# Request Based Approach Wrap Up

BWA Slides #5

# Content

- Pagination
- Clean URIs
- Fileupload
- Front controller Pattern
- Design
- Authorization

# Pagination

```
<c:choose>
     <c:when test="${CURRENT_PAGE gt 0}">
          <a href=”...” >Prev</a>
     </c:when>
     <c:otherwise>
          <a href="" disabled >Prev</a>
     </c:otherwise>
</c:choose>
```


```
<c:choose>
     <c:when test="... use CURRENT_PAGE and PAGE_SIZE...">
          <a href=”...” >Next</a>
     </c:when>
     <c:otherwise>
          <a href="" disabled >Next</a>
     </c:otherwise>
</c:choose>
```

Navigation in long lists (spread over many pages)

- Use a CURRENT_PAGE and PAGE_SIZE as a session attribute
- Use EL/JSTL in page

Example

```
<!-- The previous link -->
<a href="shop?view=products&amp;page=${sessionScope.CURRENT_PAGE-1}"
class="btn">Prev</a>
```

# Clean URIs

Non clean

```
http://www.example.com/Blogs /Posts.php?
Year=2006&Month=12&Day=19
```

Clean

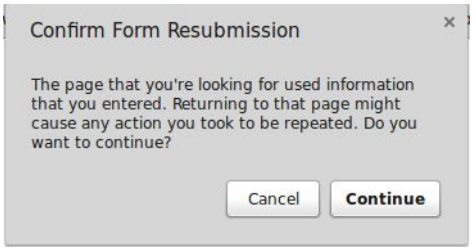
```
http://www.example.com/Blogs/2006/12/19/
```

"...Rewritten URIs (sometimes known as clean URIs) are used to provide shorter and more relevant-looking links to web pages. The technique adds a degree of separation between the files used to generate a web page and the URI that is presented to the world [hide implementation details]."
//Wikipedia

We can implement Fancy URIs directly in pages by using relative URIs with leading "/" or ${pageContext.request.contextPath}

NOTE: Who will be responsible for clean URIs. Admin mapping server URIs to application?
Easy to administer! Application developer, Other ...?

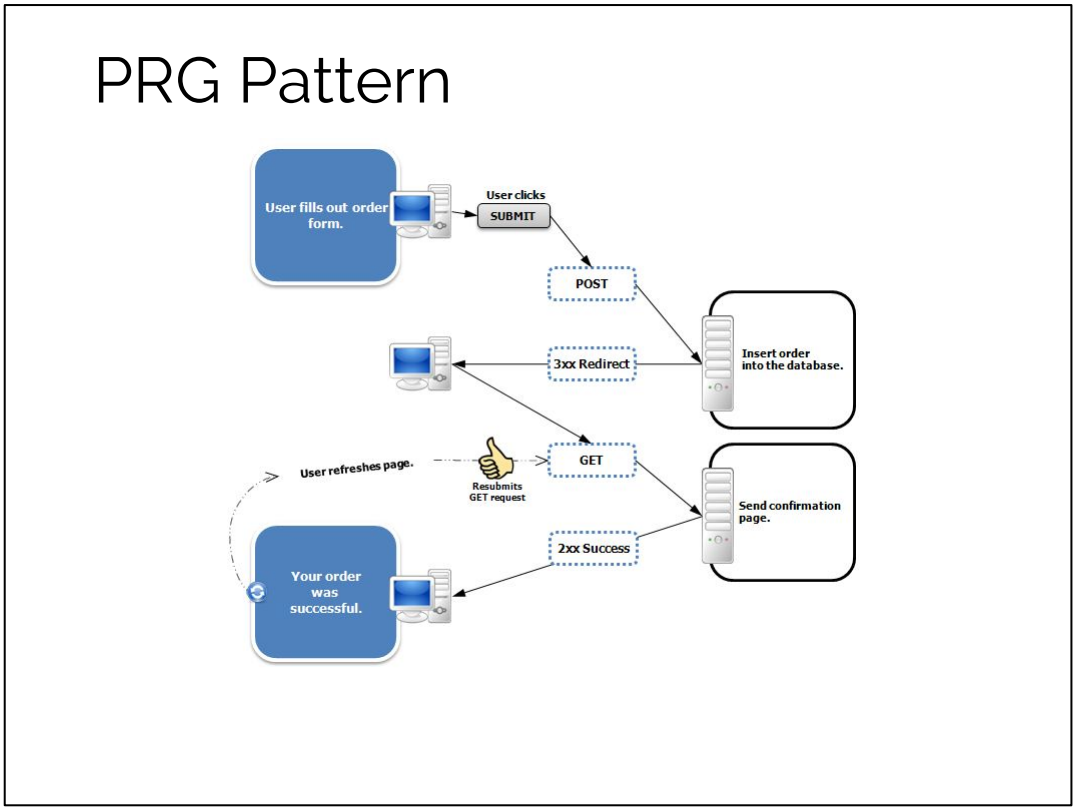
# The Double Submit Problem




Shouldn't be possible to submit same POST data twice i.e. buy two pair of shoes or similar (POST not idempotent)
Problems if:

- Clicking submit button twice.
- Using Refresh button (remember: POST will render a page)
- Using browser back button traverse back and re-submitting the form.
- Using browser history feature and re-submit the form.

