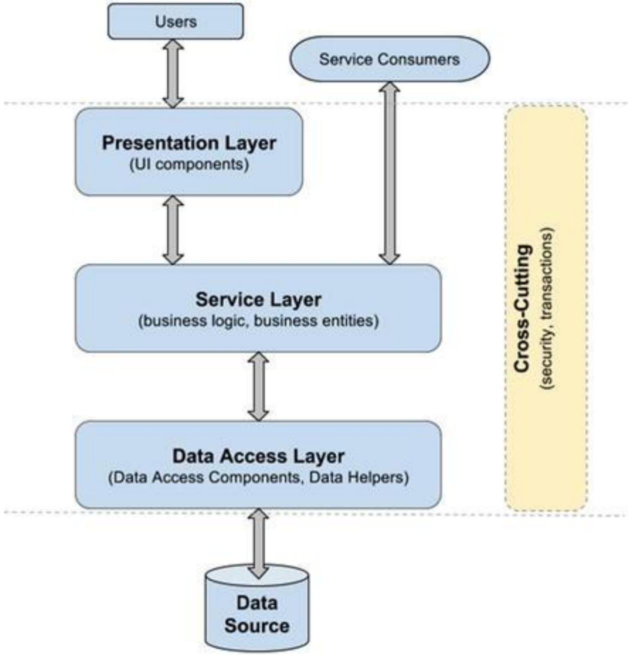
Servlet, JSP, MVC

Web Application Architecture



HTTP Request / Response

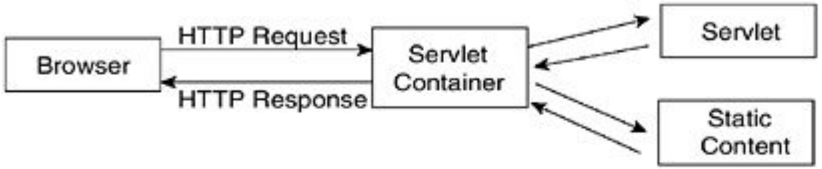
```
GET /servlet/default.jsp HTTP/1.1
Accept: text/plain; text/html
Accept-Language: en-gb
Connection: Keep-Alive
Host: localhost
Referer: http://localhost/ch8/SendDetails.htm
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
Content-Length: 33
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate

LastName=Franks&FirstName=Michael
```

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/4.0
Date: Mon, 3 Jan 1998 13:13:33 GMT
Content-Type: text/html
Last-Modified: Mon, 11 Jan 1998 13:23:42 GMT
Content-Length: 112

<HTML>
<HEAD>
<TITLE>HTTP Response Example</TITLE></HEAD><BODY>
Welcome to Saga Course
</BODY>
</HTML>
```

HTTP Verbs

Method	**Description**
GET	GET is the simplest, and probably, most used HTTP method. GET simply retrieves the data identified by the URL. If the URL refers to a script (CGI, servlet, and so on), it returns the data produced by the script.
HEAD	The HEAD method provides the same functionality as GET, but HEAD only returns HTTP headers without the document body.
POST	Like GET, POST is also widely used. Typically, POST is used in HTML forms. POST is used to transfer a block of data to the server in the entity body of the request.
OPTIONS	The OPTIONS method is used to query a server about the capabilities it provides. Queries can be general or specific to a particular resource.
PUT	The PUT method is a complement of a GET request, and PUT stores the entity body at the location specified by the URI. It is similar to the PUT function in FTP.
DELETE	The DELETE method is used to delete a document from the server. The document to be deleted is indicated in the URI section of the request.
TRACE	The TRACE method is used to tract the path of a request through firewall and multiple proxy servers. TRACE is useful for debugging complex network problems and is similar to the traceroute tool.

Servlet Container Simplified



Servlet

- Java Servlets are server-side Java program modules that process and answer client requests and implement the servlet interface

```
@WebServlet(value="/servlet")
public class ServletResponse extends HttpServlet {

