



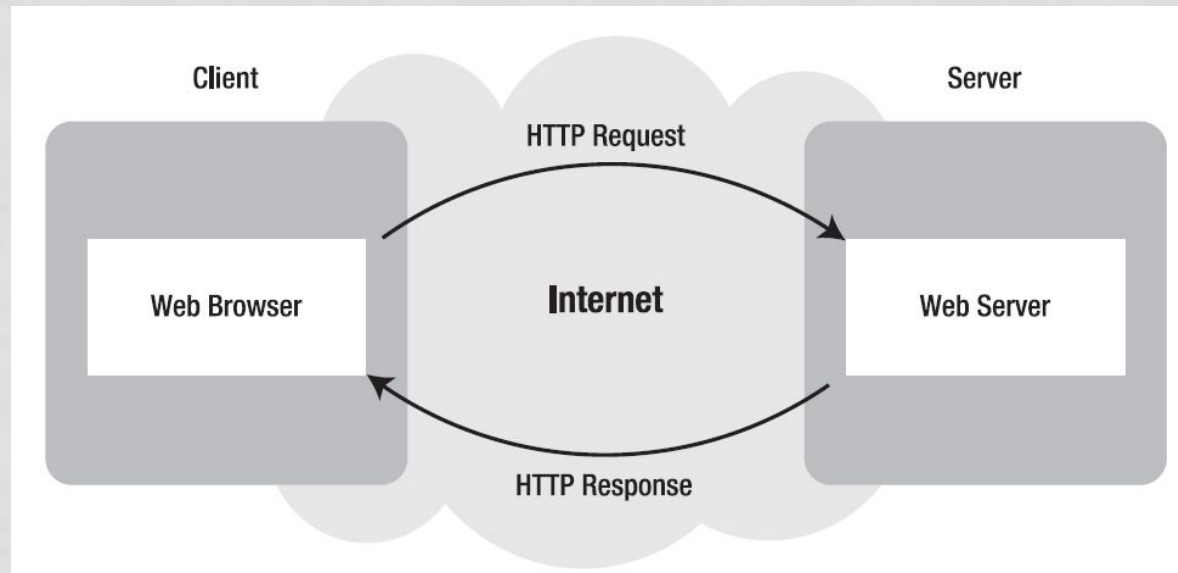
# WEB PROGRAMMING. JSP

# CLIENT-SERVER

- Client
- Server
- Protocol

# CLIENT-SERVER

- Protocol is HTTP
- Client sends HTTP request
- Server generates HTTP response
- Data are returned as document(HTML, XML, image)



# HTTP

- Hypertext transfer protocol
- Resources are identified by URI or URL
- Command consists of header and data
- Stateless protocol
- Request defines action to perform
- Status codes are returned(200, 404, 500, etc)

# HTTP REQUEST

Header	Value
Accept	text/html,application/xhtml+xml,application/xml;
Accept-language	en-gb,en
Accept-Encoding	gzip, deflate
Connection	keep-alive
Host	www.google.com.ua
User-Agent	Mozilla/5.0 (Windows NT 6.1) Firefox/20.0

# HTTP REQUEST

Parameter	Value
type	device
name	test
limit	10
order	asc

# HTTP REQUEST

Method	Description
GET	Requests representation of the resource
POST	Modifies web resource
PUT	Modifies or creates new resource
DELETE	Deletes specified resources

# HTTP RESPONSE

Header	Value
Content-Encoding	gzip
Content-Type	text/html; charset=UTF-8
Date	Tue, 23 Apr 2013 15:07:27 GMT
Expires	-1
Cache-Control	private, max-age=0
Status	200 OK

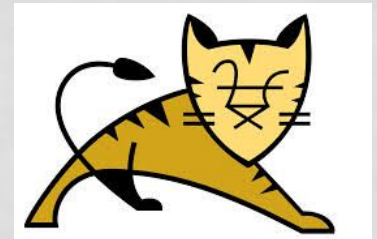
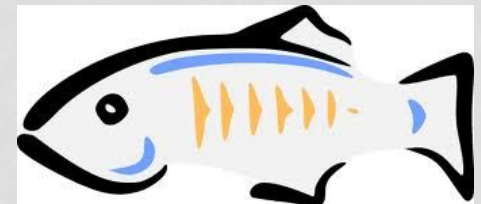


