Spring REST: Getting Started

INTRODUCTION

Peter van Rijn

www.little-world.nl

Introduction

What is REST?

What is Spring?

Getting Started

- Basic Messages

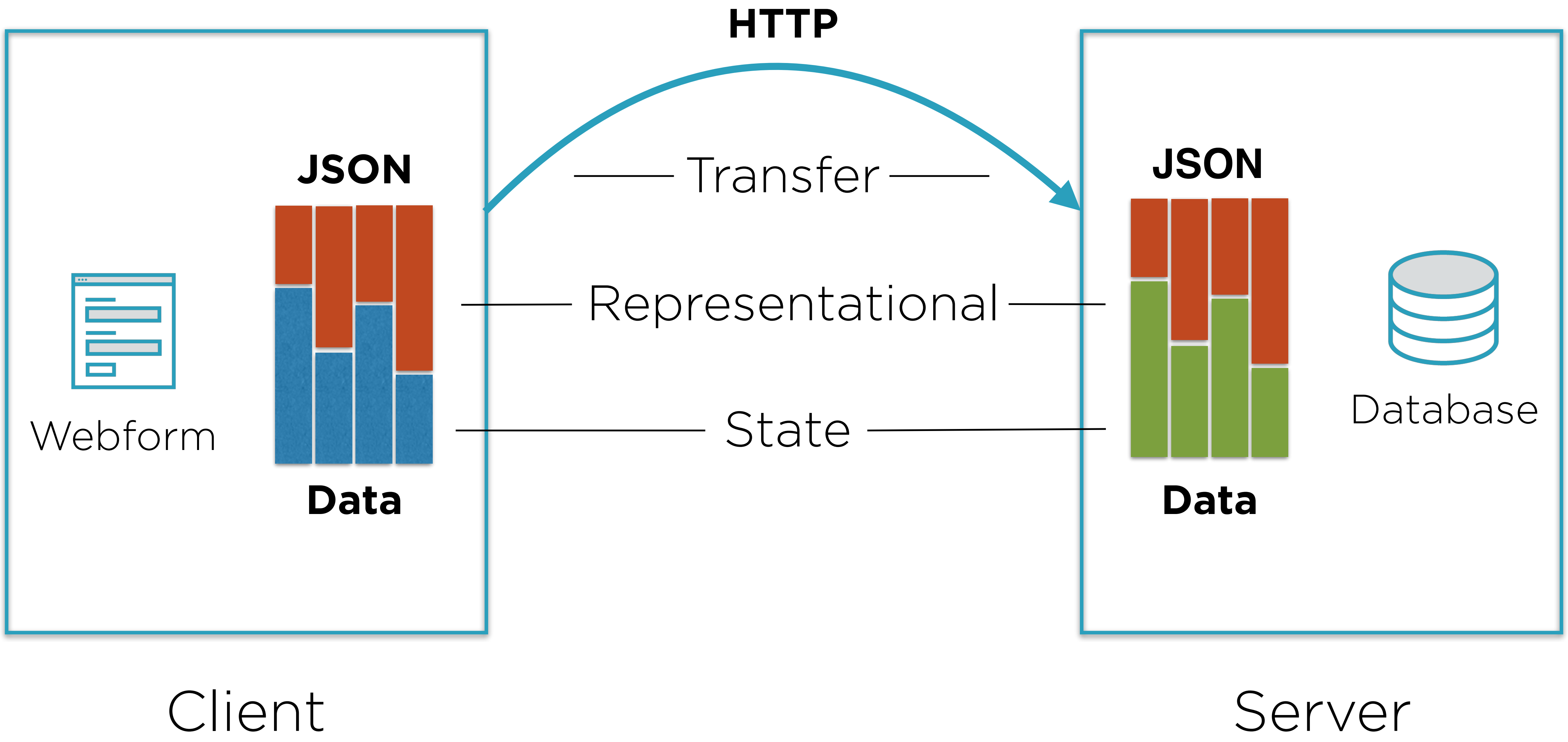
Explanation

What is REST?

Representational State Transfer (**REST**) is an architectural style that defines a set of constraints and properties based on HTTP.

Wikipedia

REST


HTTP
JSON
Transfer
Representational
State
Webform
Data
Client
JSON
Data
Database
Server

REST

Client - Server Architecture

Based on HTTP

- Request - Response Model
- Universal Resource Locator (URL)

HTTP Verbs:

- GET, POST, PUT, DELETE

HTTP Methods

Method	Description
GET	Retrieve data from the server
POST	Add data to the server
PUT	Modify already existing data
DELETE	Delete data from the server

What is Spring?

The **Spring Framework** is an application framework and inversion of control container for the Java platform.

