Advanced Java

Trainer: Nilesh Ghule

JSP

JSP is outdated.

Servlets = business logic* + presentation logic

JSPs = presentation logic* + business logic

JSP is translated into servlet while execution.

① JSP syntax (other than markup)

- @ <%@ .... %> → directives
  - ↳ page, include, taglib
- ⓑ <%! ... %> → declaration
  - ↳ jspInit(), jspDestroy(), fields, ...
- ⓒ <% .... %> → scriptlet
  - ↳ java statements - req. handling
- ⓓ <%= .... %> → expression
  - ↳ java expression - req. handling
  - ↳ output added to resp directly.
- ⓔ <%-- ... --%> → server side comment

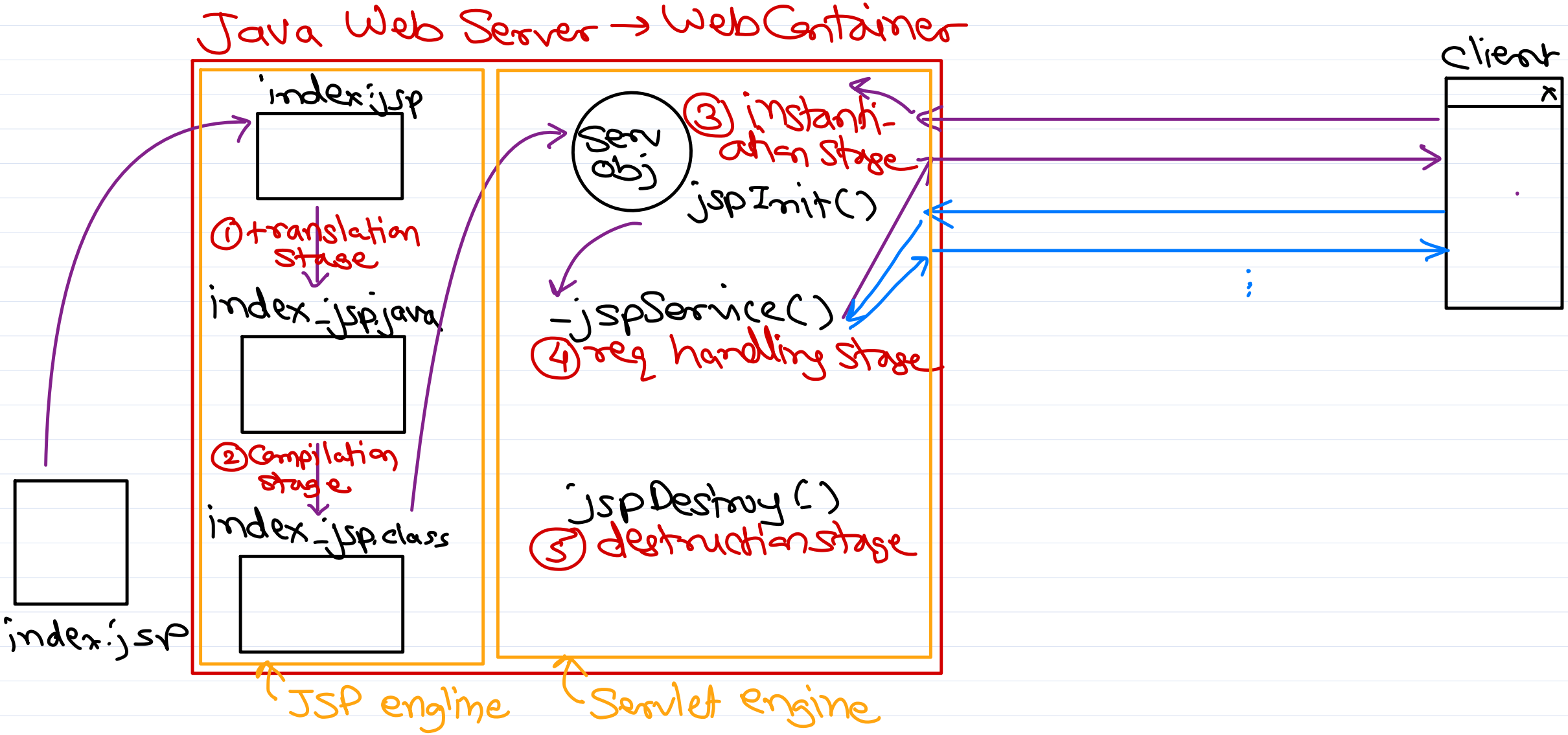
② JSP implicit objects

variables available in req handling stage.

internally - local vars of _jspService() method.

