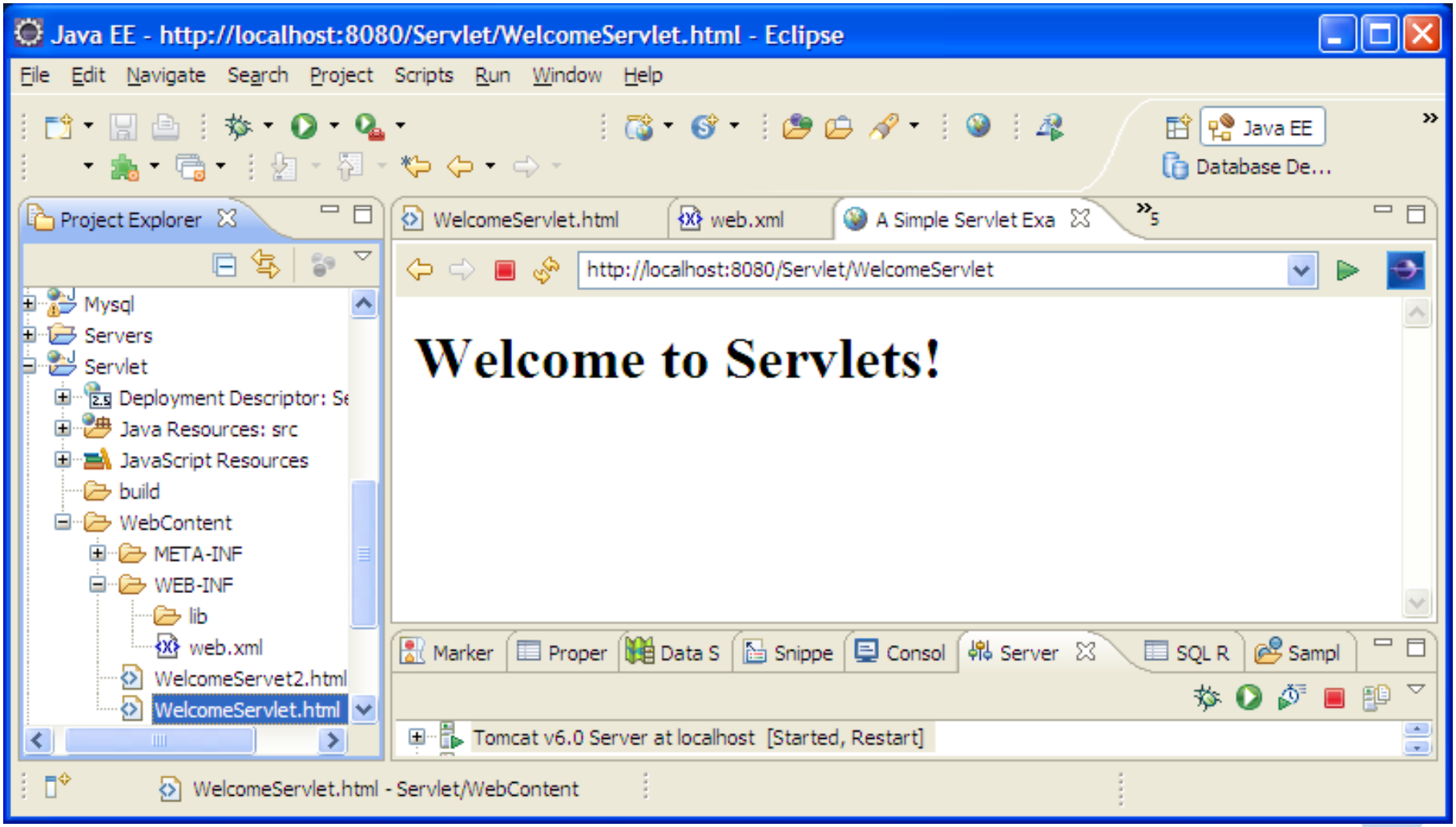
Project Servlet trong Eclipse

– File web.xml lưu trong thư mục **WEB-INF**



Project Servlet trong Eclipse

– Run: right click file html / Run as/ Run on Server



HTTP get Requests

- HTTP get request is to retrieve the content of a URL.

```
public class WelcomeServlet extends HttpServlet
{
    // process "get" requests from clients
    protected void doGet( HttpServletRequest request,
      HttpServletResponse response ) throws
      ServletException, IOException {
      //…
    }
}
```

