Components

THE UNIVERSITY
OF ARIZONA

Motivation

- Enterprise problems require prefabricates
- We need to develop quickly from bricks not just sand
- Frameworks provide quick patterns to combine

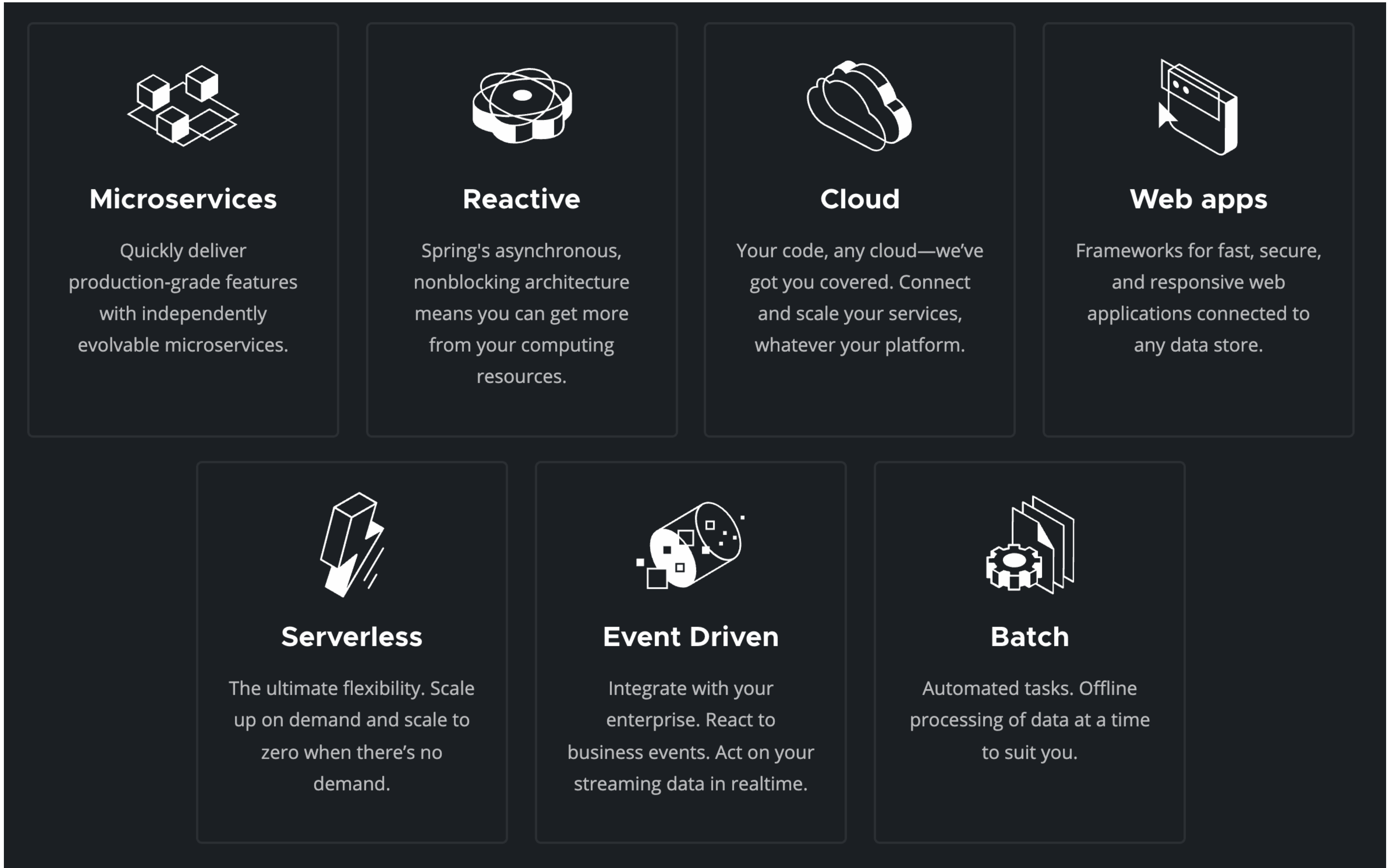
Java SE vs EE

- Delivery

Spring

- One of many choices
- Provides comprehensive programming & configuration model
- Targets Java-based enterprise applications - on any kind of deployment
- Typically web-based
  - Implied a web server

Many application domains



Microservices

Quickly deliver production-grade features with independently evolvable microservices.

