

# A Request Based Approach with JEE

BWA Slides #7

# Finally Arriving at Request Based Application

Have platform and near all techniques

Need some design

- MVC
- PRG-pattern

# Characterization Review

Request based approach interact directly with low level HTTP concepts, explicitly handling request and responses

## Pro

- Full control
- Many programmer like

## Cons

- Low abstraction level

Approach present in many other platforms (PHP,..)

# MVC for Request Based JEE

This is often hidden in some framework but we decided not to use any framework (because of time constraints)

- To get some understanding of the inner works of frameworks we'll do a simple MVC implementation...
- ...using the **FrontController (JEE) design pattern**

Some frameworks: Spring, Seam, ...

# Front Controller Pattern

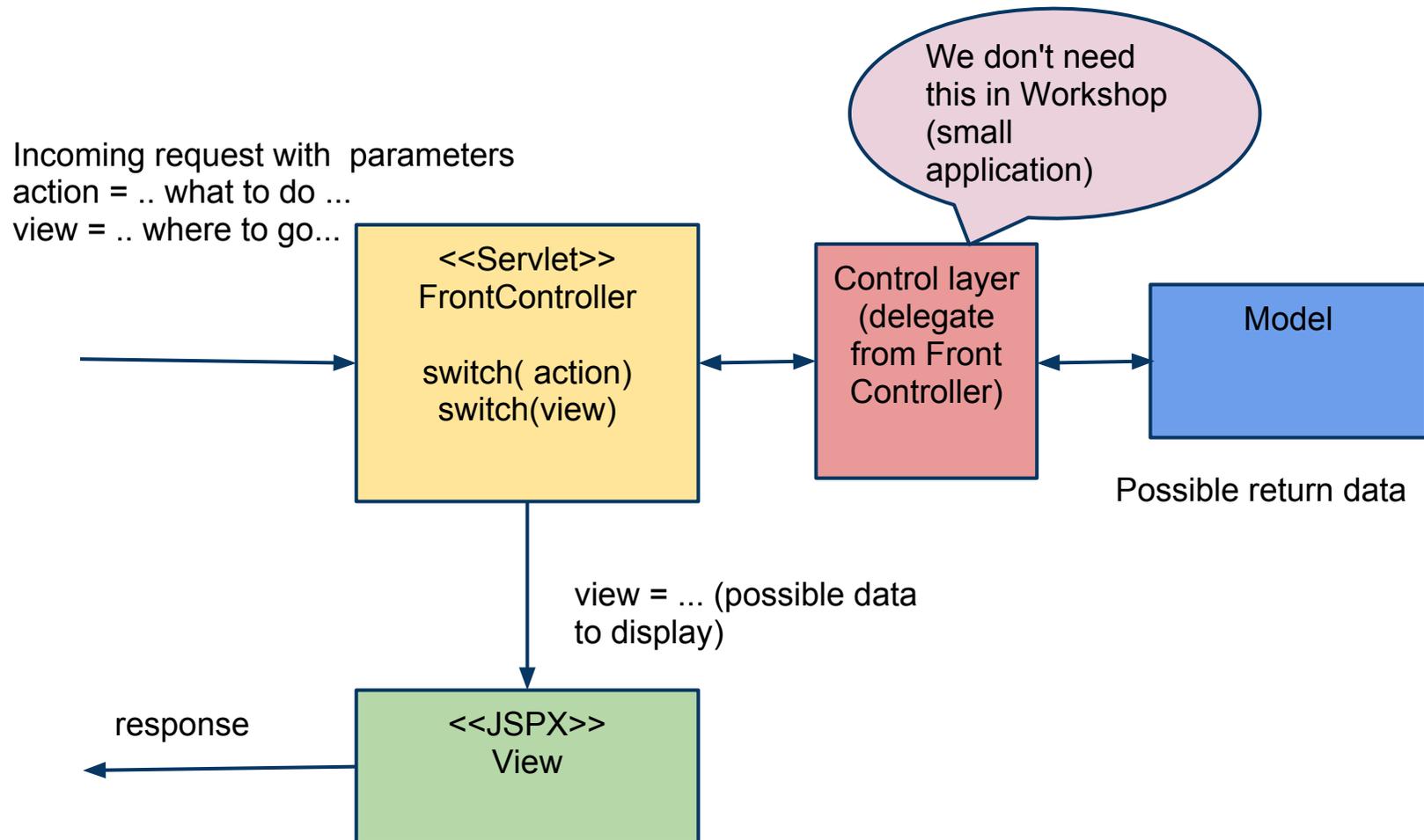
Model: Model objects (ShoppingCart,...)

View: JSP's

Control: A (single for now) "FrontController" Servlet

- "All" request passing through the Servlet
- Servlet do all processing (or delegate to control layer) using model objects and lastly decides the outcome view
- Asynchronous processing possible (but not used in course)

# Front Controller Pattern, cont.



# The “Double Submit” Problem

Shouldn't be possible to submit same post data twice  
i.e. buy two shorts or other (post not [idempotent](#))

If page rendered is result of a post (remember!) we possibly get another post if

- reloading result page using Refresh/Reload browser button (explicit page reload, implicit resubmit of request)
- clicking Back and then Forward browser buttons (implicit page reload and implicit resubmit of request)
- returning back to HTML form after submission, and clicking Submit button on the form again (explicit resubmit of request)

Browser shows warning (but possible confuses user)

# The PRG pattern

Post-Redirect-Get pattern is a solution to the double submit problem

- Never show pages in response to POST
- Always load pages using GET
- Navigate from POST to GET using REDIRECT
- Reloaded with GET no problem, it's a read

More issues and details

[http://www.theserverside.com/tt/articles/article.tss?  
l=RedirectAfterPost](http://www.theserverside.com/tt/articles/article.tss?l=RedirectAfterPost)

**JEE MVC:** Let FrontController servlet redirect if handling a post

# Authentication

For now simple filter

Basic idea, see workshop

# Fancy URI's

"...Rewritten URIs (sometimes known as short or fancy 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

## Examples

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

No direct support in JEE 6 (will come in JEE 7)

# Designing URI's

Hard to say where it's going to end. i.e what URI will we finally show the user

Possible adhere to using "technical" URI inside application

- Easy for developer to understand

Map technical URIs' to user friendly (fancy) URI's

- Possible done by application (or configuration of application)
- Possible using a Filter
- Also possible done by server (application external)