SOEN 387 Web-based Enterprise Application Design

Stuart Thiel

Concordia University
Department of Computer & Software Engineering

Fall, 2015

Outline

Filters and Wrappers

Resource Links / Permanent Links

Client-Side Architecture

Applying AJAX

Splitting Off Front Controller Responsibilities

- Front Controllers Do A Lot
- From a maintenance perspective, messy
- Particularly per-request issues
- Filters and Wrappers can resolve this

Filters and Wrappers 1

- Filters/Wrappers “probably” interchangeable terms
- I use Filters
- Supported explicitly in some environments, like Java
- Supported implicitly in other languages/systems
    - Clever Use of Inlcudes?
    - Proxying?
    - Aspects?

Filters and Wrappers 2

- They allow things to happen:
  - before the processing of the main request
  - after the processing of the main request
- They generally have access to the Request context
- They sometimes even have access to parameters (and cookies)

What to Filter

- What do we filter before, what after the main body?
- What's in the main body?
  - Logging
  - Authentication (who are you?)
  - Authorization (are you allowed to?)
  - DB Connection Setup/Teardown
  - GUI? (maybe, I am skeptical. . . but imagine it)
  - URL Parsing!

REST and Resource Links Life-Cycle

- Makes it easy to interact with a site/server
- UI usually works by:
  - Showing stuff about the thing being interacted with
  - Allowing the stuff to be interacted with
- This means a request to **view**, then a Request to **act**
- Or, of course, a request to **view**, then a Request to **view** some more

Permanent Links, Server-Side

- Need to be able to determine where to go based on URL (Use Case)
- Need to be able to extract parameters from URL
- Then just proceed as normal

Example Permanent Link Information

- We talked about using actions // /SpaceTime/Pilot/1/promote
- Imagine using it for paging? Looking for page 2 of your pilots? // /SpaceTime/Pilot/p2
- Looking for page 2 of your pilots, with 25 pilots per page? // /SpaceTime/Pilot/p2c25

Other Permalink Structures?

- Deep Linking, parsing otherwise non-functional parts of the URL to adjust a page
- Anything else that you've seen?

Resource Links and HTTP Methods

- GET to do any of the View Requests
- PUT/DELETE/POST for the others (POST)
- On regular web pages, this means you:
  - look at a resource (or list of resources) using URLs/links
  - then either change the state of the resources using form submission
  - or just look at more resources
- With AJAX or thick-client applications:
  - look at a resource (or list of resources) using URLs (GET)
  - then either change the state of the resources (POST/PUT/DELETE)
  - or just look at more resources
- What's the difference?

Web Browsers

- Give you a default interface
- Meant to be driven by a human
- Has a means to issue GET requests with URLs
- Has a means to support the viewing of form submission
- Has a means to parse it for consumption by the human (rendering the HTML/CSS/etc. . .)
- Can encapsulate AJAX
- Otherwise only uses Resource URLs as provided by the server

AJAX/Thick Client

- Tools to support the issuing of requests
- Tools to support parsing XML/JSON responses
- Persistence of client
- Must explicitly understand any Resource URL structure/semantics

AJAX Architecture

- Not an explanation of the use of AJAX
- I like splitting of Use Case requests by procedure
- They mirror the active procedures in TDD?
- They are exactly the active procedures in the TDD, just written in Javascript
- The REST is just
  - Context of making those calls (like the test cases)
  - Parsing or processing User interaction with UI
  - Parsing the XML/json/snippet responses

AJAX Responses

- We're effectively going to "embed" responses into the live UI, so what types of responses do we have?
- XML: Bulkier, well supported, more robust, queries well supported
- JSON: Lighter, less explicit, slightly fewer features, queries well supported, already javascript, effectively
- HTML snippet, a bad idea, but super light and often convenient
- What should your server support?

When to AJAX?

- If using AJAX, do we want to only have AJAX use resource URLS?
- When do we want the browser URLS to adjust?
- How often should AJAX poll on persistent pages?
- Do we need AJAX to do a full refresh sometimes? Why?