HTTP get Requests

WelcomeServlet.java

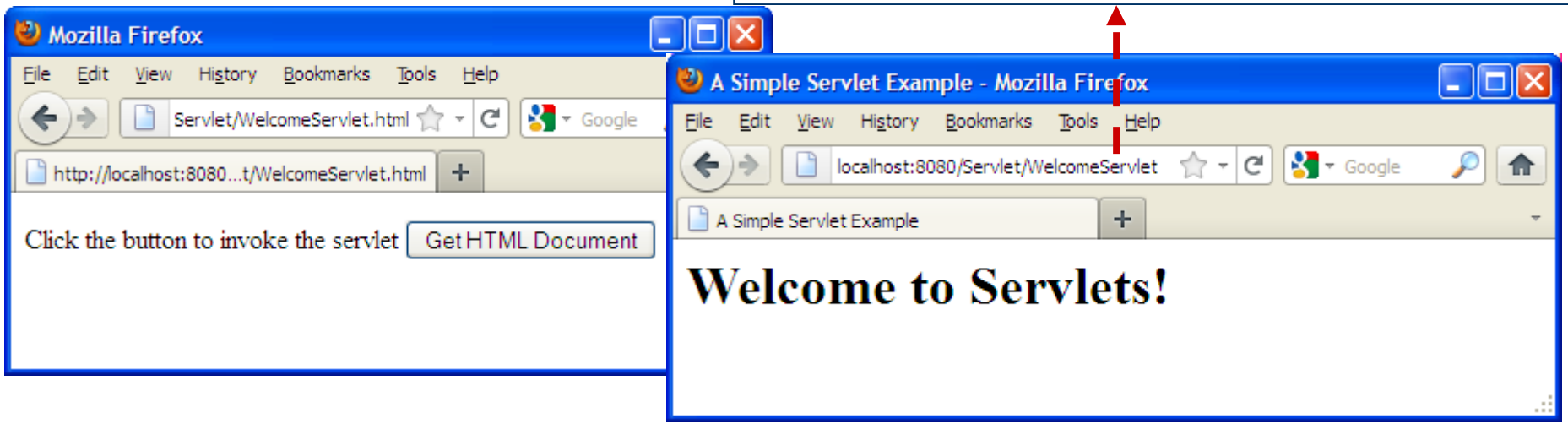
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
@WebServlet("/WelcomeServlet")
public class WelcomeServlet extends HttpServlet {
    public WelcomeServlet() { //... }
    protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<head>");
        out.println("<title>A Simple Servlet Example</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Welcome to Servlets!</h1>");
        out.println("</body>");
        out.println("</html>");
        out.close();}
    protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    }
}
```

HTTP get Requests