Wikipedia

Spring Framework

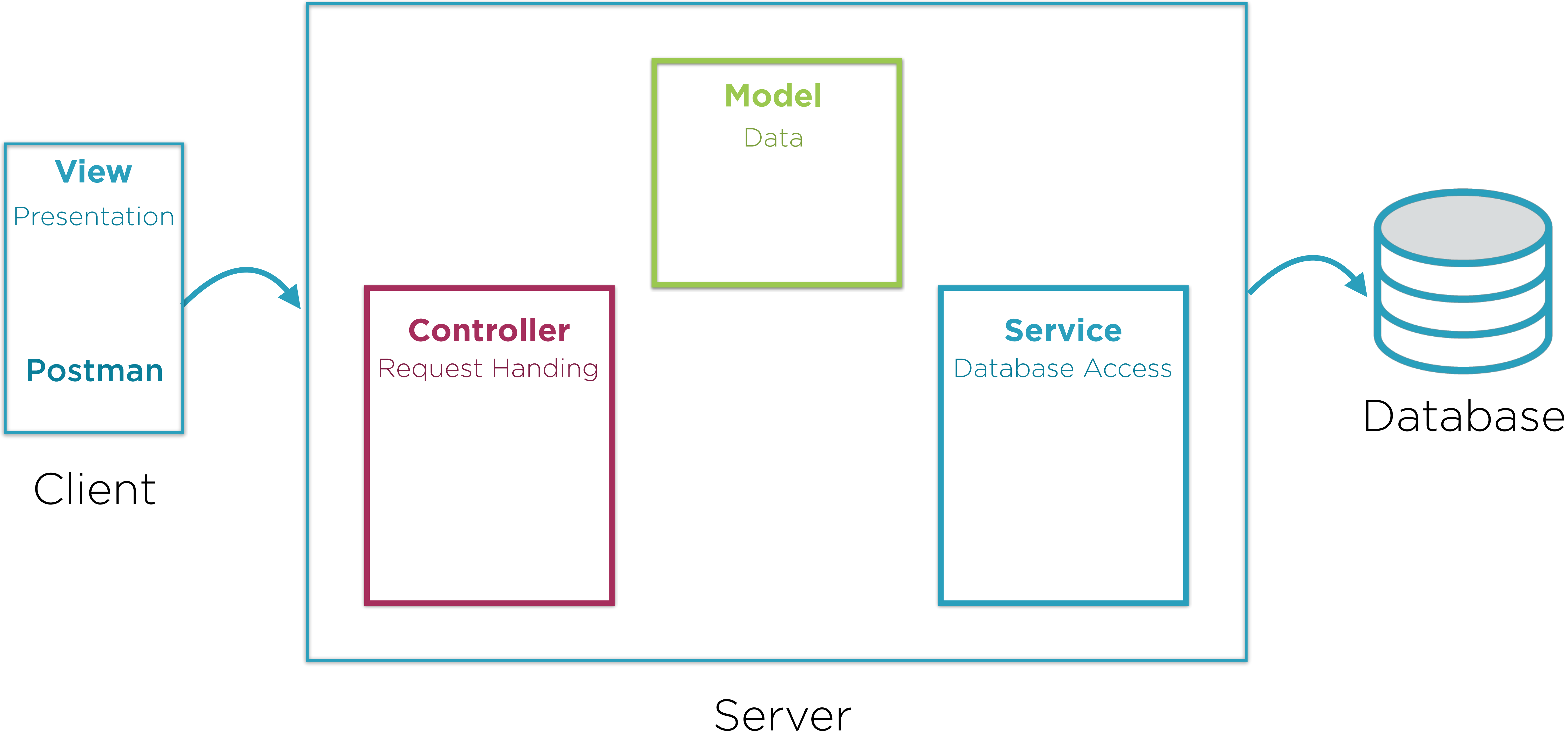
Enterprise Application Framework

- Large set of Libraries
  - Spring Boot
  - Spring MVC
  - Spring Data

Spring Container

- Life Cycle Management
- Dependency Injection
- Build from Configuration
  - Annotations
- Strength
  - Modularity
  - Extensibility

Spring REST Architecture



View

Presentation

Postman

Client

Model

Data

Controller

Request Handing

Service

Database Access

Server

Database

Getting Started

Generate a Demo Project

- http://start.spring.io

Import the Project into IntelliJ

Run and Test

Retrieve Data

Retrieve Data from the Server

- Request
  - GET http://localhost:8080/message
- Response
  - {"content": "first message"}
  - Test with Browser

Send Data

Send Data to the Server

- Request
  - POST http://localhost:8080/message
  - Data {“content”:”a post message”}
- Response
  - Echo the Request Data
- Test with Postman