- @ request : HttpServletRequest
- ⓑ response : HttpServletResponse
- ⓒ config : ServletConfig
- ⓓ session : HttpSession
- ⓔ application : ServletContext
- ⓕ page : Object (this pointer)
- ⓖ pageContext : PageContext
- ⓗ out : JspWriter
- ⓘ exception : Throwables
  - avail only in err pages.

JSP Life Cycle



JSP file should have presentation logic i.e. markup syntax (tags).

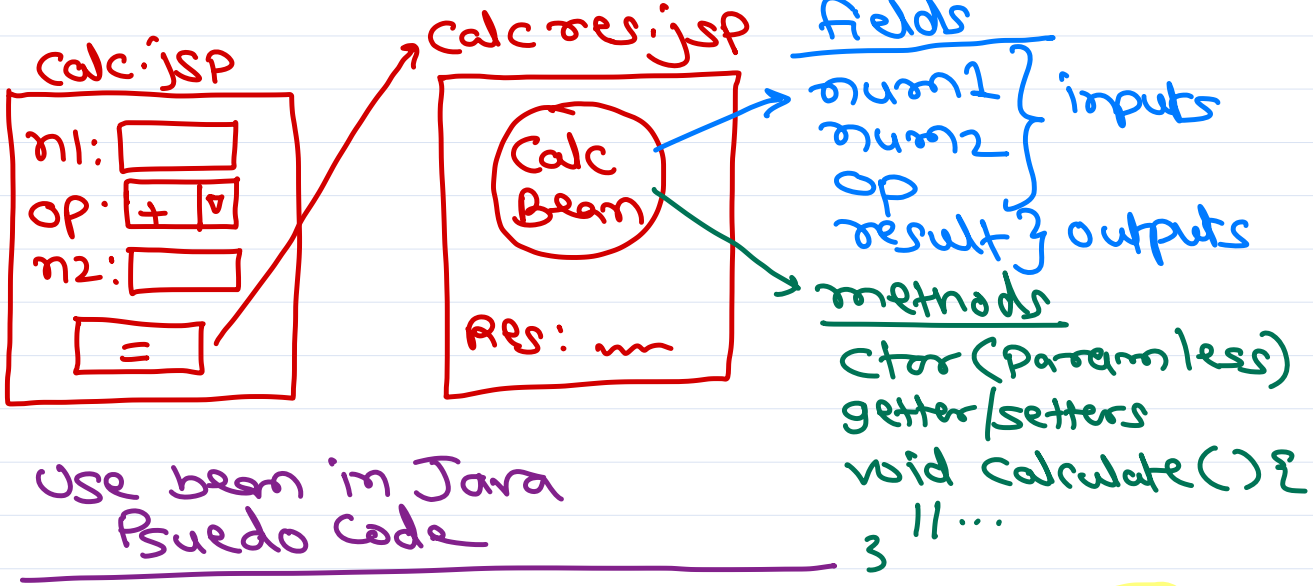
Ideal JSP do not have any java code in it (i.e. scriptlets or expression).

To reduce scriptlets from JSP pages:

1. JSP standard actions
2. Java beans
3. JSP EL
4. JSTL tags
5. Custom tags

Java Beans

Java Bean = fields + ctor + getter/setter + business logic method(s).