```
<!-- WelcomeServlet.html -->
<html>
<body>
   <form action = "WelcomeServlet" method = "get" >
   <p>Click the button to invoke the servlet
   <input type = "submit" value = "Get HTML Document" />
   </p>
   </form>
</body>
</html>
```

WelcomeServlet.class

http://localhost:8080/Servlet/WelcomeServlet



Web.xml

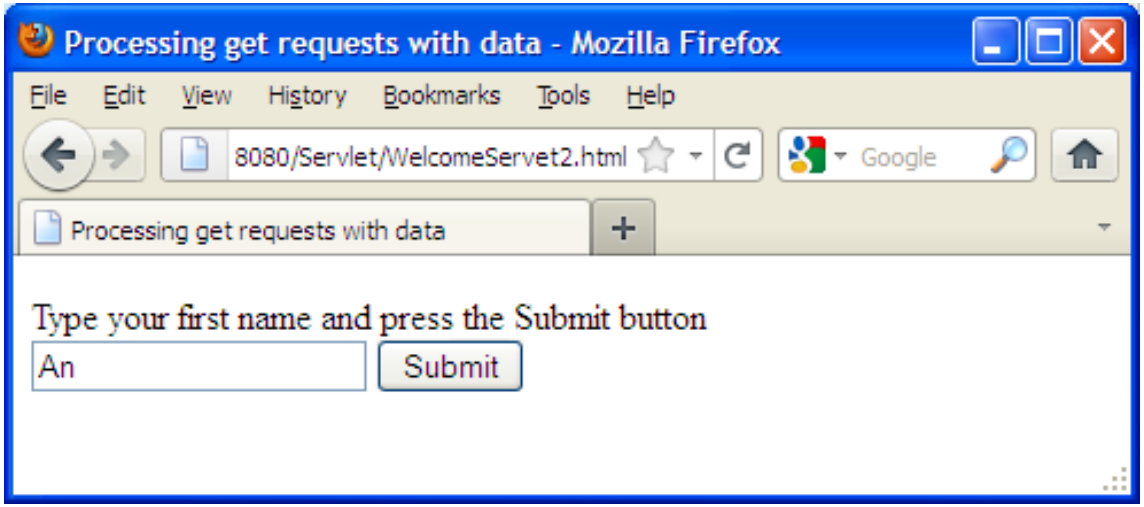
```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>Servlet</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-list>
  <servlet>
    <description></description>
    <display-name>WelcomeServlet</display-name>
    <servlet-name>WelcomeServlet</servlet-name>
    <servlet-class>WelcomeServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>WelcomeServlet</servlet-name>
    <url-pattern>/WelcomeServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

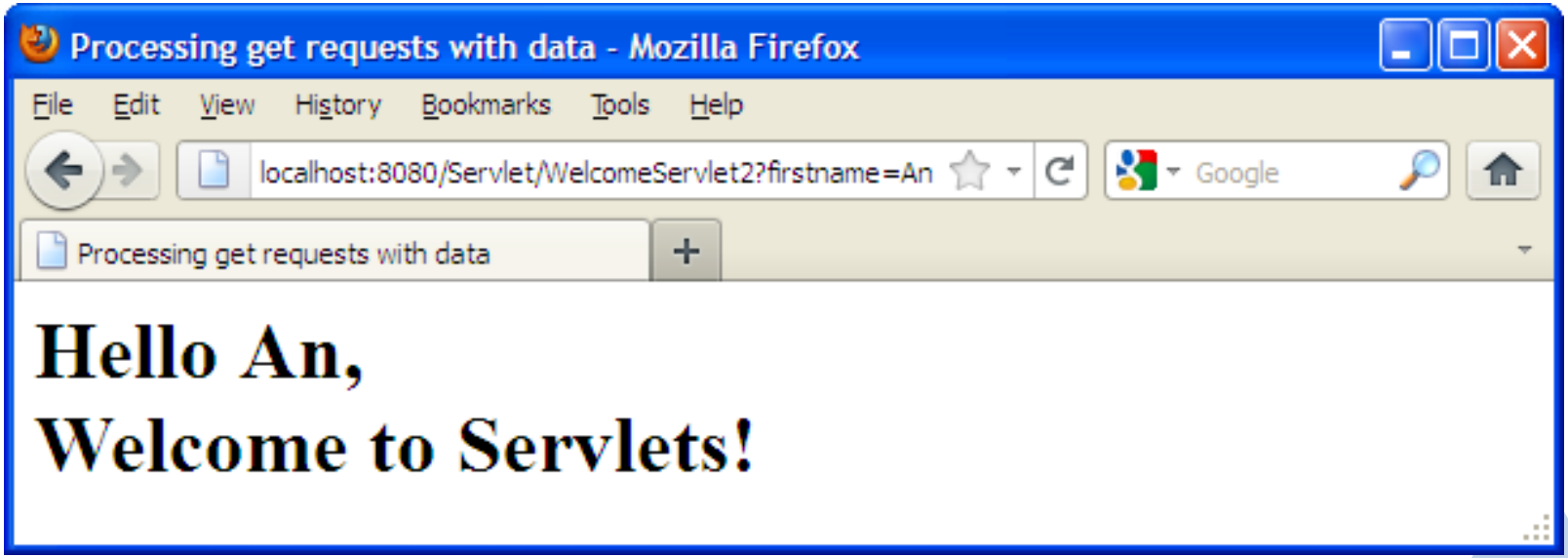
```java
import java.io.*;
import javax.*;
import javax.servlet.*;
@WebServlet("/WelcomeServlet2")
public class WelcomeServlet2 extends HttpServlet {
    public WelcomeServlet2() {
        super();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        String firstName = request.getParameter("firstname");
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Processing get requests with data</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello " + firstName + ",<br />");
        out.println("Welcome to Servlets!</h1>");
        out.println("</body>");
        out.println("</html>");
        out.close();
    }
```

