Building Web Applications

WITH SPARK

Introduction to the Spark framework. A practical example: a REST server for tasks management

Goal

- Create REST services
  - in Java
- Learn a simple framework
  - but extensible

Spark

- "*A micro framework for creating web applications in Java 8 with minimal effort*"
- http://sparkjava.com/
- Mainly used for creating REST APIs
- Support
  - session, cookie, redirect, …
- Extensible
  - no HTML templating engine, ORM, etc. included
  - various packages to choose from

Spark Installation

- Use gradle

```
implementation "com.sparkjava:spark-
core:2.8.0"
```

- or Maven

```
<dependency>
    <groupId>com.sparkjava</groupId>
    <artifactId>spark-core</artifactId>
    <version>2.8.0</version>
</dependency>
```

Spark Quick Start

```
import static spark.Spark.*;

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

- Running the application starts the web server
  - it will run until you kill it

The Web Server

- By default, Spark runs a web server on:
  - http://127.0.0.1:4567
- Port can be customized, before anything else:

```
port(8080); // Spark will run on port 8080
```

- You can replace the embedded web server (Jetty) with other servers
  - by adding a configuration in your web.xml
  - http://sparkjava.com/documentation#embedded-web-server

Web Resources

- Each resource is implemented by a method

```
get("/", (request, response) -> {
    return("Hello World");
});
```

- Must specify
  - the (local) URL at which the resource will be visible
  - the HTTP verb
  - the content of the resource: return statement

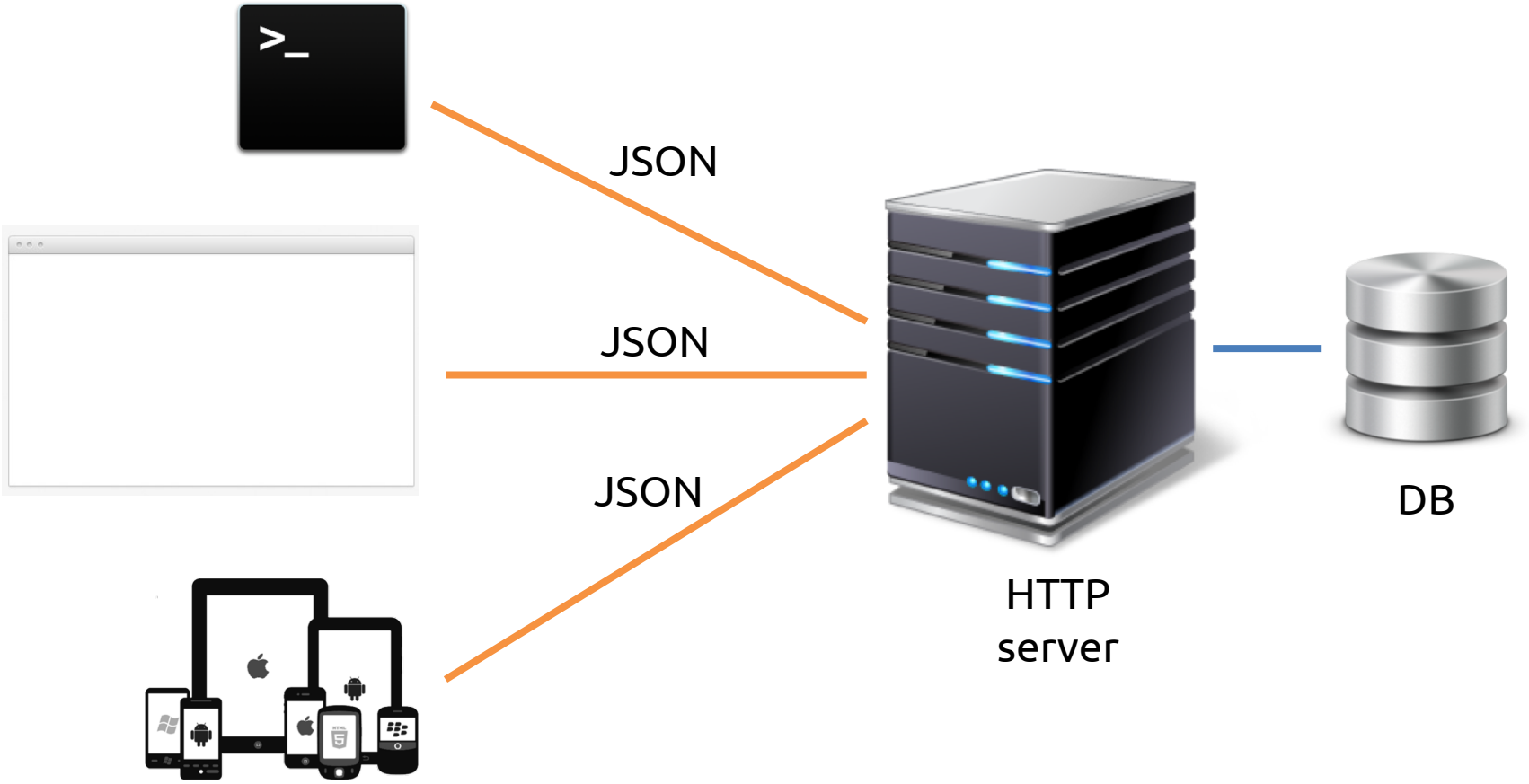
A case study

REST SERVER WITH SPARK

Spark Tasks Server

- A basic server that:
  - Connect to a database that contains a series of tasks
  - For each task, it provides:
    - task description
    - whether the task is "urgent" or not
  - Can show
    - a single task
    - all the existing tasks
  - Can create a new task

Spark Tasks Server



Resources

- Identify the resources to expose
  - Task
    - Represents a single task with its own information
    - Each task is identified by a unique id
  - Tasks
    - The set of tasks available

REST service design

- URLs built on resources
  - `/api/v1.0/tasks` → the collection of tasks
- How to represent a single task
  - They are identified by a unique id
    - `/api/v1.0/tasks/22` →the task with id=22

Tasks

- GET /api/v1.0/tasks
  - return all the tasks in JSON
- GET /api/v1.0/tasks/22
  - return task #22 in JSON
- Example

```
{
    "task": {
        "description": "buy a new mouse",
        "id": 22,
        "urgent": 1
    }
}
```

Tasks

- POST /api/v1.0/tasks
  - create a new task
  - an id is assigned automatically to the newly created tasks
  - task description and priority in JSON
- Example of the HTTP request body

```
{
    "description" : "This is a new task",
    "urgent" : 0
}
```

Result

- Checkout on GitHub
  - https://github.com/reti2vc-2019/spark-rest

References

- Spark
  - http://sparkjava.com/
- Spark Docs
  - http://sparkjava.com/documentation
- Spark Tutorials
  - http://sparkjava.com/tutorials

Questions?

MF0223 Reti 2

Luigi De Russis
luigi.derussis@uniupo.it

License

- This work is licensed under the Creative Commons “Attribution-NonCommercial-ShareAlike Unported (CC BY-NC-SA 4.0)” License.
- You are free:
  - to **Share** - to copy, distribute and transmit the work
  - to **Remix** - to adapt the work
- Under the following conditions:
  -  **Attribution** - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
  -  **Noncommercial** - You may not use this work for commercial purposes.
  -  **Share Alike** - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.
- To view a copy of this license, visit https://creativecommons.org/licenses/by-nc-sa/4.0/