

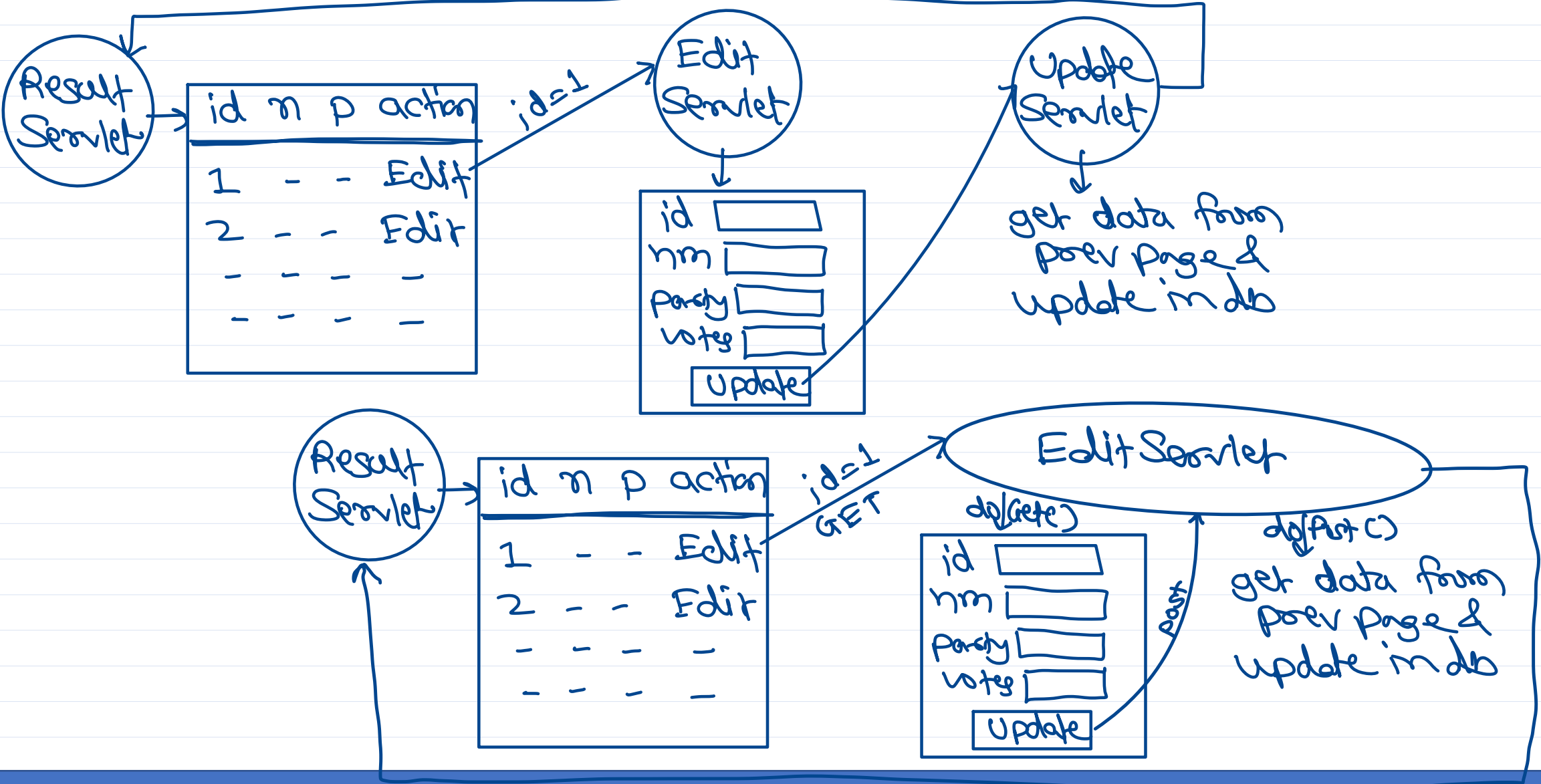


# Advanced Java

*Trainer: Nilesh Ghule*



# Edit record



# 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JSP is outdated.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)