Browser shows warning (but possibly confuses user)

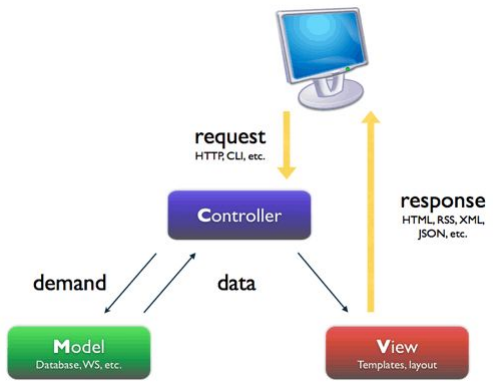



[POST/Redirect/GET-pattern]

To avoid the double submit problem send a redirect to client (a GET)

- If reload, another GET…
- … problem solved!

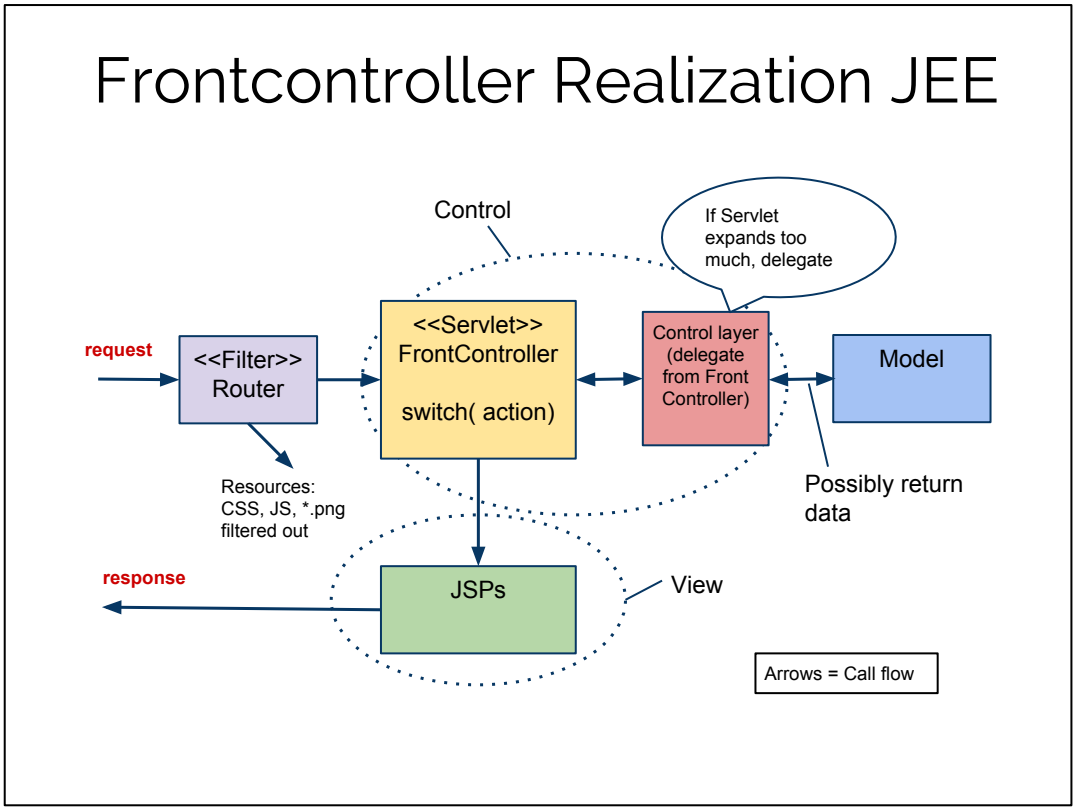
# Frontcontroller Pattern




MVC Web Style

Standard MVC design for request-based approach

- A “front controller” handles all request
- Centralized request handling, easier to control page flows
- Used in many frameworks




Finally Arriving at Request Based Style Application

- Have all techniques, need a design
- Normally using a framework like Spring or other (but we don't have the time)
- We make a simple inhouse MVC design

Components

- Router is a filter filtering out requests for static resources (CSS, JS , etc.). Static request not handled by application (container just return the resource)
- FrontController Servlet, handles logic for some part of application. Using the hidden action parameter to determine what to do
- Model is the core OO-model of the problem domain.
- JSPs handles the view parts. Servlet forwards (or redirects) and supply partials and data.
- Using fancy URLs inside application

No authentication or authorization for now

# Authentication


Very basic application managed

- If a user in session, login successful
  - Let a filter check
- Else redirect to login page
  - Let Servlet handle login