Servlet Part 2

Objectives

- Collaboration
- Session Management

Collaboration

Collaboration

- When 2 components of same web application are interacting with each other, that process is known as collaboration.

Collaboration

- Benefits
  - Modularity
  - Reusability

RequestDispatcher

RequestDispatcher

- Used to achieve collaboration between the components running within the same web application.

RequestDispatcher

- Methods:
  - `public void forward(ServletRequest, ServletResponse);`
  - `public void include(ServletRequest, ServletResponse);`

RequestDispatcher

- `forward()`
  - A called resource generates the final response back to the client.
- `include()`
  - A calling resource generates the final response back to the client.

Session Management

Session Management

- HTTP is a stateless protocol.
- In a web application, an end user can make some transaction through one or multiple requests.

Session Management

- During this, server needs to maintain a conversational state along with the client.
- This technique is known as session tracking.

Session Management

- Different methods used for Session Tracking:
  - URL Rewriting
  - Hidden Fields
  - Cookies
  - Servlet API - HttpSession

HttpSession

HttpSession

- A Servlet API that is used to handle Session Tracking.
- `HttpServletRequest` is used to obtain the object of `HttpSession`.
  - `getSession()`
  - `getSession(boolean)`

HttpSession

- Important Methods:
  - `setAttribute()`
  - `getAttribute()`
  - `isNew()`
  - `setMaxInactiveInterval()`
  - `invalidate()`

Let's Summarize

- Collaboration
- Session Management