CS544 EA

Integration

REST: ResponseEntity

HTTP Response Contains

- An HTTP reponse actually contains multiple things
  - An HTTP status number
  - HTTP Headers
  - (Optionally) a response body

- Our controllers so far have only returned 1 thing
  - A response body (with a default status of 200)

ResponseEntity

- ResponseEntity lets you specify the full HTTP
  - Can be usefull in Controllers
  - Required part of RestTemplate API (coming up)

Constructor or Builder

```
@RequestMapping("/handle")
public ResponseEntity<String> handle1() {
  URI location = ...;
  HttpHeaders responseHeaders = new HttpHeaders();
  responseHeaders.setLocation(location);
  responseHeaders.set("MyResponseHeader", "MyValue");
  return new ResponseEntity<String>("Hello World", responseHeaders, HttpStatus.CREATED);
}

// same thing using the builder API
@RequestMapping("/handle")
public ResponseEntity<String> handle2() {
  URI location = ...;
  return ResponseEntity.created(location) // has .status(HttpStatus.CREATED)
     .header("MyResponseHeader", "MyValue")
     .body("Hello World");
}
```

Builder Start Methods

- accepted()
- badRequest()
- created(URI location)
- internalServerError()
- noContext()
- notFound()
- ok()
- ok(T body)
- status(HttpStatus status)
- status(int status)
- unprocessableEntity()

@ResponseStatus

- Using ResponseEntity might be a bit overkill
  - Perhaps you just want to add a different HttpStatus

- With @ResponseStatus you can do just that
  - Especially useful for Error Handlers

@ReponseStatus

```
@ResponseStatus(HttpStatus.CREATED)
@PostMapping(value="/person/")
public String post(@RequestBody Person person) {
  long id = personService.add(person);
  return "/person/" + id;
}
```