



AARHUS UNIVERSITET

Software Engineering and Architecture

Coding the HTTP in Java

The Good News

- Web is big! Lots of high quality frameworks for building both clients and servers on top of HTTP and REST!
 - Google a bit around to find some...
- Major qualities in teaching are
 - *Learnability* (tutorial quality, simple API)
 - *Code compactness* (expressiveness, code size)
- UniRest

```
Unirest.post("http://httpbin.org/post")  
  .queryString("name", "Mark")  
  .field("last", "Polo")  
  .asJson()
```

Spark-Java

```
import static spark.Spark.*;  
  
public class HelloWorld {  
    public static void main(String[] args) {  
        get("/hello", (req, res) -> "Hello World");  
    }  
}
```



AARHUS UNIVERSITET

Clients

UniRest

Minimal Demo

- (Find it as a zip file on the weekplan)

```
public class ShowUniRest {  
    public static void main(String args[]) throws UnirestException {  
        HttpResponse<String> reply =  
            Unirest.get("http://www.baerbak.com/contact.html").asString();  
        System.out.println("Reply status code: "+reply.getStatus() + " " + reply.getStatusText());  
        System.out.println("Content-type: "+reply.getHeaders().getFirst( key: "Content-type" ));  
        System.out.println("Body (200 chars):\n"+reply.getBody().substring(0, 200));  
    }  
}
```



```
ShowUniRest  
/usr/lib/jvm/java-8-oracle/bin/java ...  
Reply status code: 200 OK  
Content-Type: text/html  
Body (200 chars):  
<html>  
  <head>  
    <title>Flexible, Reliable Software</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">  
    <link href="style.css" rel="stylesheet" type="text/c  
  
Process finished with exit code 0
```

Actually a Bloch Builder

- Unirest.(verb)
 - Creates a verb object, but does not execute it
- .queryString, .field, .header, .body
 - ‘setter’ methods on the verb object
- asJson(), asString()
 - Builds and executes the verb
 - i.e. transmits the POST on the given URL with the set parameters

```
GET /contact.html HTTP/1.1
Host: www.baerbak.com
Accept: text/html
```

```
HttpResponse<JsonNode> jsonResponse = null;

String path = Constants.BLOODPRESSURE_PATH;
try {
    jsonResponse = Unirest.post( url: baseUrl + path).
        header( name: "Accept", MediaType.APPLICATION_JSON).
        header( name: "Content-Type", MediaType.APPLICATION_JSON).
        body(payload).asJson();
} catch (UnirestException e) {
    throw new IPCEException("POST failed for 'processAndStore'", e);
}
```

- Java 11 actually includes a Http Client library...

- Ala...

```
public RestMonitor() throws InterruptedException, IOException {  
    logger = LoggerFactory.getLogger(RestMonitor.class);  
    // We use the Java 11 httpClient library, if you are pre Java 11,  
    // I recommend the UniRest library  
    httpClient = HttpClient.newHttpClient();  
    request = HttpRequest.newBuilder()  
        .uri(URI.create("http://localhost:4567/temperature/217"))  
        .build();  
    monitorTemperature();  
}
```

- But...

- I actually find
UniRest more
Analyzable...

```
private void monitorTemperature() throws InterruptedException, IOException {  
    int count = 0;  
    while (count < 100) {  
        HttpResponse<String> response = httpClient.send(request, HttpResponse.BodyHandlers.ofString());  
        logger.info("Reading RSP from Rest Gateway: T = {} | response body: {}", response.statusCode(), response.body());  
        Thread.currentThread().sleep(Monitor.READ_DELAY_MILLISECOND);  
        count++;  
    }  
}
```



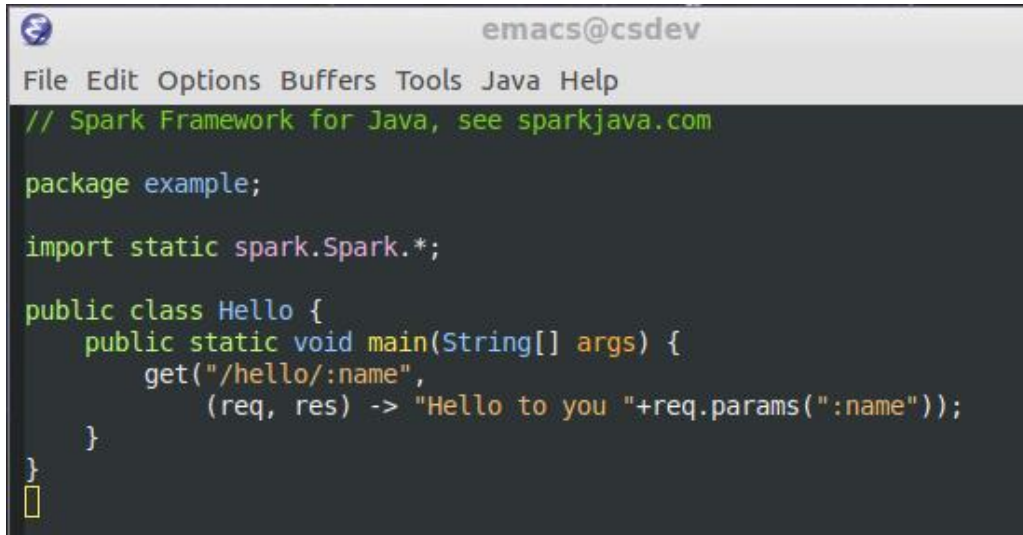
AARHUS UNIVERSITET

Servers

Spark-Java

Minimal Demo

- (Find it as a zip file on the weekplan)

