Module 3-7

HTTP Post

POST Requests

- Having seen GET Requests we will now examine POST requests.
- POST requests are often used when we want to affect the state of the application (i.e. having our server code insert rows into our database).
    - For example, a form to register a new user to an application will be submitted with a POST.

POST vs GET Requests

- A POST is preferable over a GET in the following circumstances:
  - It can better handle "accidental multiple submissions", i.e. continuously hitting a send / submit button.
  - Data transmitted through POST requests is not readily viewable from just the URL, POST data is in the request body, so POSTs are preferred for data that merits discretion.
  - Because all the transmitted data will be in the request body, POSTs are more efficient for when large amounts of data need to be transmitted.
- A GET is preferable when:
  - The transaction is "read-only", i.e. just pulling data from the database and displaying it on a page.
  - When you need to bookmark a request.

POST syntax

- The following pattern is used to tie a method to a POST request

```
@RequestMapping(path = "/surveyInput", method = RequestMethod.POST)
public String processSurveyInput(Survey survey) {

}
```
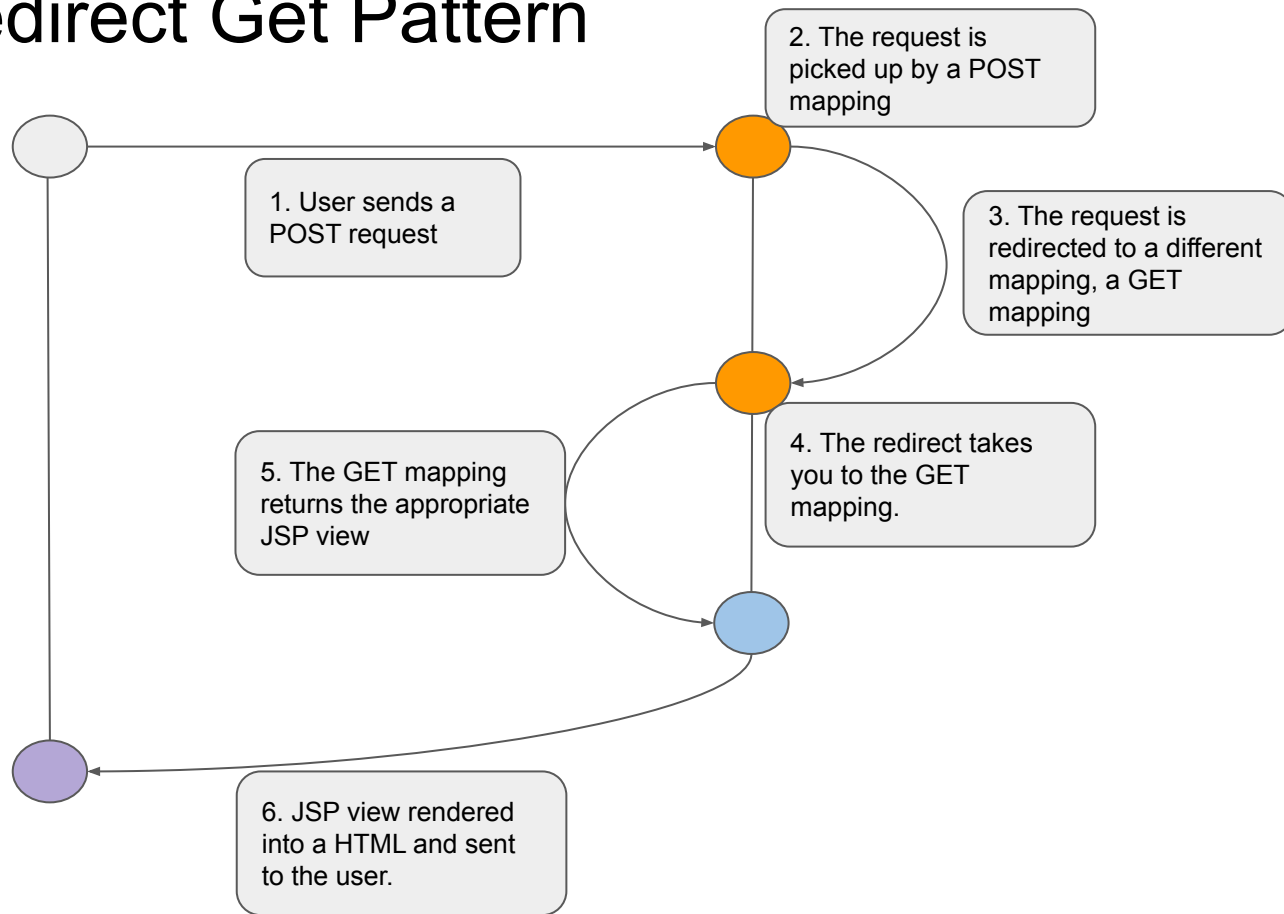
- Note that the same pattern can be adapted to do a GET request by just specifying RequestMethod.GET.

Coding Time!

POST Redirect Get Pattern

- Ideally, the server side logic to handle a POST should use the PRG (Post Redirect Get) pattern to prevent unpredictable results if the user refreshes the page.
- The PRG pattern requires three things:
  - A method designed to handle the POST request.
  - The "redirect" statement at the end of the aforementioned method that sends the user to a different endpoint.
  - A method to handle the GET request to which the request will be redirected to.

POST Redirect Get Pattern

2. The request is picked up by a POST mapping

1. User sends a POST request

3. The request is redirected to a different mapping, a GET mapping

5. The GET mapping returns the appropriate JSP view

4. The redirect takes you to the GET mapping.

6. JSP view rendered into a HTML and sent to the user.

POST Redirect Get Pattern

- Consider the following code:

```java
@RequestMapping(path = "/surveyInput", method = RequestMethod.POST)
public String processSurveyInput(Survey survey) {

    return "redirect:/surveyResult";
}


@RequestMapping(path = "/surveyResult", method = RequestMethod.GET)
public String showSurveyResult() {

    return "surveyResult";
}
```

Coding Time!