



# Web development with Servlets & JSP

**Bertalan Fodor**  
**[Bertalan\\_Fodor@epam.com](mailto:Bertalan_Fodor@epam.com)**

2016

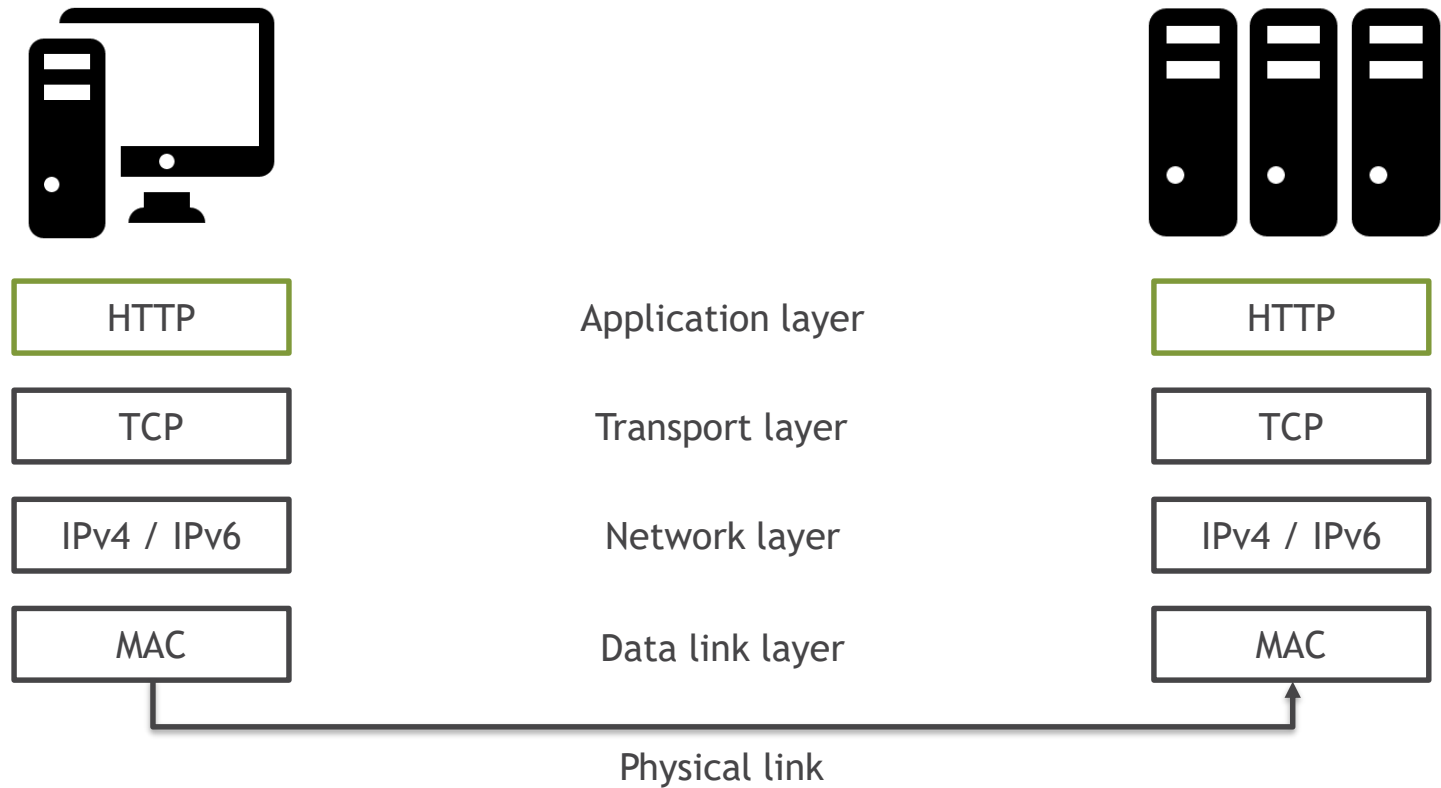




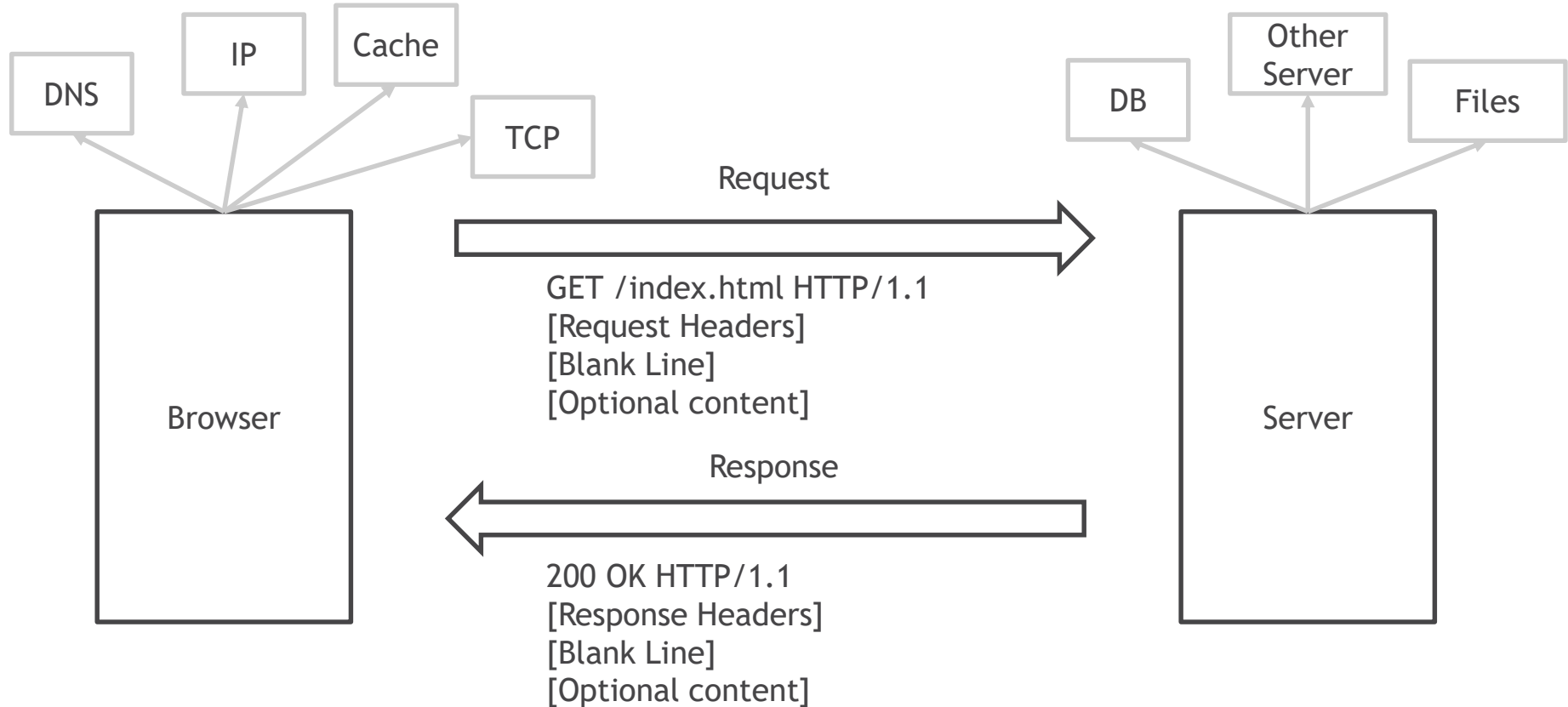
**Web basics, Servlet & JSP**

# WEB BASICS

# Lower Layers



# Browser-Server Communication



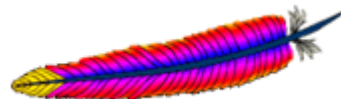
# Basic building stones of the Web

- HTTP as protocol
- HTML as document format
- CSS as formatting language
- JavaScript to run code inside the browser



# HTTP Server

- Apache HTTP Server
- Nginx
- Apache Tomcat
- Jetty