Use bean in Jsp
Pseudo code

```
CalcBean cb = new CalcBean();
cb.setNum1(req.getParameter("num1"));
cb.setNum2(req.getParameter("num2"));
cb.setOp(req.getParameter("op"));
cb.calculate();
println(cb.getResult());
```

```
<jsp:useBean id="cb"
        class="pkg.CalcBean" />
<jsp:setProperty name="cb"
 property="num1" param="num1" />
<jsp:setProperty name="cb"
 property="num2" param="num2" />
<jsp:setProperty name="cb"
 property="op" param="op" />
<% cb.calculate(); %>
<jsp:getProperty name="cb"
        property="result" />
```

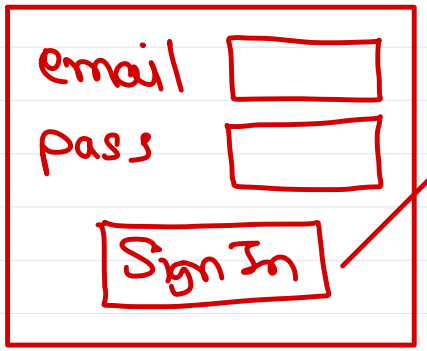
property = "op" : field name
param = "op" : req param name

Java Beans

fields: email, password, user
method:
authenticate() {
//jdbc code
or call dao.
}

Login Bean

demo6.jsp



login bean.jsp

```
<jsp:useBean id="lb" class="pkg.LoginBean"/>
    // LoginBean lb = new LoginBean();
<jsp:setProperty name="lb" property="email" param="email"/>
    // lb.setEmail(request.getParameter("email"));
<jsp:setProperty name="lb" property="password" param="passwd"/>
    // lb.setPassword(request.getParameter("passwd"));
<% lb.authenticate(); %>
<% if(lb.getUser() != null) { %>
    Welcome, <jsp:getProperty name="lb" property="email"/>
<% } else { %>
    Failed
<% } %>
```

// lb.getEmail() → o/p in resp.

Java Bean Scopes

Scope	attribute of
① page (lowest) ← default scope	→ PageContext → accessible in cur req to cur page only.
② request	→ HttpServletRequest → accessible on all pages where cur req is fwd or incl.
③ session	→ HttpSession → accessible in all req to all pages for cur user.
④ application (highest)	→ ServletContext → accessible in all req to all pages for all users

```
<jsp:useBean id="lb"
  class="pkg.LoginBean" scope="session" />
```