HTTP get Requests Containing Data

```
<!-- WelcomeServlet2.html -->
<html>
<head>
    <title>Processing get requests with data </title>
</head>

<body>
<form action = "WelcomeServlet2" method = "get" >
   <p>
   Type your first name and press the Submit button
   <br /><input type = "text" name = "firstname" />
   <input type = "submit" value = "Submit" />
    </p>
</form>
</body>
</html>
```

HTTP get Requests Containing Data





web.xml

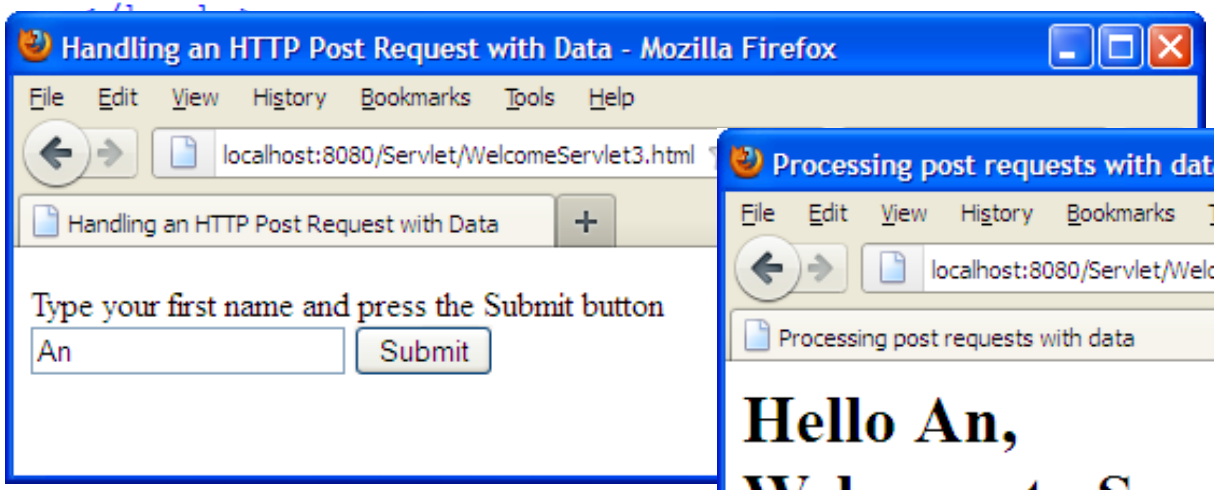
```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <display-name>Servlet</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <display-name>WelcomeServlet2</display-name>
    <servlet-name>WelcomeServlet2</servlet-name>
    <servlet-class>WelcomeServlet2</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>WelcomeServlet2</servlet-name>
    <url-pattern>/WelcomeServlet2</url-pattern>
  </servlet-mapping>
</web-app>
```