- ...



NGINX

*jetty://*



# HTTP Client



# HTML

- HTML 4.01 Strict / Transitional / Frameset

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
    "http://www.w3.org/TR/html4/strict.dtd">
```

- XHTML Strict / Transitional / Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- HTML5

```
<!DOCTYPE html>
```

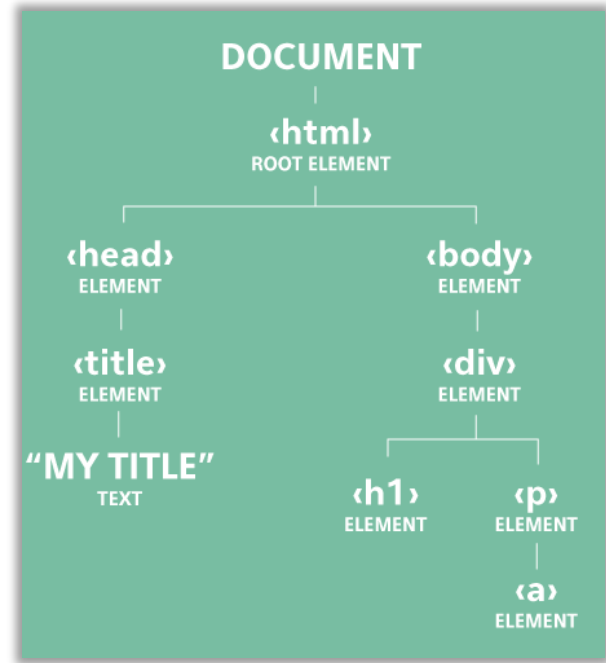


# Sample HTML5 page

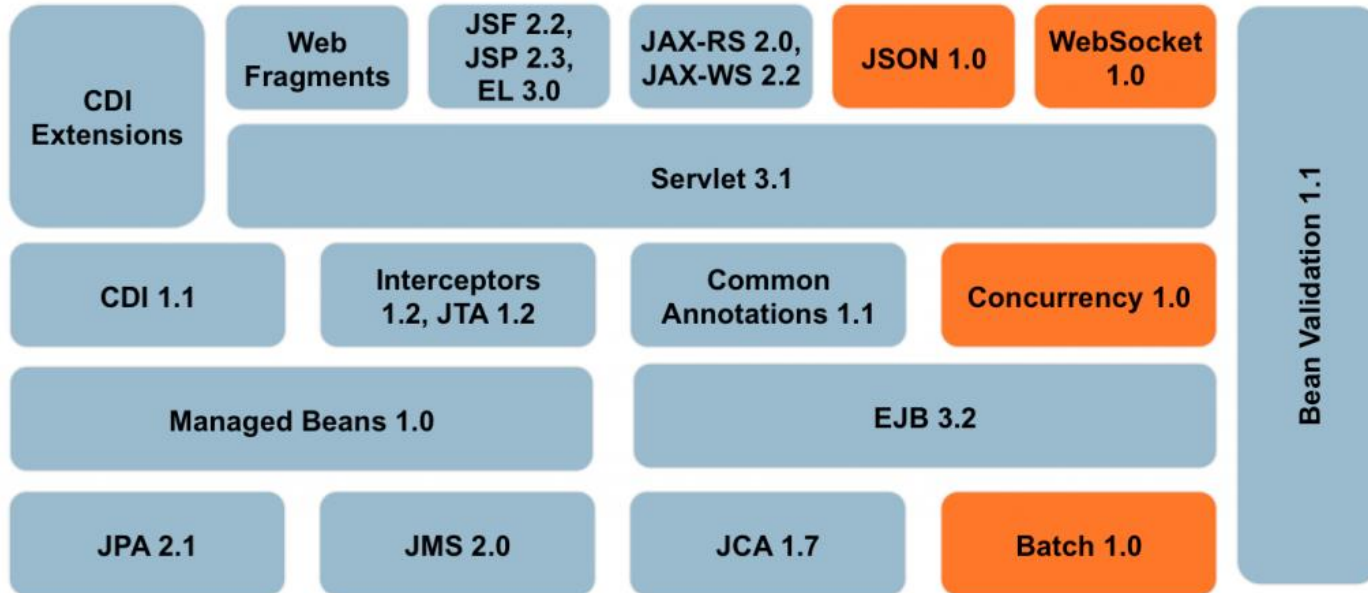
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Title of the document</title>
  </head>
  <body>
    Content of the document.....
  </body>
</html>
```