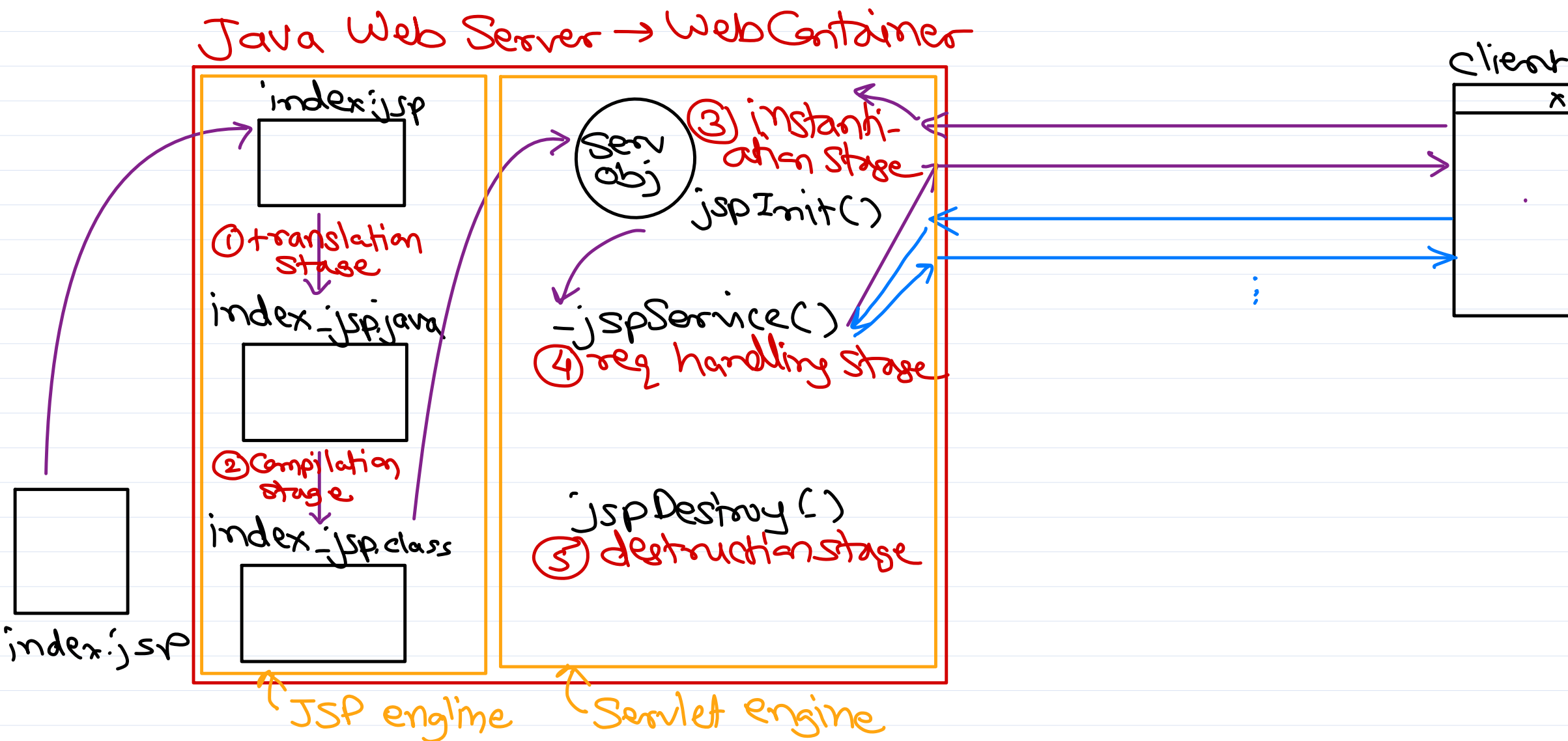
Ⓖ pageContext: PageContext

Ⓗ out: JspWriter

Ⓘ exception: Throwable  
avail only in err pages.



