Session State

Objectives

- Identify and discuss some complicating factors in enterprise information systems
  - *Concurrency*
  - Session State
  - Integrating functional and platform services
- Summarize some key concepts
- Point to design patterns to address these concerns
  - Read Fowler and other sources for details

Stateless and Stateful

- Problem: How to store the data that's used within a business transaction (user or client) that is not yet ready to be committed to the shared database of record?
- Stateless: An object does not retain state (values of its attributes) between method requests
  - A stateful server object must keep its state (and other resources) waiting between requests
  - A stateless server object can process other requests from other sessions
    - **Pool objects**, so need fewer objects to handle more user sessions
    - Fewer objects to handle idle, slow users
  - More scalable to handle many concurrent users
  - Consistent with HTTP as a stateless protocol

Client Interactions are Inherently Stateful

- Consider a shopping cart in an on-line purchasing application
  - Many (stateless) HTTP requests as the customer browses the catalog, checks out, manages their account, etc.
- **Session state** during a user ("business") **transaction is isolated** from other user sessions
  - **Shared state** across sessions must be recorded to "data of record" – long-term **persistent** and shared data committed to a database
- Session state may have ACID properties as do system transactions
  - <u>**Consistency**</u> and <u>**isolation**</u> are key challenges
- Some session data is not necessarily "stateful"
  - May **cache** data for convenience and performance

Ways to Store Session State

- Three options (with some blurring between them)

  - **Client Session State pattern**
  - **Server Session State pattern**
  - **Database Session State pattern**

Read Fowler’s discussions to guide your design trade-off decisions

Ways to Store Session State

KVPs+”gibberish”

- **Client Session State pattern**: Store data on client
  - Encode in URL, use cookies, use hidden fields on a web form, use objects in a rich client application

Ways to Store Session State

- **Server Session State pattern**
  - Hold data in a shared object, in a container-provided session dictionary, in a shared file system, etc.
  - Indexed and accessed using a session ID

Ways to Store Session State

- **Database Session State pattern**
  - Map session data to data tables and store in a database

An example with Ajax

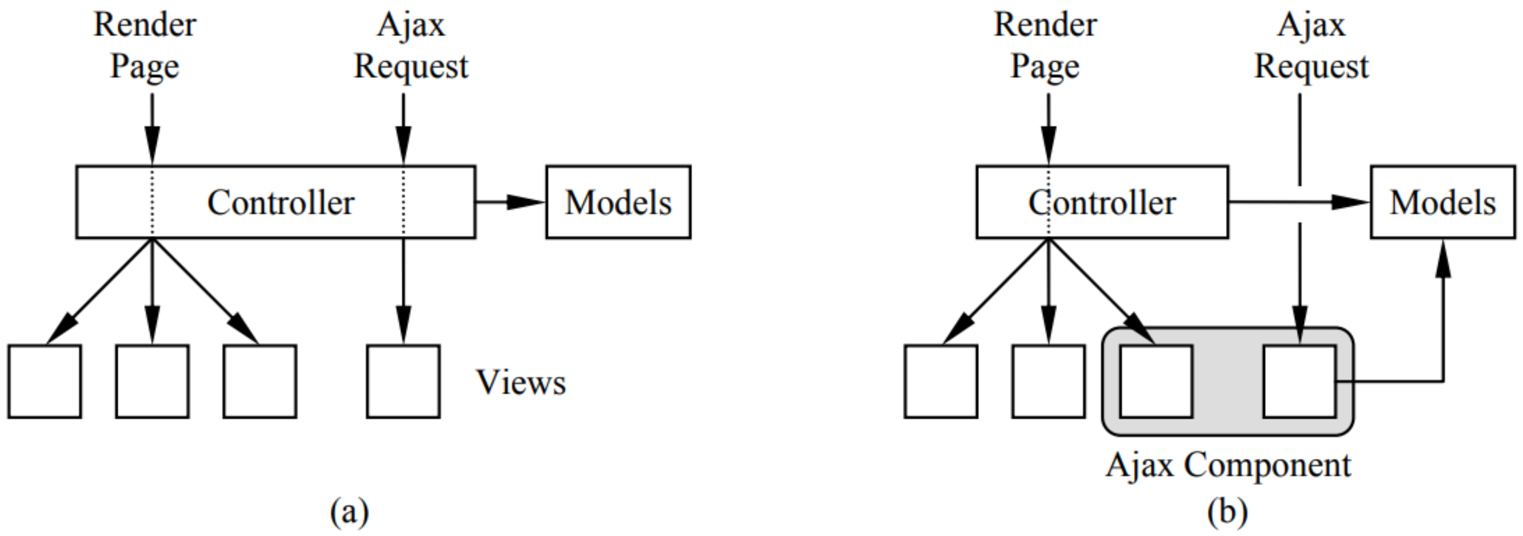

Figure 3. (a) The structure of a typical Web application where all HTTP requests for a page (including Ajax requests) are mediated by the page's controller class; (b) A component-oriented implementation in which all aspects of the Ajax-driven element, including both its original rendering and subsequent Ajax requests, are encapsulated in a reusable component. In (b) Ajax requests are dispatched directly to the component, bypassing the controller, and the component fetches its own data from model classes

Excerpts from: Managing State for Ajax-Driven Web Components - Ousterhout and Stratmann

An example with Ajax

- ***Reminders***: stores the state information on the **browser** with the page and returns the information to the server in subsequent Ajax requests.
  - Cons:
    - Overhead (depends on granularity)
    - Security (can use additional keys)

- ***Page Properties***: stores the state information on the **server** as part of the session.
  - Cons:
    - Crash implies need for persistence after each request (<Overhead)
    - Garbage collection (when to delete old page properties)

Conclusions

- Session Management is an important concern for enterprise information systems
  - There are design patterns to address these.
  - Consider how your enterprise information system can leverage persistence of session information.
  - Make sure to understand what options are afforded to you in the chosen platform.
- Evaluate not only the why but the when session data is used.
- Understand the underlying choices made by the framework.

Activity: Session State Research

Report a **brief list** (URLs included) of Session State based tutorials or resources that are most applicable to the mechanism your team is planning on using (or is currently using) to maintain session data in your assigned component(s).

Write a **brief description** on your team's approach to Session state management. Include:

- What does the approach require?
- Why are you using/choosing to use this approach?
- What are the PROS/CONS?
- What are the most important actions required by your team to support your approach under the class-wide ERP? (identify any dependencies or specific transactions/use-cases where this will be required)