# HTML DOM

- Representation of the HTML document as loaded by the browser



# The Java EE Big Picture



A vibrant, high-contrast photograph of a busy London street, likely Regent Street, during the "golden hour" of sunset. The sun is positioned high in the background, creating a strong lens flare and casting long, warm shadows across the pavement. A large, diverse crowd of pedestrians is walking towards the camera, filling the lower half of the frame. On the left, a grand, light-colored stone building with classical architectural details, including arched windows and a balcony with a Union Jack flag, stands prominently. A street sign on this building reads "GENT STREET W1". To the right, a modern building is partially visible, featuring a red awning and a Starbucks logo. A red flag with the "Superdry" logo is also visible in the mid-ground. The overall atmosphere is one of a bustling, historic urban environment.

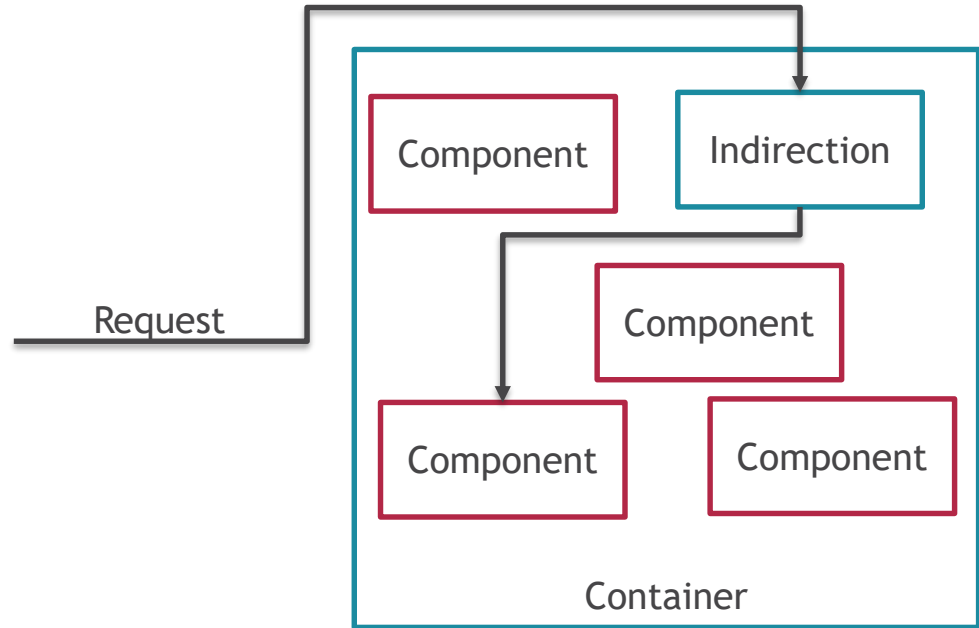
**Web basics, Servlet & JSP**

# JAVA SERVLET



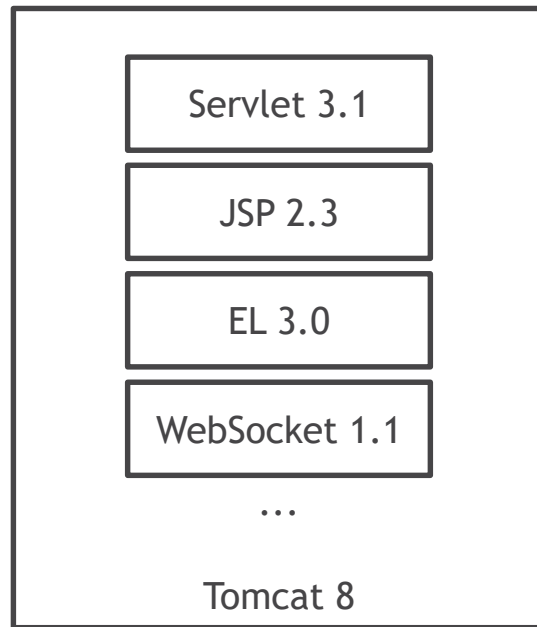
# Container model

- For cross-cutting logic
  - Communication
  - Lifecycle management
  - Thread management
  - Security