# JSP Life Cycle



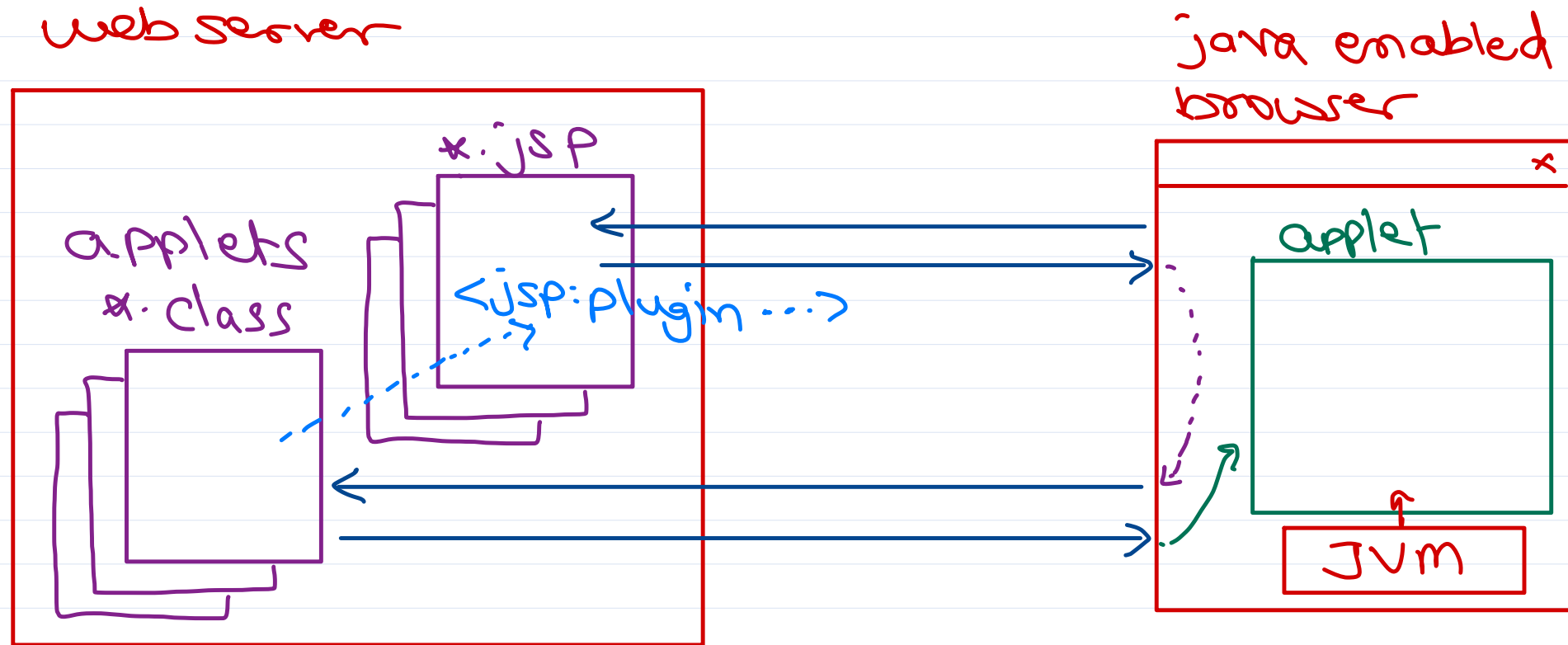
JSP file should have presentation logic  
i.e. markup syntax (tags).

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

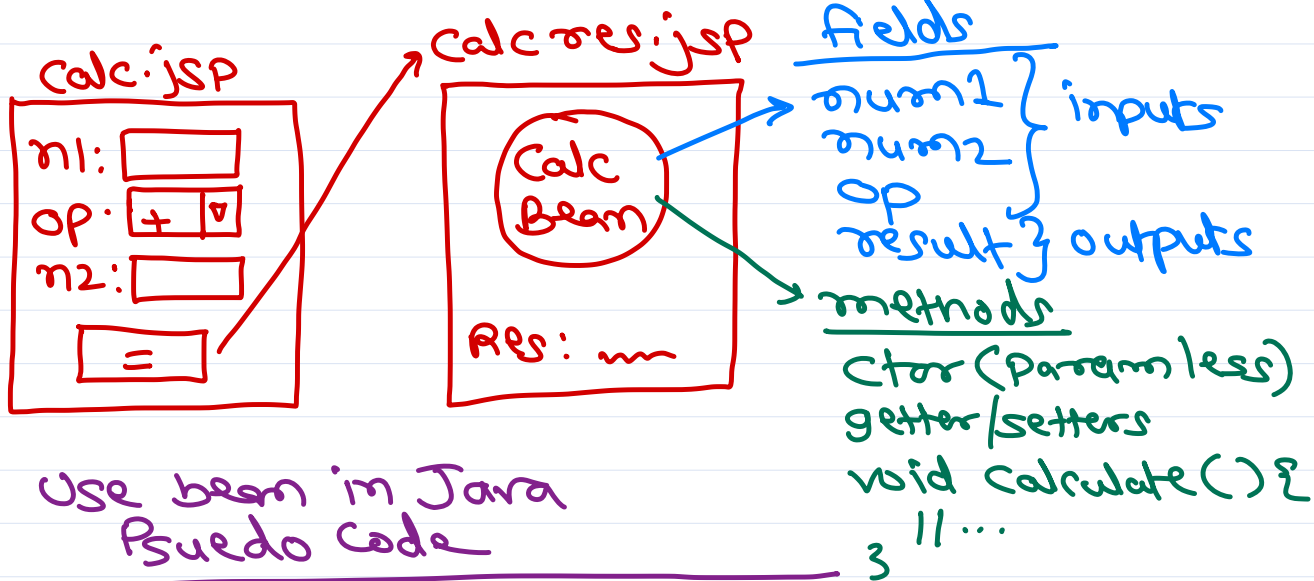




Applet is a java class that is downloaded to client side upon request and execute in browser JVM.