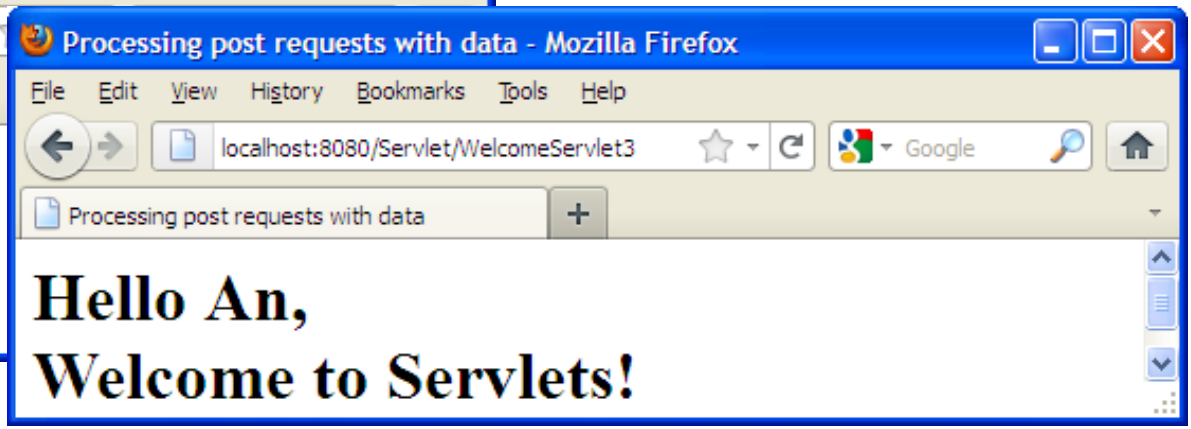
HTTP post Requests

```
import java.io.*;
import javax.*;
import javax.servlet.*;
@WebServlet("/WelcomeServlet3")
public class WelcomeServlet3 extends HttpServlet {
    public WelcomeServlet3() { super(); }

    protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        String firstName = request.getParameter( "firstname" );
        response.setContentType( "text/html" );
        PrintWriter out = response.getWriter();
        out.println( "<html>" );
        out.println( "<head>" );
        out.println( "<title>Processing post requests with data</title>" );
        out.println( "</head>" );
        out.println( "<body>" );
        out.println( "<h1>Hello " + firstName + ",<br />" );
        out.println( "Welcome to Servlets!</h1>" );
        out.println( "</body>" );
        out.println( "</html>" );
        out.close();
    }
}
```

HTTP post Requests

```
<!-- WelcomeServlet3.html -->
<html>
<body>
   <form action = "WelcomeServlet3" method = "post">
      <p>
         Type your first name and press the Submit button<br />
         <input type = "text" name = "firstname" />
         <input type = "submit" value = "Submit" />
      </p>
   </form>
```