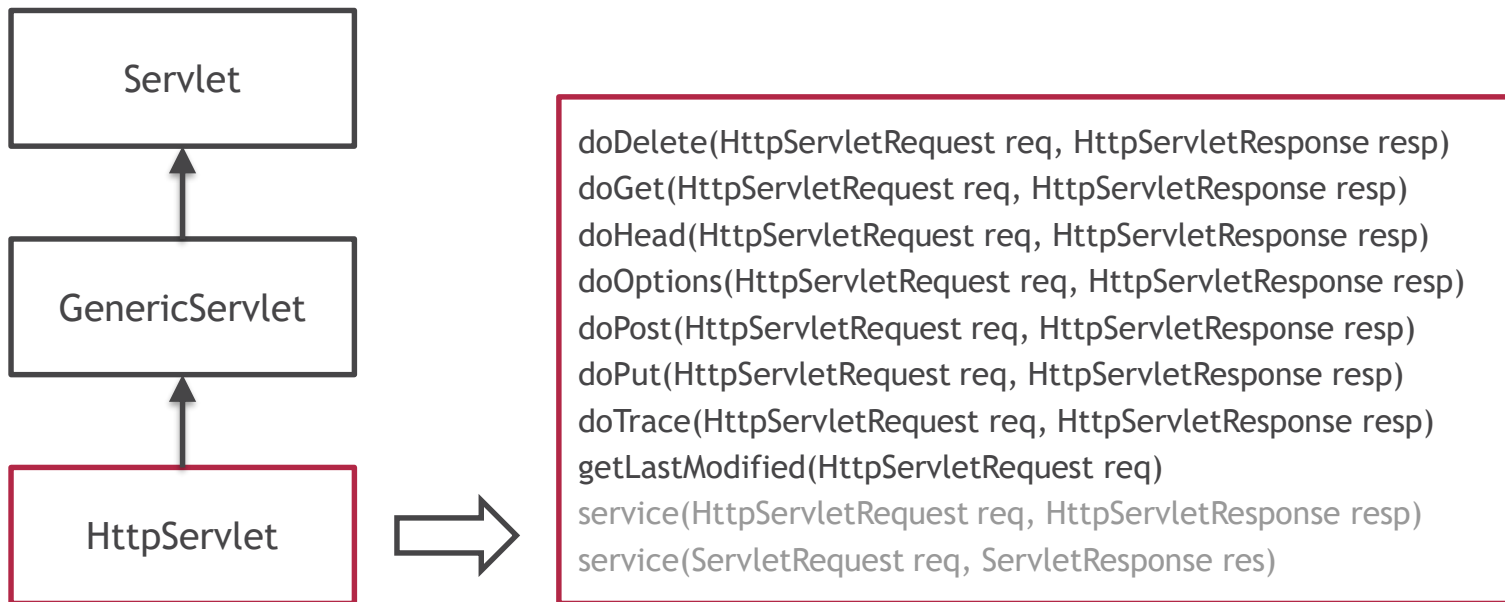
# Servlet container implementations

- Tomcat
- Jetty
- GlassFish
- WildFly
- WebLogic
- WebSphere



# Java Servlet

- Template method design pattern



# URL encoding (Percent-encoding)

- Reserved Characters

! \* ' ( ) ; : @ & = + \$ , / ? # [ ]

- Unreserved Characters

A B C D E F G H I J K L M N O P Q R S T U V W  
X Y Z a b c d e f g h i j k l m n o p q r s t  
u v w x y z 0 1 2 3 4 5 6 7 8 9 - \_ . ~

- Percent-encoding

Character	space	%	/	a	™
ASCII	%20	%25	%2F	%61	%99
UTF-8	%20	%25	%2F	%61	%E2%84



# HTML encoding

- Preferred: UTF-8 (without BOM)
- The served HTML file: **meta tag**
- HTTP response header: **Content-Type header**



# Encoding

- UTF-8 everywhere
- Content-Type
- Source files (Java, JSP)
- Request / Response



# Java Servlet deployment descriptor

```
<servlet>
  <servlet-name>simple</servlet-name>
  <servlet-class>com.epam.training.SimpleServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>simple</servlet-name>
  <url-pattern>/simple</url-pattern>
</servlet-mapping>
```

```
graph TD
    request[request] --> urlPattern["<url-pattern>/simple</url-pattern>"]
    subgraph descriptor [ ]
        direction TB
        servletName["<servlet-name>simple</servlet-name>"]
        servletClass["<servlet-class>com.epam.training.SimpleServlet</servlet-class>"]
        urlPattern
    end
    servletClass --> SimpleServlet[SimpleServlet]
```

SimpleServlet

# Specification of Mappings

- *A string beginning with a '/' character and ending with a '/\*' suffix is used for path mapping.*
- *A string beginning with a '\*. ' prefix is used as an extension mapping.*
- *The empty string ("") is a special URL pattern that exactly maps to the application's context root, i.e., requests of the form http://host:port/<contextroot>/. In this case the path info is '/' and the servlet path and context path is empty string ("").*
- *A string containing only the '/' character indicates the "default" servlet of the application. In this case the servlet path is the request URI minus the context path and the path info is null.*
- *All other strings are used for exact matches only.*