  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response) {
        int id = Integer.valueOf(request.getParameter("id"));
        Player player = new PlayerDAO().findPlayer(id);
        response.setContentType("text/html; charset=ISO-8859-1");
        Writer out =  response.getWriter();
        out.write("\r\n");
        out.write("\r\n");
        out.write("<html>\r\n");
        out.write("<head>\r\n"); // ommiting a lot of lines
        out.write("\r\n");
        out.write("</body>\r\n");
        out.write("</html>");
       }
 }
```

JSP

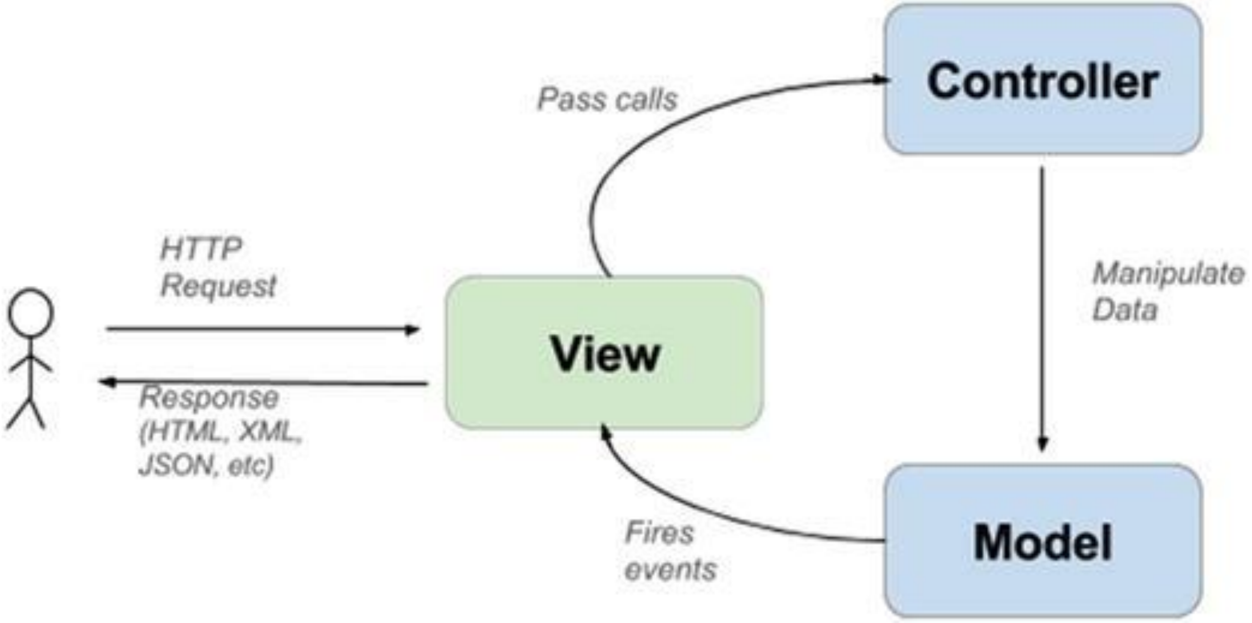
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1" %>
<%@ page import="java.sql.*" %>
<%@ page import="rs.saga.mvc.model.PlayerDAO" %>
<%@ page import="rs.saga.mvc.model.Player" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
   <title>Example</title>
</head>
<body>
       <%
          PlayerDAO playerDAO = new PlayerDAO();
          Player player = playerDAO.findPlayer(Integer.valueOf(request.getParameter("id")));
          request.setAttribute("firstName", player.getFirstName());
       %>
<h1>Hello ${requestScope.firstName}</h1>

</body>
</html>
```

Web Application Directory Structure

- Standard Directory Structure

```
src -|
     main -|
            java
            resources
            webapp -|
                    jsp
                    WEB-INF
```

MVC


Controller
Pass calls
HTTP Request
Manipulate Data
View
Response (HTML, XML, JSON, etc)
Model
Fires events

MVC Applied

// Servlet

```java
public class HelloWorldServlet extends HttpServlet {
  @Override
  protected void doGet(HttpServletRequest req, HttpServletResponse resp) {
     int id = Integer.valueOf(req.getParameter("id"));
     req.setAttribute("player", new PlayerDAO().findPlayer(id));
     req.getRequestDispatcher("/WEB-INF/hello.jsp").forward(req, resp);
  }
}
```

// JSP

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!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>Player detail</title>
</head>
<body>
  <h1>Hello ${requestScope.player.firstName}</h1>
</body>
</html>
```