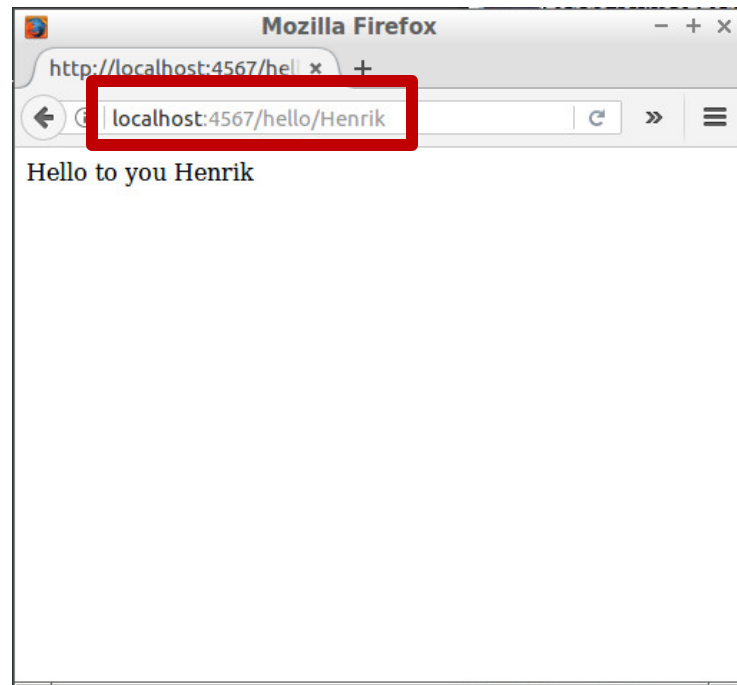


```
emacs@csdev
File Edit Options Buffers Tools Java Help
// Spark Framework for Java, see sparkjava.com

package example;

import static spark.Spark.*;

public class Hello {
    public static void main(String[] args) {
        get("/hello/:name",
            (req, res) -> "Hello to you "+req.params(":name"));
    }
}
```



req = request object
res = response object

```
// POST = processAndStore
String storeRoute = "/" + Constants.BLOODPRESSURE_PATH;
post(storeRoute, (req,res) -> {
    lastVerb = req.requestMethod();

    // Demarshall the body into the teleobservation posted
    String body = req.body();
    TeleObservation teleObs = gson.fromJson(body, TeleObservation.class);

    // Upcall to servant
    String id = teleMed.processAndStore(teleObs);

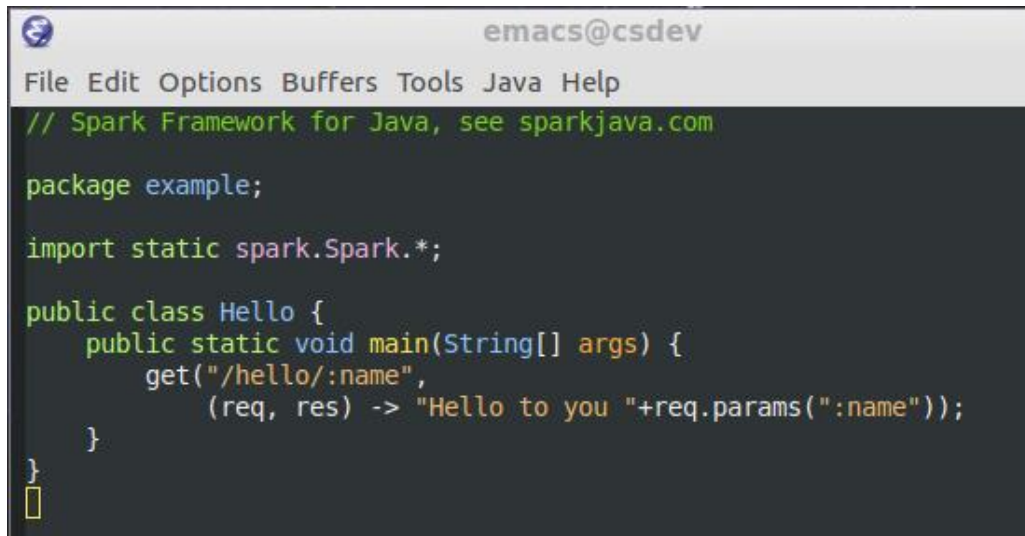
    // Set the CREATED status code
    int statusCode = HttpServletResponse.SC_CREATED;
    res.status(statusCode);
    res.type(MimeType.APPLICATION_JSON);

    // Location = URL of created resource
    res.header( header: "Location", value: req.host() + "/" + Constants.BLOODPRESSURE_PATH + id);

    lastStatusCode = statusCode;
    // Return the tele observation as confirmation
    return gson.toJson(teleObs);
});
```

Relating to SWEA Theory

- Analyzing the code
 - *A web browser is a rather complex piece of software*



```
emacs@csdev
File Edit Options Buffers Tools Java Help
// Spark Framework for Java, see sparkjava.com

package example;

import static spark.Spark.*;

public class Hello {
    public static void main(String[] args) {
        get("/hello/:name",
            (req, res) -> "Hello to you "+req.params(":name"));
    }
}
```

- Exercise
 - So – what is 'get()' method here in relation to the Spark-java framework?

Summary

- Rule #1 in software engineering:
 - *Someone has probably solved this problem before*
- So – go look for candidates
 - **Timebox** the learning curve and quality assessment
 - I have found Open Source libraries that were
 - Extremely hard to understand and use
 - Often also full of defects and inconsistencies
 - Pick the best candidate
 - And prepare for **do over**