From Java™ Servlet Specification Version 3.0

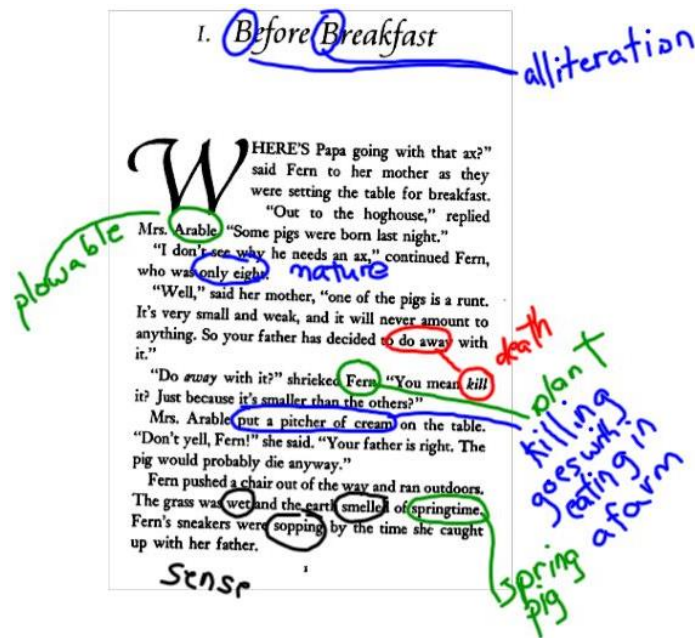


# Deployment descriptor (web.xml)

- context-param
- description
- display-name
- distributable
- ejb-ref
- ejb-local-ref
- env-entry
- error-page
- filter
- filter-mapping
- icon
- jsp-config
- listener
- login-config
- mime-mapping
- resource-env-ref
- resource-ref
- security-constraint
- security-role
- servlet
- servlet-mapping
- session-config
- web-app
- welcome-file-list

## Annotation configuration

- @WebServlet
- @WebListener
- @WebInitParam
- @ServletSecurity
  - @HttpMethodConstraint
  - @HttpConstraint
- @MultipartConfig
- @WebFilter



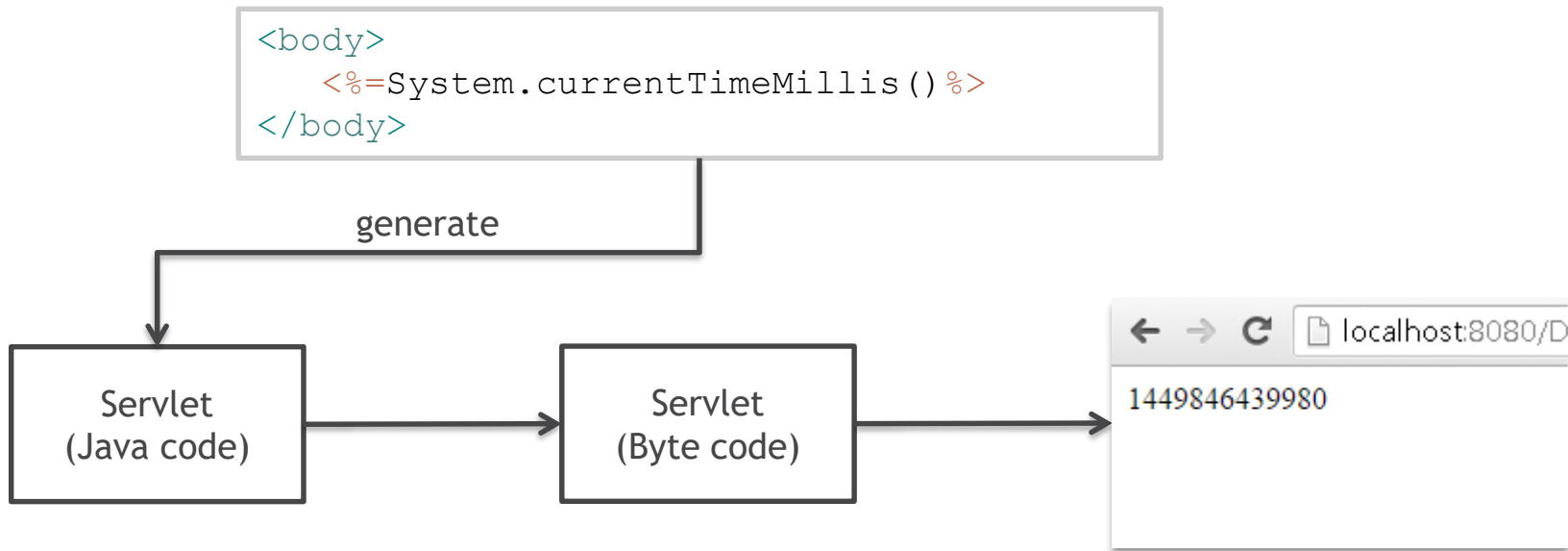
A vibrant street scene in London during the "golden hour" of sunset. The sun is low in the sky, casting a warm, golden glow over the scene and creating long shadows. A large, diverse crowd of pedestrians is walking across the street and along the sidewalks. On the left, a grand, light-colored stone building with classical architectural features like arched windows and a balcony with a Union Jack flag is visible. A street sign on this building reads "GENT STREET W1". Further down the street, a red flag with the "Superdry" logo is visible. On the right, a Starbucks logo is seen on a building facade, and a red awning extends over a storefront. The overall atmosphere is one of a bustling, historic urban environment.

