**MVC ARCHITECTURE :**

**MVC** stands for Model View and Controller. It is a **design pattern** that separates the business logic, presentation logic and data.

**Controller** acts as an interface between View and Model. Controller intercepts all the incoming requests.

**Model** represents the state of the application i.e. data. It can also have business logic.

**View** represents the presentaion i.e. UI(User Interface).

Advantage of MVC (Model 2) Architecture

1. Navigation Control is centralized
2. Easy to maintain the large application



**HTTP Session :**

In such case, container creates a session id for each user.The container uses this id to identify the particular user.An object of HttpSession can be used to perform two tasks:

1. bind objects
2. view and manipulate information about a session, such as the session identifier, creation time, and last accessed time.



How to get the HttpSession object ?

The HttpServletRequest interface provides two methods to get the object of HttpSession:

1. **public HttpSession getSession():**Returns the current session associated with this request, or if the request does not have a session, creates one.
2. **public HttpSession getSession(boolean create):**Returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session.

**Expression Language :**

The **Expression Language** (EL) simplifies the accessibility of data stored in the Java Bean component, and other objects like request, session, application etc.

There are many implicit objects, operators and reserve words in EL.

It is the newly added feature in JSP technology version 2.0.

Syntax for Expression Language (EL)

${ expression }

There are many implicit objects in the Expression Language. They are as follows:

|  |  |
| --- | --- |
| **Implicit Objects** | **Usage** |
| pageScope | it maps the given attribute name with the value set in the page scope |
| requestScope | it maps the given attribute name with the value set in the request scope |
| sessionScope | it maps the given attribute name with the value set in the session scope |
| applicationScope | it maps the given attribute name with the value set in the application scope |
| param | it maps the request parameter to the single value |
| paramValues | it maps the request parameter to an array of values |
| header | it maps the request header name to the single value |
| headerValues | it maps the request header name to an array of values |
| cookie | it maps the given cookie name to the cookie value |
| initParam | it maps the initialization parameter |
| pageContext | it provides access to many objects request, session etc. |

**<%@page directive> :**

* SP directives are the messages to JSP container. They provide global information about an entire JSP page.
* JSP directives are used to give special instruction to a container for translation of JSP to servlet code.
* In JSP life cycle phase, JSP has to be converted to a servlet which is the translation phase.
* They give instructions to the container on how to handle certain aspects of JSP processing
* Directives can have many attributes by comma separated as key-value pairs.
* In JSP, directive is described in <%@ %> tags.

**Syntax of Directive:**

<%@ directive attribute="" %>

**JSP Page directive**

**Syntax of Page directive:**

<%@ page…%>

* It provides attributes that get applied to entire JSP page.
* It defines page dependent attributes, such as scripting language, error page, and buffering requirements.
* It is used to provide instructions to a container that pertains to current JSP page.

Following are its list of attributes associated with page directive:

1. Language
2. Extends
3. Import
4. contentType
5. info
6. session
7. isThreadSafe
8. autoflush
9. buffer
10. IsErrorPage
11. pageEncoding
12. errorPage
13. isELIgonored