

SOEN 387 Web-based Enterprise Application Design

Stuart Thiel

Concordia University
Department of Computer & Software Engineering

Fall, 2015

Outline

SoenEA

SOEN 387
Web-based
Enterprise
Application
Design

Stuart Thiel

SoenEA

Front Controllers in SOEN EA

- ▶ `org.dsrq.soenea.application.servlet.impl.SmartDispatcherServlet`
- ▶ This is a FrontController
- ▶ implements the idea of basic setup in the FrontController
- ▶ Both Servlet Setup and Request Setup
- ▶ Gives basic implementation using other bits of SOENEA

SmartDispatcherServlet Servlet Setup

- ▶ setup `#{sp}` and `#{br}`, convenience in JSP
- ▶ Sets up relative servlet context path (a convenience)
- ▶ many things pulled from `MyResources.properties`, sets up application context
 - ▶ `BaseAbsoluteURL` lets you avoid use of relative urls in JSPs!
 - ▶ `BaseURL` lets you know about context-path
 - ▶ `DefaultCommand` gives you a default `Command/Dispatcher` if none specified
 - ▶ `file.uploadDir` specifies a spot for uploaded files (if using disk)... probably want to clean this regularly
 - ▶ `errorJSP` specifies a default HTML JSP to print errors, SOENEA debugging
- ▶ Sets up a `FileUpload` handler, choosing between two standard implementations
- ▶ Sets up `DBRegistry` for subsequent DB access in requests

SmartDispatcherServlet Basic Request Setup

- ▶ The idea that FrontControllers have a pre-request setup, the actual request, post-request teardown
- ▶ Database Transaction started before the request
- ▶ Database Transaction rolled back at the end of the request
- ▶ ThreadLocals registered with ThreadLocalTracker nuked

SmartDispatcherServlet Request Setup

- ▶ All methods end up going through processRequest, capture method as a request attribute.
- ▶ Set up SOENEA's Helper, hiding the difference between multipart and regular requests
- ▶ Figures out Command/Dispatcher to use
- ▶ Basic logging
- ▶ Provides Basic Authentication, looking for a user id in session and setting the User to the request context
- ▶ Checks whether the user has role-based access to the Command/Dispatcher
- ▶ Generates a Dispatcher with the provided Command and executes it
- ▶ Application-level errors sent to an error View

Some of what the helper offers:

«interface» Helper
<pre>getString(pName:String) : String getInt(pName:String) : Integer getLong(pName:String) : Long ... getStrings(pName:String) : String[] getInts(pName:String) : Integer[] getLongs(pName:String) : Long[] ... getAttribute(key:String) : Object getRequestAttribute(key:String) : Object getSessionAttribute(key:String) : Object getApplicationAttribute(key:String) : Object ... and setters getFile(pName:String) : File getFileName(pName:String) : File getFileMime(pName:String) : File</pre>

Front Controllers Additional Behaviour

- ▶ Where does it get the command?
- ▶ What is this role-based application-level authorization
- ▶ Do we need to know details of the Helper Implementation?