**Web basics, Servlet & JSP**

# JAVASERVER PAGES

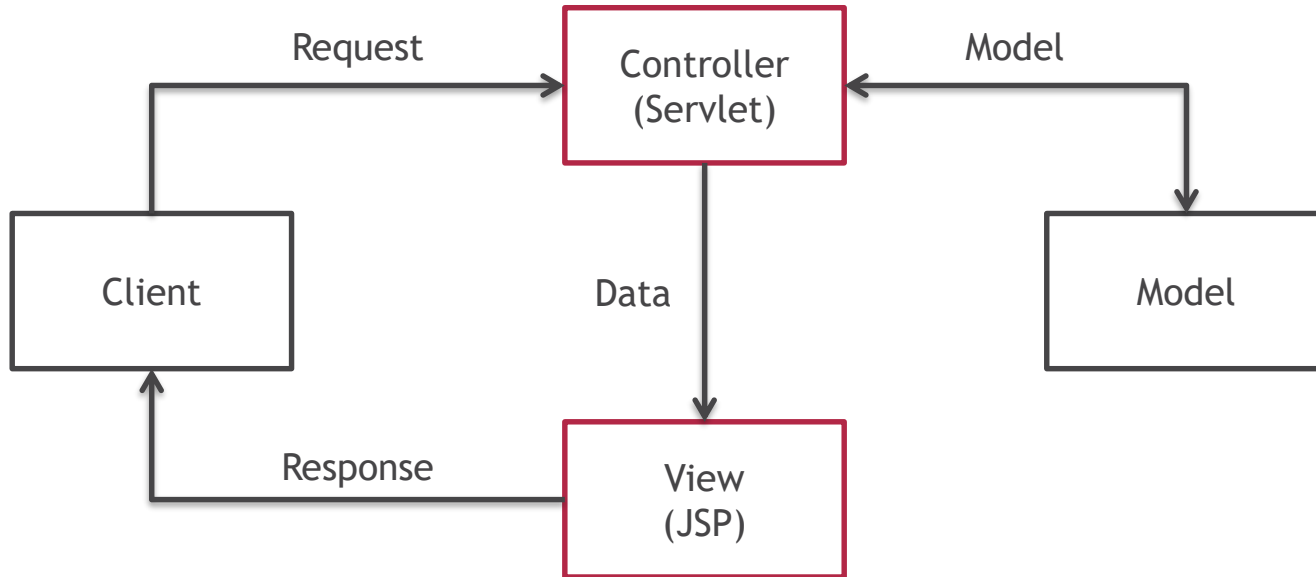
# JavaServer Pages

- Embed Java into HTML





# MVC pattern



# JSP elements

- Text

```
<!DOCTYPE html>
```

```
out.write("<!DOCTYPE html>\r\n");
```

- Scriptlet

```
<% int a; %>
```

```
int a;
```

```
<%! int a; %>
```

```
private int a;
```

```
<%= a %>
```

```
out.write(a);
```

# JSP directives

- Page

```
<%@ page import="foo.*,java.util.*" %>  
<%@ page import="static com.my.domain.MyClass.*" %>
```

- Static include

```
<%@ include file="header.jsp" %>
```

# JSP implicit objects

- |                          |             |
|--------------------------|-------------|
| • JspWriter              | out         |
| • HttpServletRequest     | request     |
| • HttpServletResponse    | response    |
| • HttpSession            | session     |
| • ServletContext         | application |
| • ServletConfig          | config      |
| • Throwable (error page) | exception   |
| • PageContext            | pageContext |
| • Object                 | page        |



**DO NOT SET  
YOURSELF  
ON FIRE**

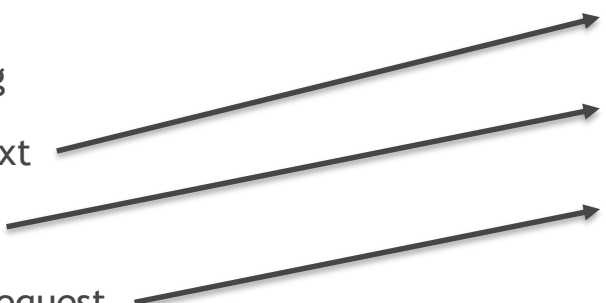
# PageContext

- Façade to reach practically everything
- Handle attributes in all scopes
- Search order: page, request, session, application

```
PageContext
  PAGE_SCOPE : int
  REQUEST_SCOPE : int
  SESSION_SCOPE : int
  APPLICATION_SCOPE : int
  PAGE : String
  PAGECONTEXT : String
  REQUEST : String
  RESPONSE : String
  CONFIG : String
  SESSION : String
  OUT : String
  APPLICATION : String
  EXCEPTION : String
  PageContext()
  initialize(Servlet, ServletRequest, ServletResponse)
  release() : void
  getSession() : HttpSession
  getPage() : Object
  getRequest() : ServletRequest
  getResponse() : ServletResponse
```

# Servlet API

- GenericServlet
- HttpServlet
- ServletConfig
- ServletContext
- HttpSession
- HttpServletRequest
- HttpServletResponse