# HTTP RESPONSE

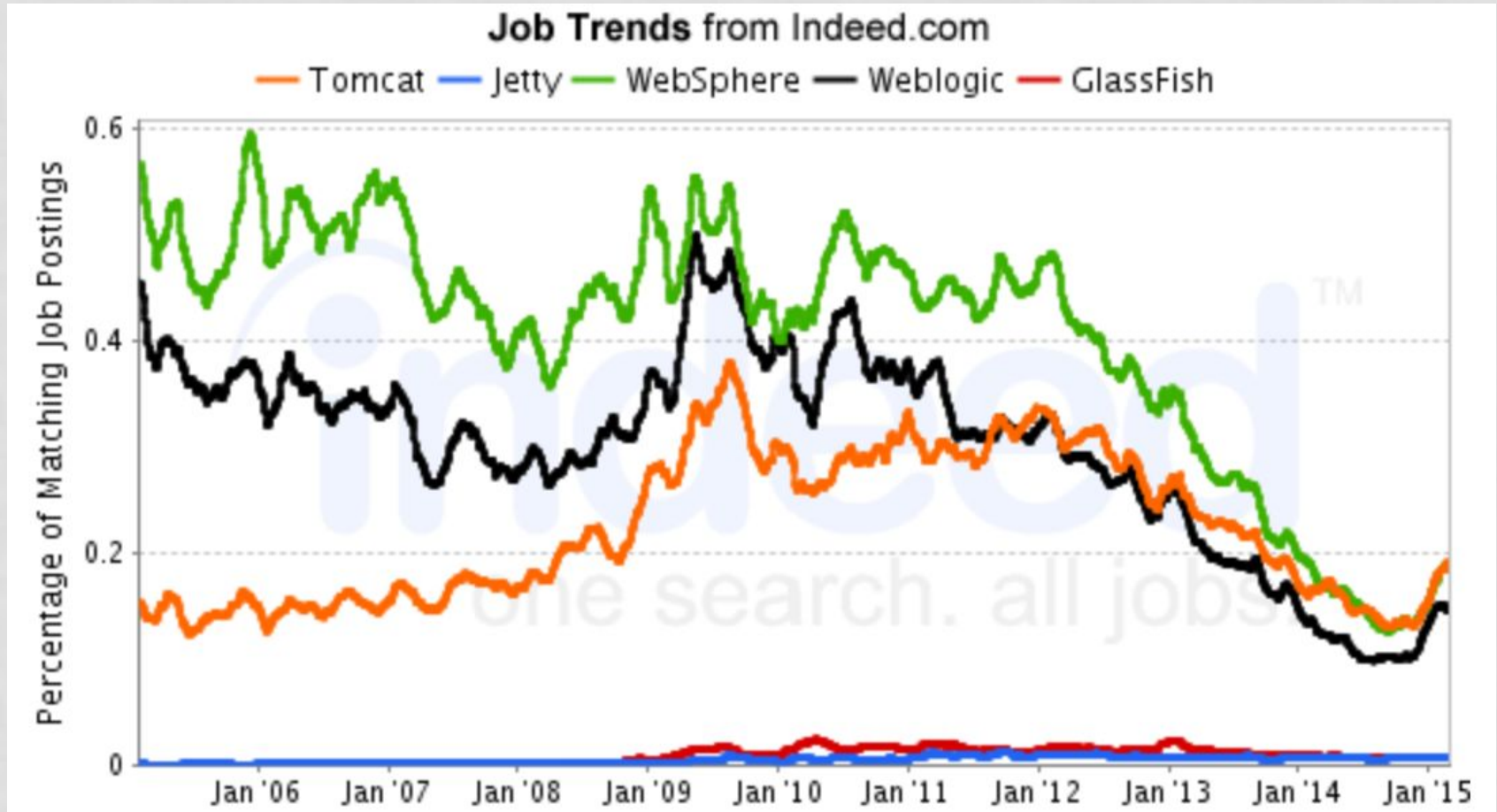
- `<!DOCTYPE html>`
- `<html>`
- `<body>`
- `<h1>My First Heading</h1>`
- `<p>My first paragraph.</p>`
- `</body>`
- `</html>`