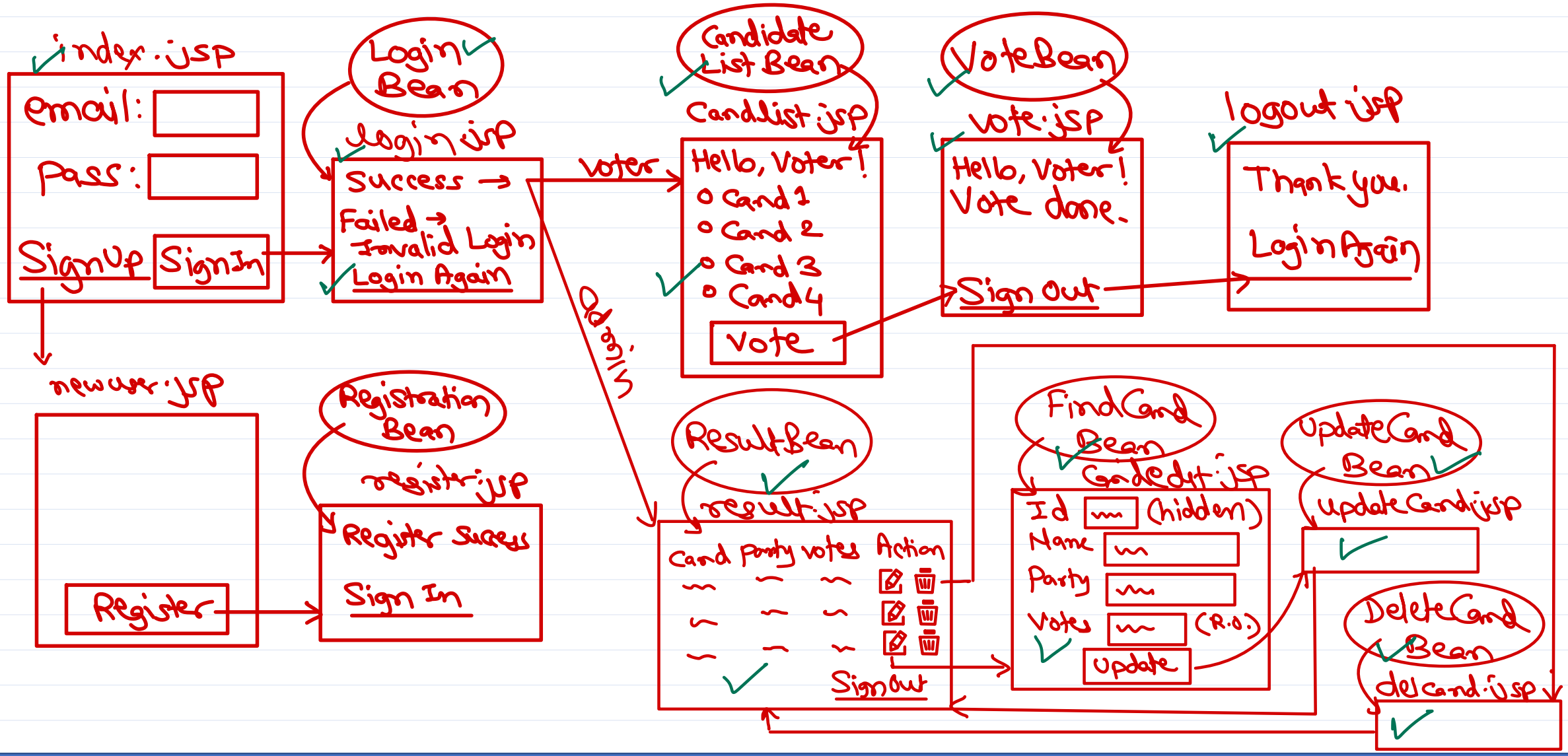
① check if given bean is avail in given scope. if avail access it.

② if not avail, create a new bean and add it into that scope.

```
LoginBean lb
  = session.getAttr("lb");
if(lb == null){
  lb = new LoginBean();
  session.setAttr("lb", lb);
}
```

To access bean obj or do setProperty/getProp in a page useBean must be done above in page.

Election Management



predefined tag lib by Sun-micro system.

JSTL 5 main components :

1. Core → redirect, vars, if, choose, forEach, ...
2. fmt → number/currency | date formatting
3. functions → String ops e.g. length, ...
4. xml → xml generation & parsing.
5. Sql → execute Sql queries from JSP page.

JSP EL → ${ expr }

Applications

① arbitrary calculations.

${ 2+3*4 }

${ 22 div 7 } ... /

${ 22 mod 7 } ... %

② Can access obj from any scope.

${pageScope.objName}

${requestScope.objName}

${sessionScope.objName}

${applicationScope.objName}

${objName} → auto search in all scopes from lowest to highest & refer the first found obj

③ Can access fields of objects (via getters).

${objName.fieldName}

↳ objName.getFieldName().

④ Can call methods of objects.