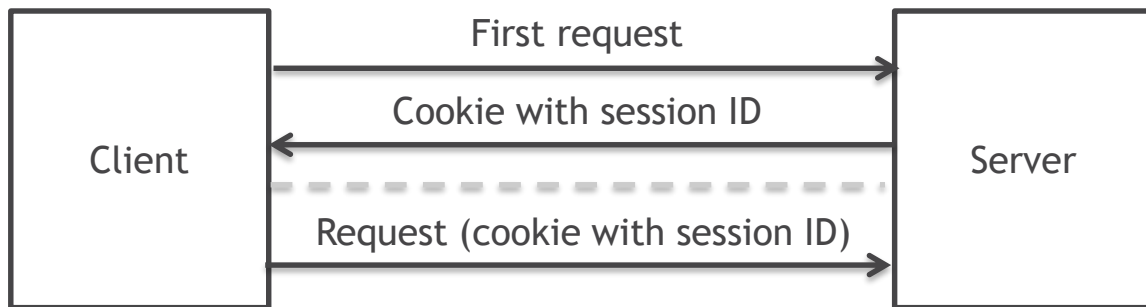


Scope	How long?
Application	Life of the deployed app
Session	User session (multiple requests)
Request	Life of the Request
Page	Within the page
Tag	Within the tag



# Session management

- Store client specific information between requests
- Identify the requests
- Store data: memory / database / external



# Init parameters, deploy time constants

```
<context-param>
  <param-name>email</param-name>
  <param-value>context@example.com</param-value>
</context-param>
<servlet>
  <servlet-name>simple</servlet-name>
  <servlet-class>com.epam.training.SimpleServlet</servlet-class>
  <init-param>
    <param-name>email</param-name>
    <param-value>servlet@example.com</param-value>
  </init-param>
</servlet>
```

```
@Override
protected void doGet(...) {
    ServletContext context = getServletContext();
    context.getInitParameter("email"); ←
    getInitParameter("email"); ←
}
```

# Script-free JSP

- Avoid at all costs: **scriptlets, expression, declaration**

- Use standard actions

```
<jsp:useBean>  
<jsp:setProperty>  
<jsp:getProperty>
```

- Use JSP Expression Language

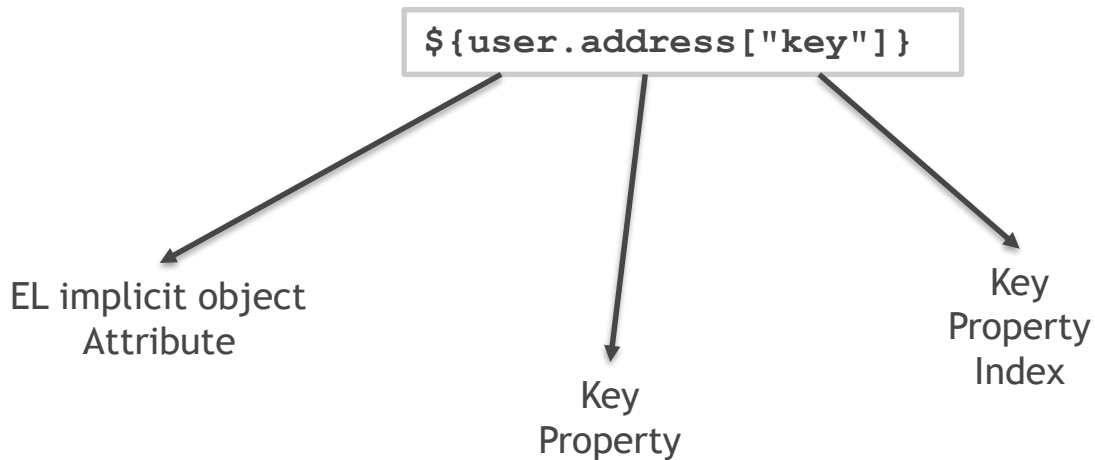
```
${user.address.city}
```

- JSP Standard Tag Library (JSTL)

```
<c:forEach var="movie" items="$movieList">  
    { $movie }  
</c:forEach>
```

# JSP Expression Language

- Read only access: properties, maps, arrays, lists
- May be disabled



# JSP Expression Language, EL operators

- Arithmetic, Logical, Relational, Conditional
- Null friendly

```
${unknown + 6 > 14}  
${unknown.something}
```

# JSP Standard Tag Library (JSTL)

- Core

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

- Format (should be avoided!)

```
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
```

- SQL (should be avoided!!!)

```
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
```