# WEB SERVER

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



# WEB SERVER



# HTML

```
<html>  
  <body>  
    <form>  
      <input type="hidden" id="code" />  
      <textarea id="pre_code" />  
    </form>  
  </body>  
</html>
```

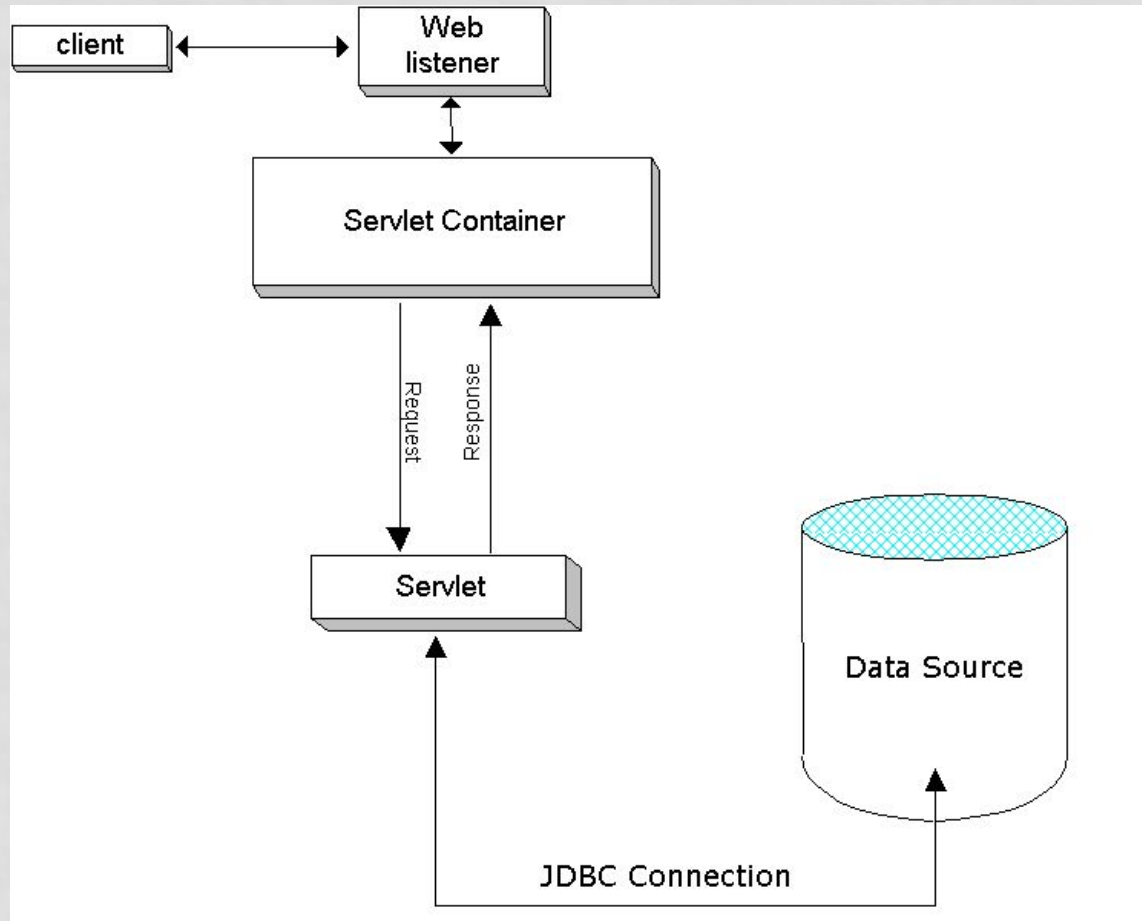
# WEB SERVER

- Java servlets
- Java server pages(JSP)
- Web sockets

# JAVA SERVLET

- Java component that runs inside web server
- Web server loads and destroys servlets
- Receives HTTP requests, generates results and sends out HTTP responses
- Part of J2EE

# JAVA SERVLET



# SERVLET CONTAINER

- Interacts with Java servlets
- Manages lifecycle of the servlet
- Maps URL to particular servlets
- Ensures security and check access rights
- Provides deployment, transaction management, concurrency and other services