Reactive

Spring's asynchronous, nonblocking architecture means you can get more from your computing resources.

Cloud

Your code, any cloud—we've got you covered. Connect and scale your services, whatever your platform.

Web apps

Frameworks for fast, secure, and responsive web applications connected to any data store.

Serverless

The ultimate flexibility. Scale up on demand and scale to zero when there's no demand.

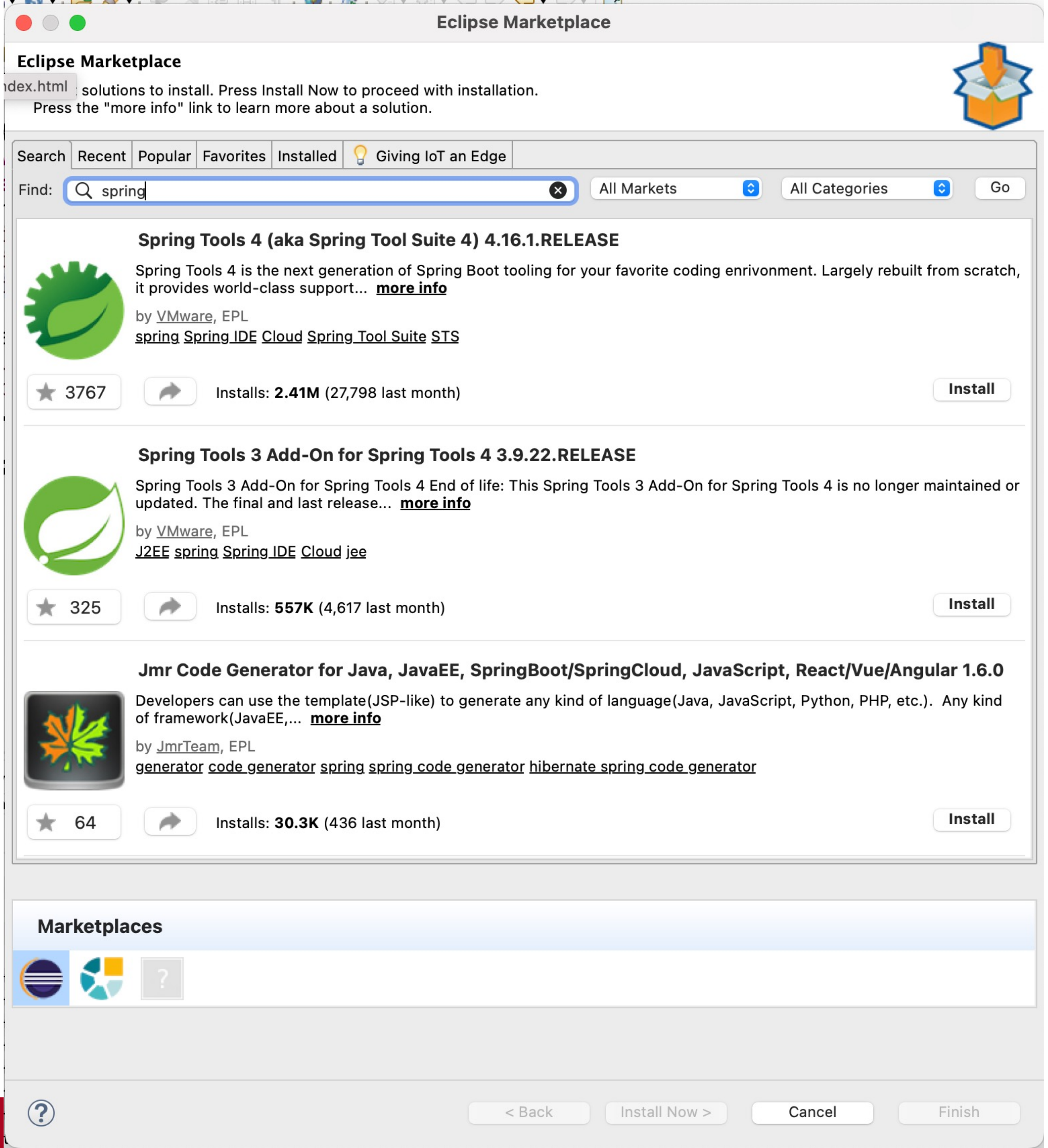
Event Driven

Integrate with your enterprise. React to business events. Act on your streaming data in realtime.

Batch

Automated tasks. Offline processing of data at a time to suit you.