# Core Tag Library

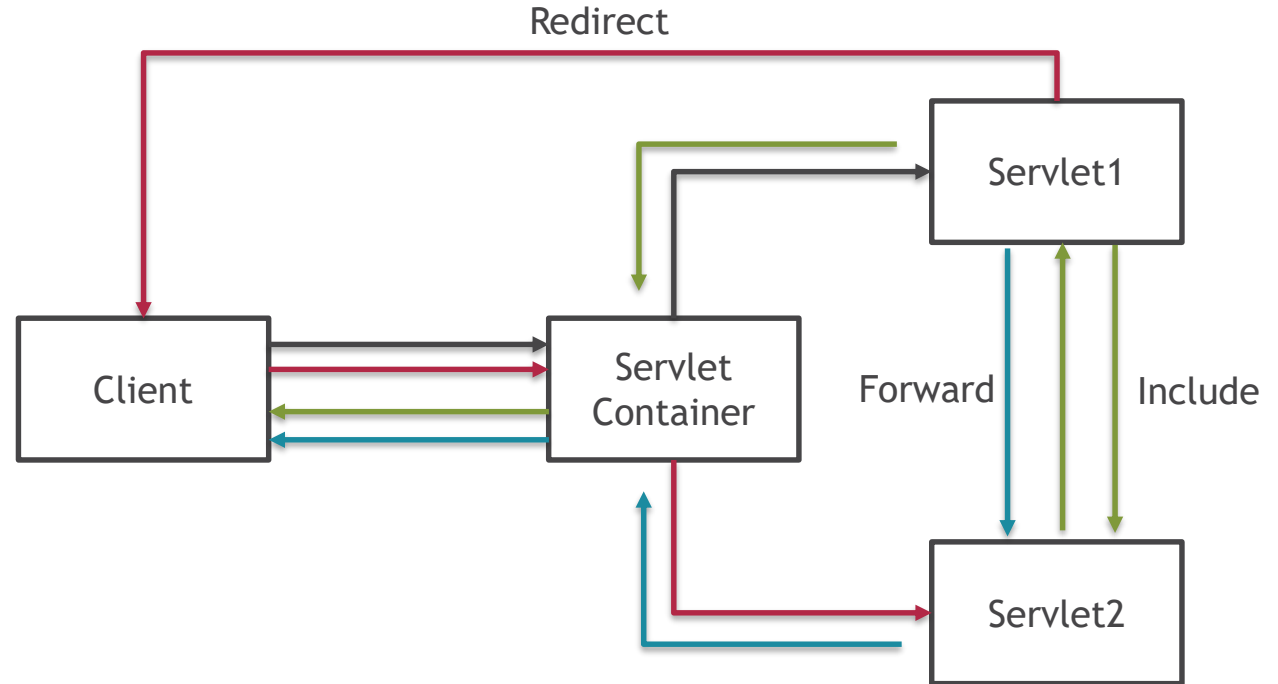
- `<c:out >`
- `<c:set >`
- `<c:remove >`
- `<c:catch>`
- `<c:if>`
- `<c:choose>`
- `<c:when>`
- `<c:otherwise >`
- `<c:import>`
- `<c:forEach >`
- `<c:forTokens>`
- `<c:param>`
- `<c:redirect >`
- `<c:url>`

# JSTL Functions

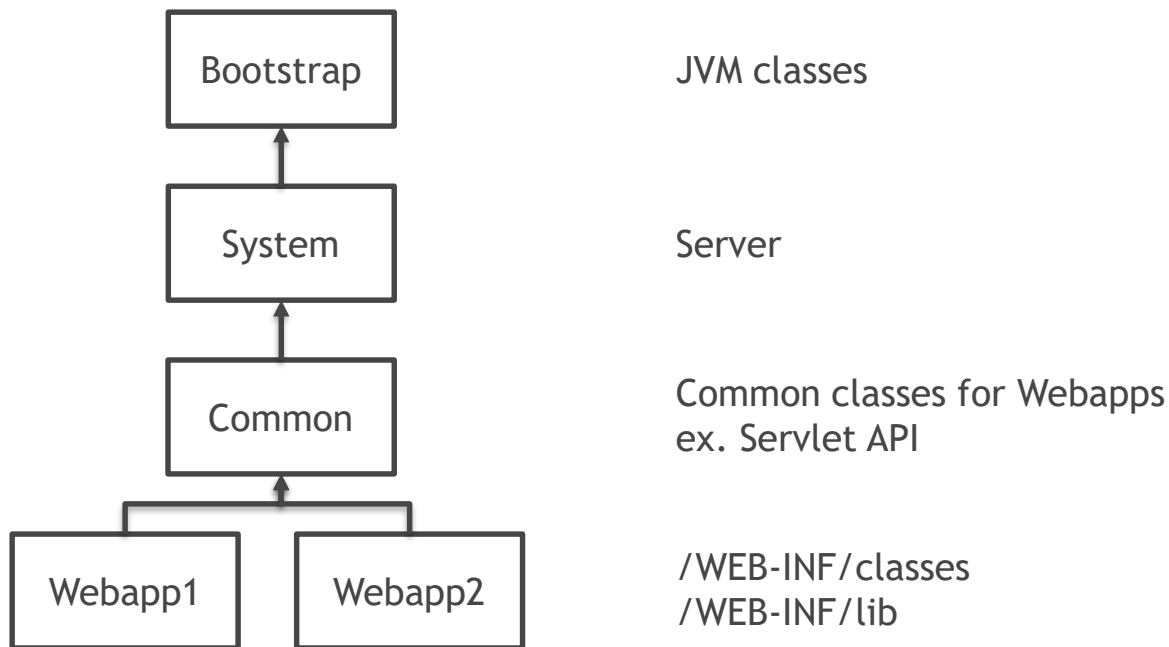
- `fn:contains()`
- `fn:containsIgnoreCase()`
- `fn:endsWith()`
- `fn:escapeXml()`
- `fn:indexOf()`
- `fn:join()`
- `fn:length()`
- `fn:replace()`
- `fn:split()`
- `fn:startsWith()`
- `fn:substring()`
- `fn:substringAfter()`
- `fn:substringBefore()`
- `fn:toLowerCase()`
- `fn:toUpperCase()`
- `fn:trim()`

# Redirect mechanisms

- Redirect
- Forward
- Include



# Servlet container class loader (Tomcat)





# Thank you!

**Bertalan Fodor**  
**[Bertalan\\_Fodor@epam.com](mailto:Bertalan_Fodor@epam.com)**

2016