# APACHE TOMCAT

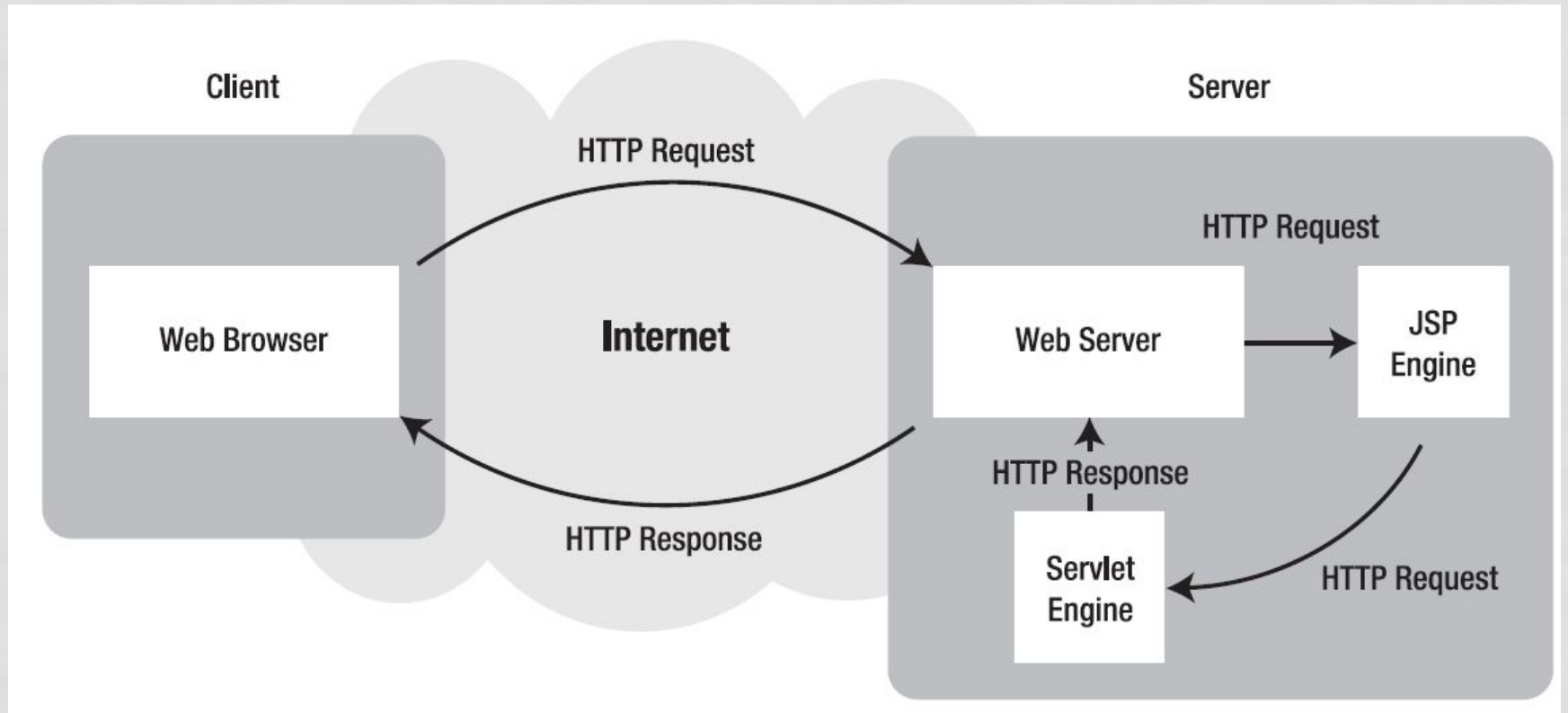
- Developed by Apache Software Foundation
- Includes web server(Coyote), servlet container(Catalina) and JSP engine(Jasper)
- Requires Java 7
- Current version is 8.0.20
- Supports Servlets, JSP, EL and Web sockets



# JSP

- Adds dynamic content to web pages
- Designed in 1999
- Introduced JSTL(JSP standard tag library)
- Introduced EL(expression language)
- Current version 2.4