Explanation

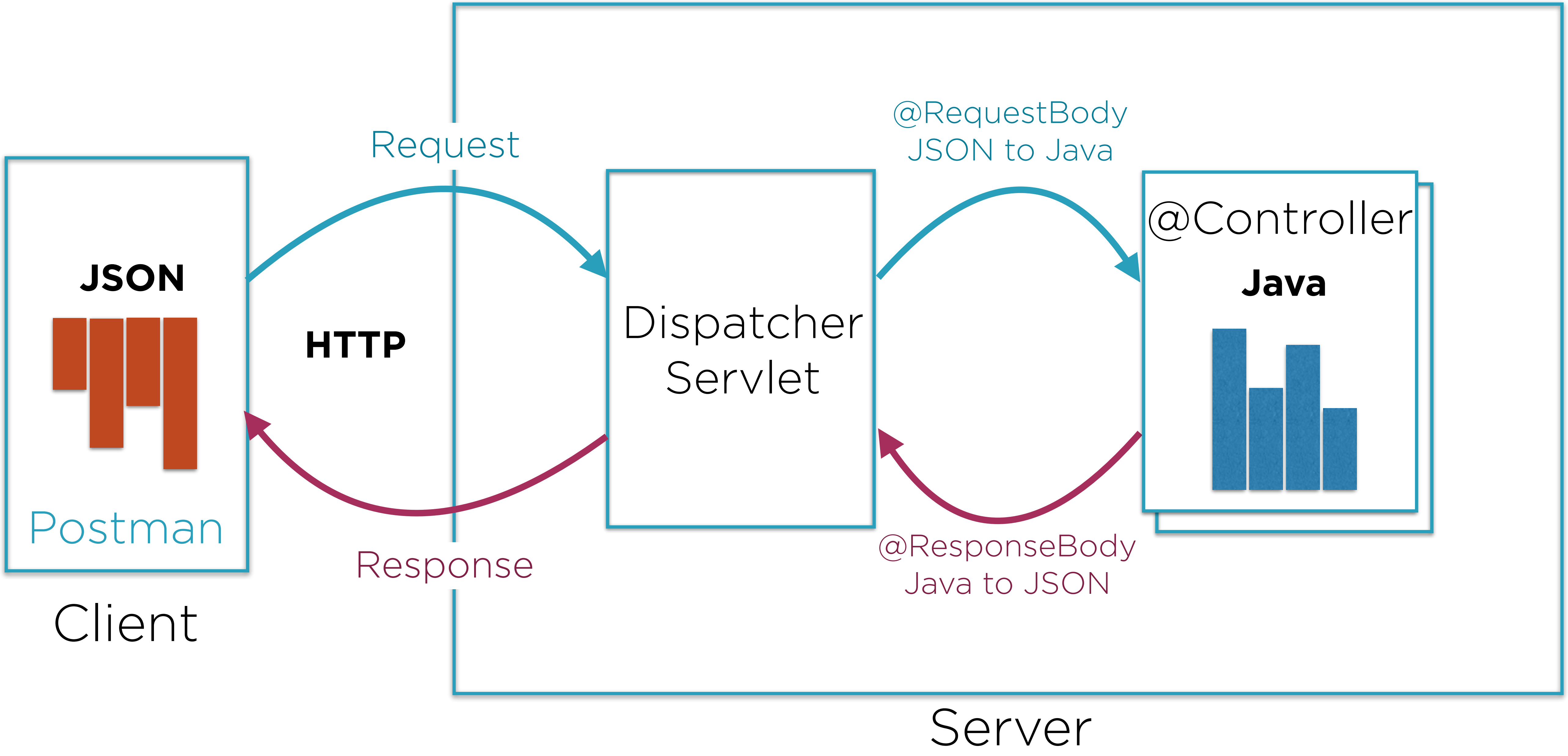
Spring Boot

- Container
- Run

Restful Web Service

- Workflow

RESTful Web Service



Summary

REST

- Representational State Transfer

Spring

- Enterprise Application Framework

- Spring MVC Architecture

Build a REST Web Services

- HTTP Methods: GET, POST

```
{
  "name": “john",

  "age": 34,

  "married": true,

  "children": ["mary", “elsa"],

  "address": {
    "street": “park lane 3",
    "city": "little town"
  },
  "unused": null
}
```

JSON types

◀ **string**

◀ **number**

◀ **boolean**

◀ **list**

◀ **object**

◀ **null**

```
@GetMapping("/message")
Message send() {
  return new Message("first message");
}
```

A Get Mapping

On a “/message GET Request

A Message object is returned