# Java Beans

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



Use bean in Java  
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"/>
```

field name ↑                      ↑ req param name

```
<% cb.calculate(); %>
```

```
<jsp:getProperty name="cb"  
                 property="result"/>
```



# Java Beans

fields: email, password, user

method:

authenticate() {

// jdbc code

// call dao.

Login Bean

login bean.jsp

demo6.jsp

email	<input type="text"/>
pass	<input type="text"/>
<input type="button" value="Sign In"/>	

```
<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="password"/>
// lb.setPassword(request.getParameter("password"));
<% lb.authenticate(); %>
<% if (lb.getUser() != null) { %>
    Welcome, <jsp:getProperty name="lb" property="email"/>
<% } else { %>
    Failed
<% } %>
```

// lb.getEmail() -> o/p in resp.





# Java Bean Scopes

<u>Scope</u>	default Scope	<u>attribute of</u>	
① page	(lowest)	→ PageContext	→ accessible in cur req to cur page only.
② request		→ HttpServletRequest	→ accessible on all pages where cur req is fwd on 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.

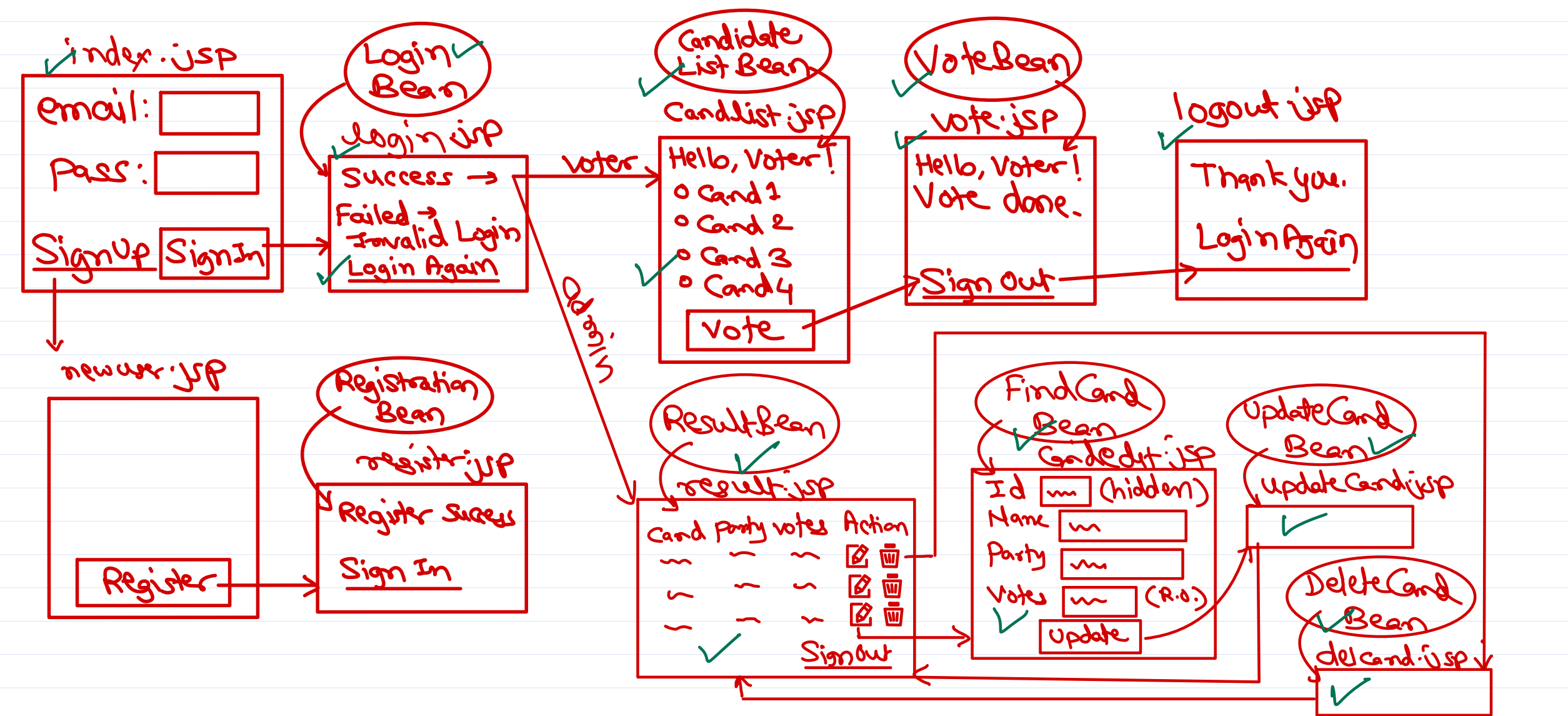
② if not avail, create a new bean and add it into that scope.

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

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



# Election Management





*Thank you!*

Nilesh Ghule <nilesh@sunbeaminfo.com>