# JSP



# JSP

- Client sends HTTP request to the server
- Server recognized HTTP request and forwards to JSP engine
- JSP engine loads JSP page and converts it into servlet
- JSP engine compiles servlet into executable code and forwards request to servlet engine
- Servlet engine executes servlet and produces HTML
- HTTP response with HTML is returned to client

# SAMPLE JSP

```
<%@page language="java"  
contentType="text/html"%>
```

```
<html>
```

```
  <head><title>Dynamic HTML</title></head>
```

```
  <body>
```

```
    Hello World!
```

```
  </body>
```

```
</html>
```

# SAMPLE JSP

- `<%`
- `out.println("<br/>Your IP address is " + request.getRemoteAddr());`
- `String userAgent = request.getHeader("user-agent");`
- `out.println("<br/>" + userAgent);`
- `%>`

# JAVA BEANS

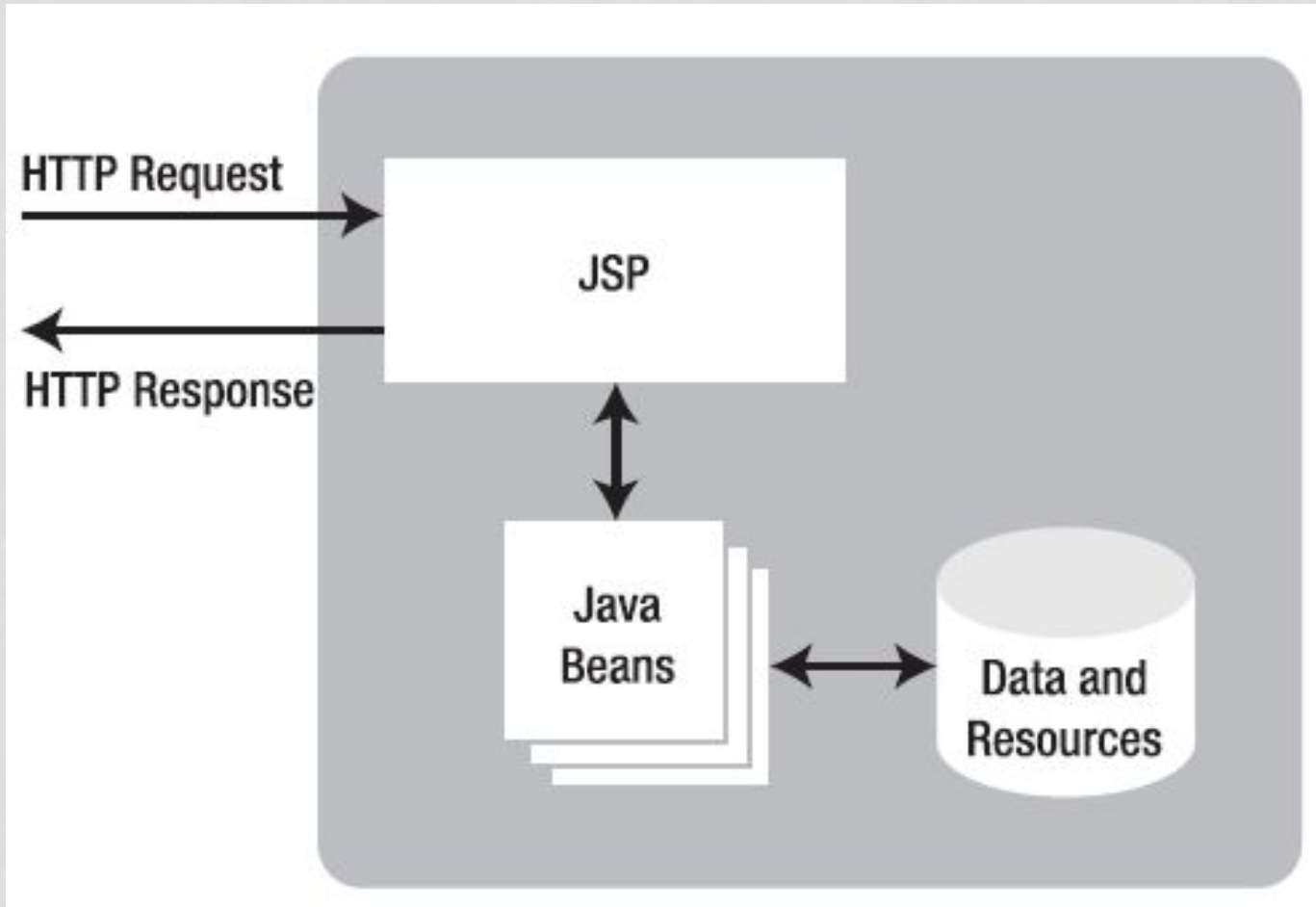
- Plain Java objects
- No-argument public constructors
- Setter and getter for each field
- May receive or generate events
- Serializable