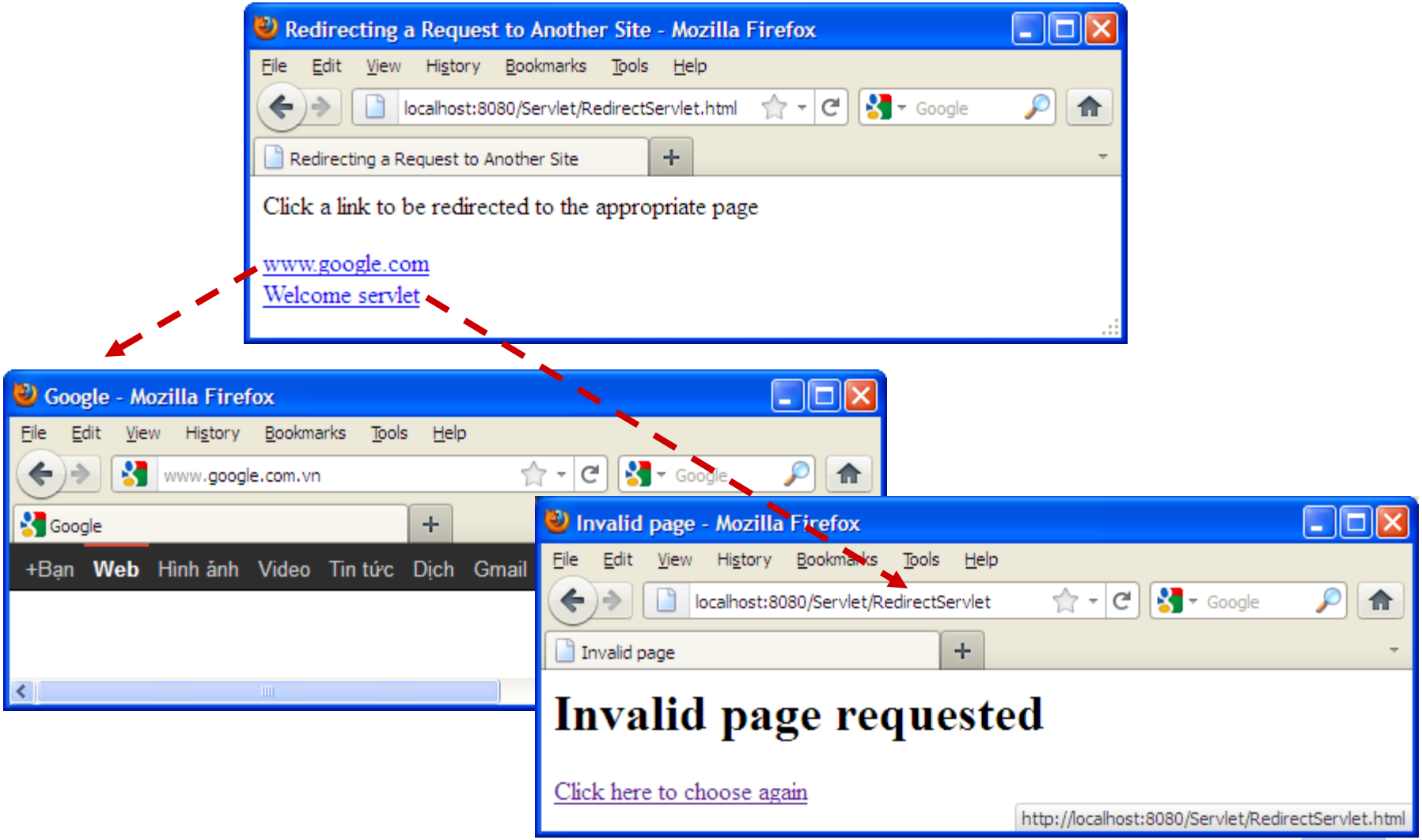
Redirecting Requests to Other Resources

```
// RedirectServlet.java .Redirecting a user to a different Web page.
...
public class RedirectServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException,IOException{
        String location = request.getParameter( "page" );
          if ( location != null ){
             if ( location.equals( "google" ) )
                response.sendRedirect( "http://www.google.com" );
             else if ( location.equals( "RedirectServlet" ) )
                response.sendRedirect( "RedirectServlet" );
          }
          // code that executes only if this servlet
          // does not redirect the user to another page
          response.setContentType( "text/html" );
          PrintWriter out = response.getWriter();
          out.println("<html>" );
          out.println( "<body>" );
          out.println( "<h1>Invalid page requested</h1>" );
          out.println( "<p><a href = RedirectServlet.html > ");
          out.println( "Click here to choose again</a></p>" );
          ...
```

Redirecting Requests to Other Resources

```
<!-- RedirectServlet.html -->
<html>
<head>
   <title>Redirecting a Request to Another Site</title>
</head>

<body>
   <p>Click a link to be redirected to the appropriate page</p>
   <p>
   <a href = "RedirectServlet?page=google">www.google.com</a>
   <br/>
   <a href = "RedirectServlet?page=RedirectServlet">
    Welcome servlet
   </a>
   </p>
</body>
</html>
```

Redirecting Requests to Other Resources



Welcome Files

- Welcome files to be loaded when the request URL is not mapped to a servlet.
- These files are typically HTML or JSP.
- Welcome files are defined using the **welcome-file-list**.
- **welcome-file-list** contains one or more **welcome-file**.
- The following **welcome-file-list** element indicates that index.html and index.htm.

```
<welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
</welcome-file-list>
```

Welcome Files

```
<!-- index.html -->
<html>
<head>
   <title>Welcome File</title>
</head>

<body>
   <p>Click a link to test each example demonstrated</p>
   <p>
   <a href = "WelcomeServlet.html">WelcomeServlet</a><br/>
   <a href = "WelcomeServlet2.html">WelcomeServlet2</a><br/>
   <a href = "WelcomeServlet3.html">WelcomeServlet3</a><br/>
   <a href = "RedirectServlet.html">RedirectServlet</a><br/>
   <a href = "Survey.html">SurveyServlet</a><br/>
   </p>
</body>
</html>
```

Welcome Files