${objName.methodName()}

⑤ EL implicit objects

${param.paramName}

${paramValues.paramName}

${cookie.cookieName}

${initParam.initParamName}

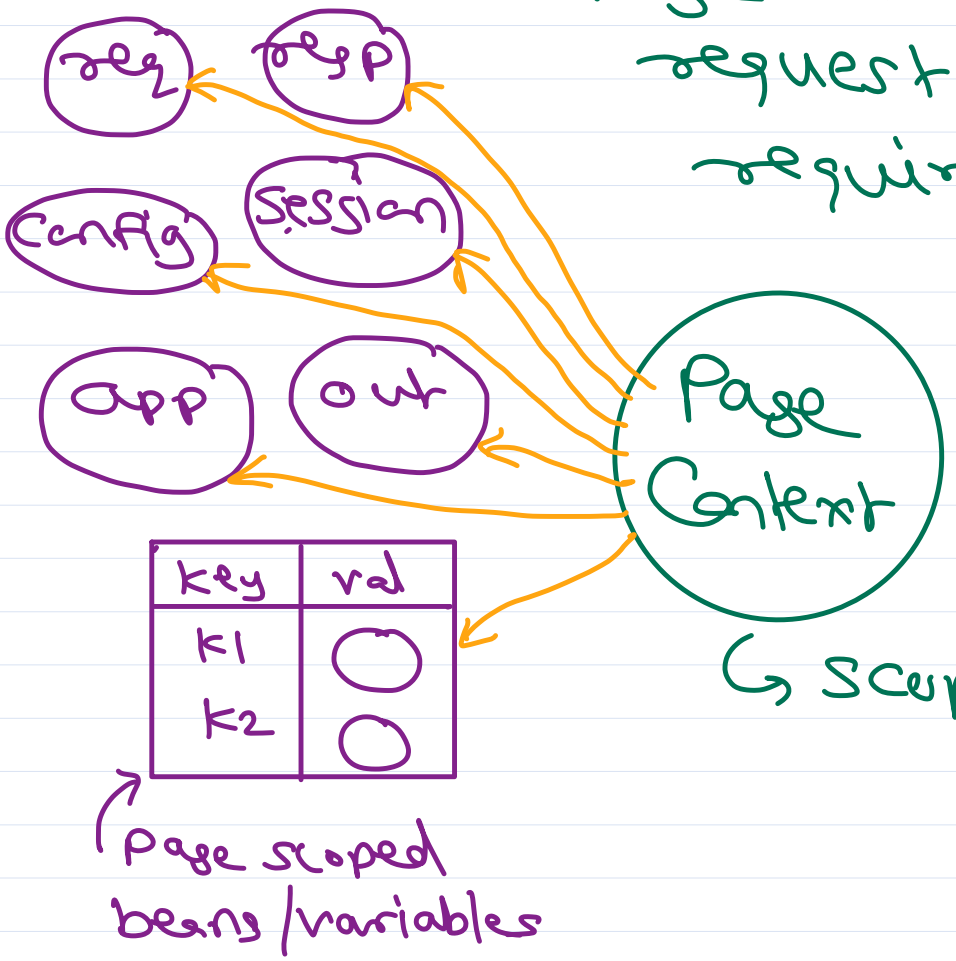
${header.headerName}

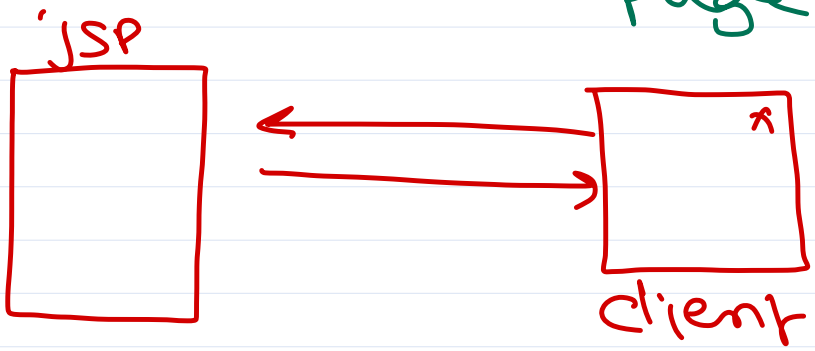
${headerValues.headerName}

${pageContext.___}

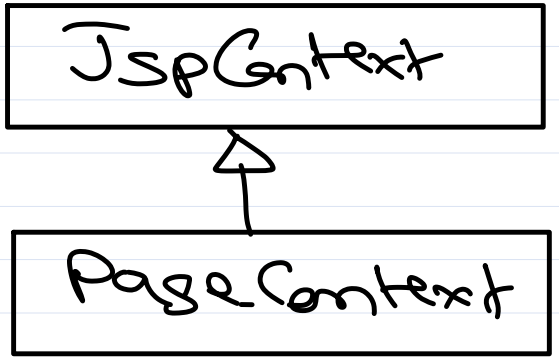
PageContext

PageContext is an object created for each request that holds all the objects required for processing req to the JSP page.





↳ scope → current request to current page.



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>