# JAVA BEANS

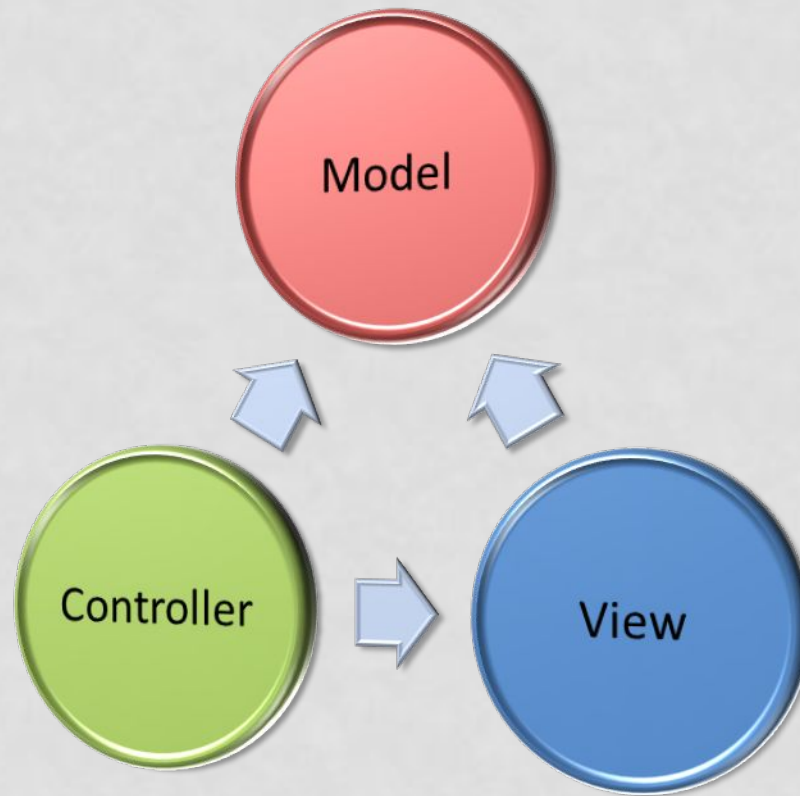
- public class Product implements **Serializable** {
- private int id;
- public **Product**() {}
- public int **getId**() {
- return id;
- }
- public void setId(int id) {
- this.id = id; }
- }



# JSP ARCHITECTURE



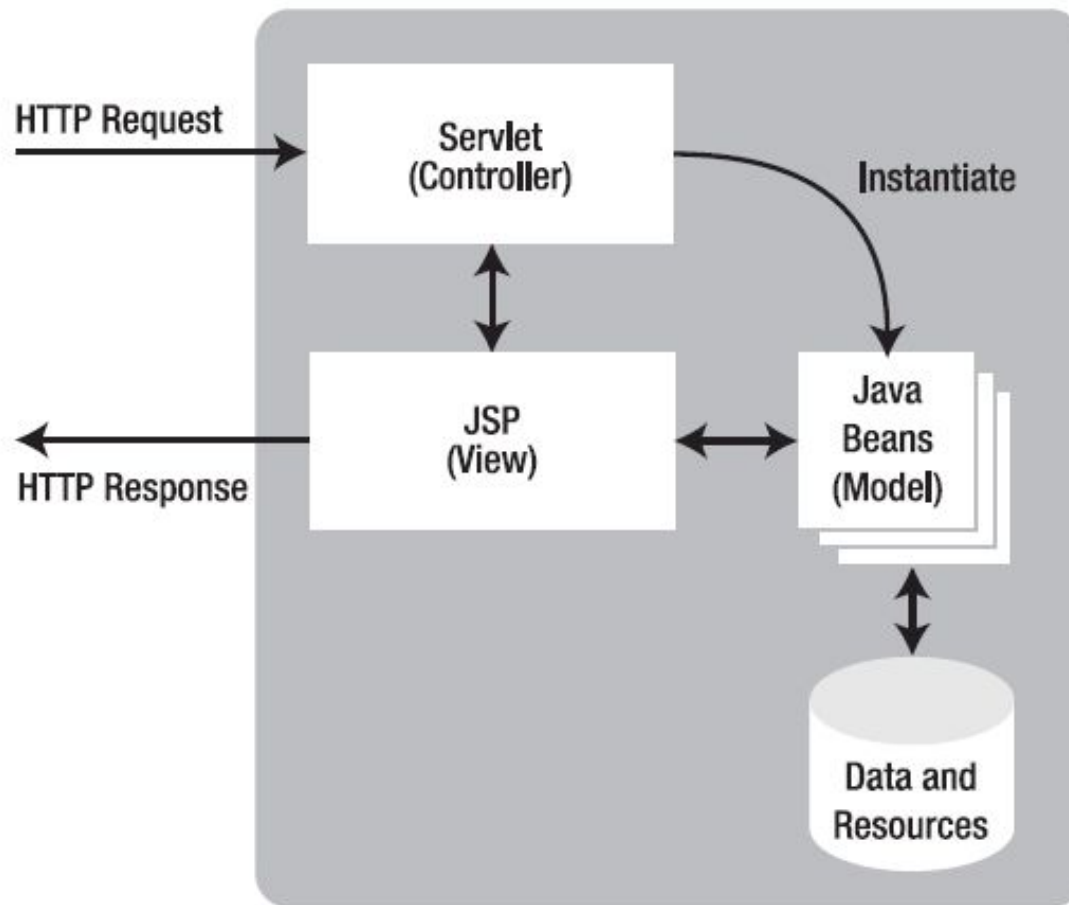
# MVC



# MVC

- **Model** contains application data and business rules
- **View** contains representation of the data
- **Controller** sends commands to view and updates model

# JSP ARCHITECTURE



# JSP VARIABLES

- application
- config
- out
- pageContext
- request
- response
- session

# HTTPServletRequest

- String **getParameter**(String name)
- Enumeration<String> getParameterNames()
- String[] **getParameterValues**(String name)
- String getServerName()
- String **getRemoteAddr**()
- String getHeader(String name);
- String **getMethod**();
- String getContextPath();
- String **getQueryString**();

# HTTPSERVLETRESPONSE

- `encodeURL(String)`
- `sendRedirect(String)`
- `getHeader(String)`
- `getContentType()`
- `getOutputStream()`
- `getCharacterEncoding()`

# HTTP SESSION

- Provides way to identify a user who requests web server
- Identifies user
- Has time limitations
- Has a cookie assigned(JSESSIONID)
- Used to store attributes



# HTTPSESSION

- `getAttribute(String)`
- `getId()`
- `getServletContext()`
- `invalidate()`
- `removeAttribute(String)`
- `setAttribute(String, Object)`

# DIRECTIVES

- Provides web server with information it needs to handle JSP request
- Executes before compilation
- page
- include
- taglib

# PAGE DIRECTIVE

- `<%@page language="java" contentType="text/html"%>`
- `<%@page import="java.util.ArrayList"%>`
- `<%@page import="java.util.*" %>`
- `<%@page import="org.hillel.it.model.Product, org.hillel.it.model.Order" %>`

# INCLUDE DIRECTIVE

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

# ACTIONS

- Executing while processing HTTP request
- forward
- include
- param
- useBean
- getProperty
- text
- setProperty

# FORWARD ACTION

- **<jsp:forward** page="login.jsp">
- **<jsp:param** name="reason" value="nosession"/>
- **</jsp:forward>**

# INCLUDE ACTION

- `<jsp:include page="header.jsp"/>`

# USEBEAN ACTION

- **<jsp:useBean** id="service" scope="application"
- class="org.hillel.it.service.ServiceImpl" />
- <% int id =  
Integer.valueOf(request.getParameter("id"));
- Product product = service.getProduct(id) %>



# SESSION SCOPE

- page
- request
- session
- application