Upgrade Eclipse



Eclipse Marketplace

Eclipse Marketplace

dex.html solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: spring

Spring Tools 4 (aka Spring Tool Suite 4) 4.16.1.RELEASE

Spring Tools 4 is the next generation of Spring Boot tooling for your favorite coding enrivonment. Largely rebuilt from scratch, it provides world-class support... **more info**

by VMware, EPL

spring Spring IDE Cloud Spring Tool Suite STS

3767 Installs: **2.41M** (27,798 last month) Install

Spring Tools 3 Add-On for Spring Tools 4 3.9.22.RELEASE

Spring Tools 3 Add-On for Spring Tools 4 End of life: This Spring Tools 3 Add-On for Spring Tools 4 is no longer maintained or updated. The final and last release... **more info**

by VMware, EPL

J2EE spring Spring IDE Cloud jee

325 Installs: **557K** (4,617 last month) Install

Jmr Code Generator for Java, JavaEE, SpringBoot/SpringCloud, JavaScript, React/Vue/Angular 1.6.0

Developers can use the template(JSP-like) to generate any kind of language(Java, JavaScript, Python, PHP, etc.). Any kind of framework(JavaEE,... **more info**

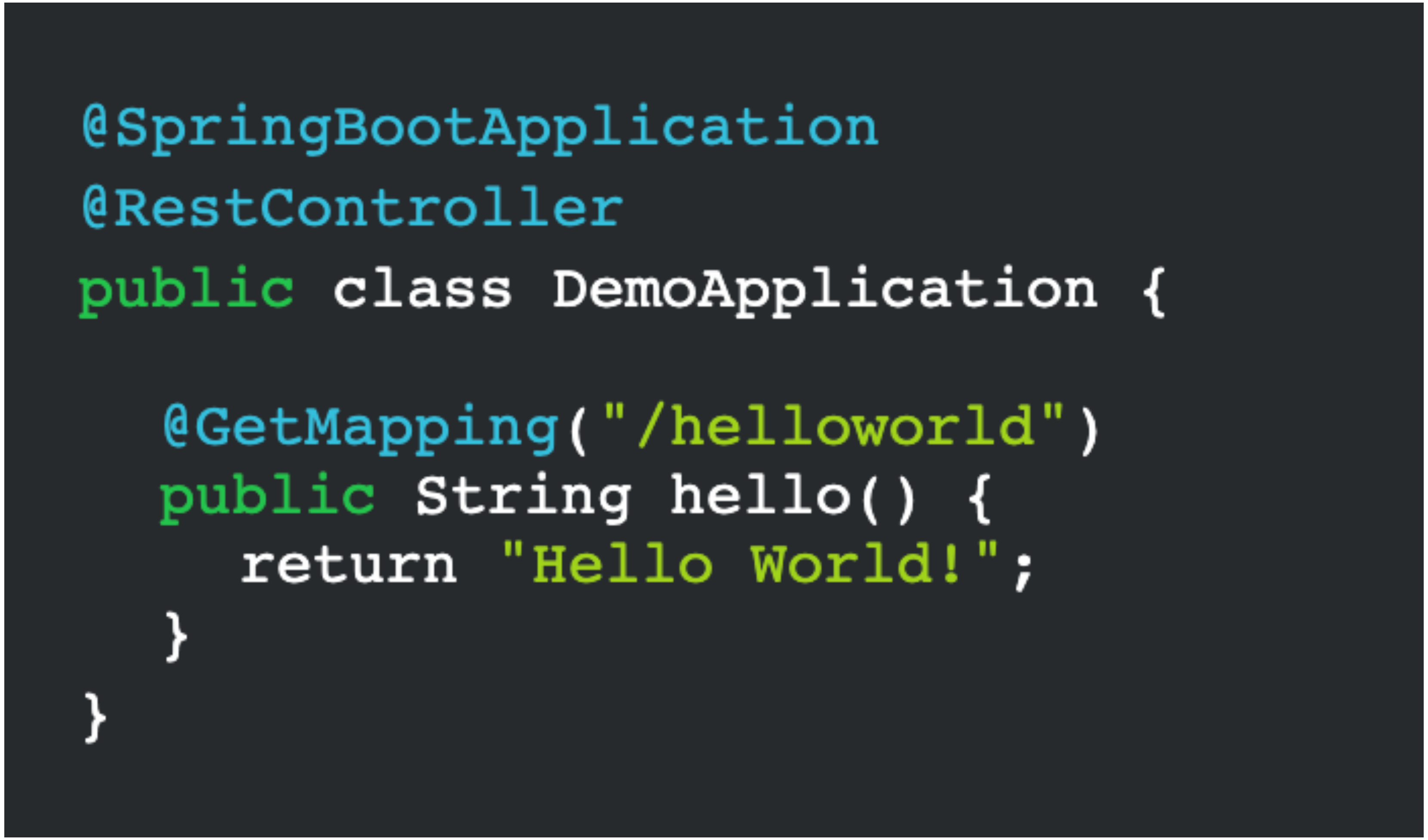
by JmrTeam, EPL

generator code generator spring spring code generator hibernate spring code generator

64 Installs: **30.3K** (436 last month) Install

Marketplaces

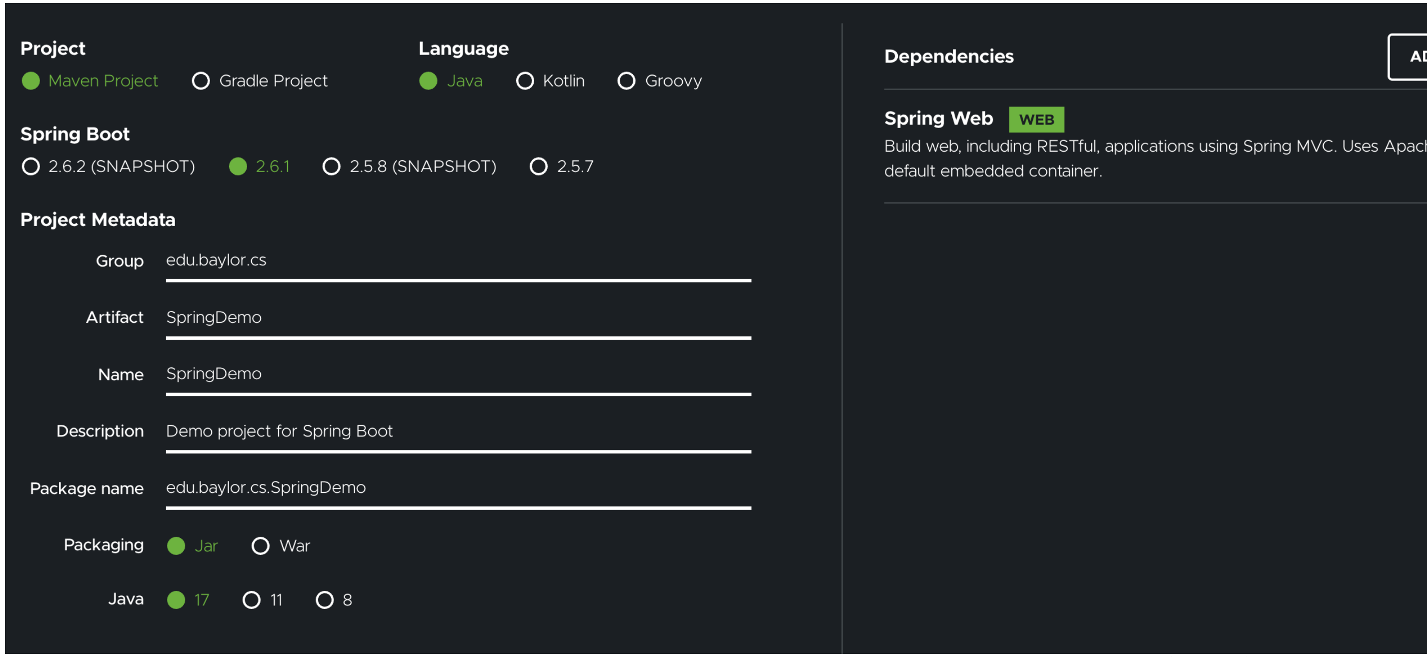
Spring quick start



```
@SpringBootApplication
@RestController
public class DemoApplication {

  @GetMapping("/helloworld")
  public String hello() {
    return "Hello World!";
  }
}
```

We need Maven!

- Use start.spring.io to create a “web” project.
- Generate and download as zip
- Unzip/import mvn



Project

● Maven Project ○ Gradle Project

Language

● Java ○ Kotlin ○ Groovy

Spring Boot

○ 2.6.2 (SNAPSHOT) ● 2.6.1 ○ 2.5.8 (SNAPSHOT) ○ 2.5.7

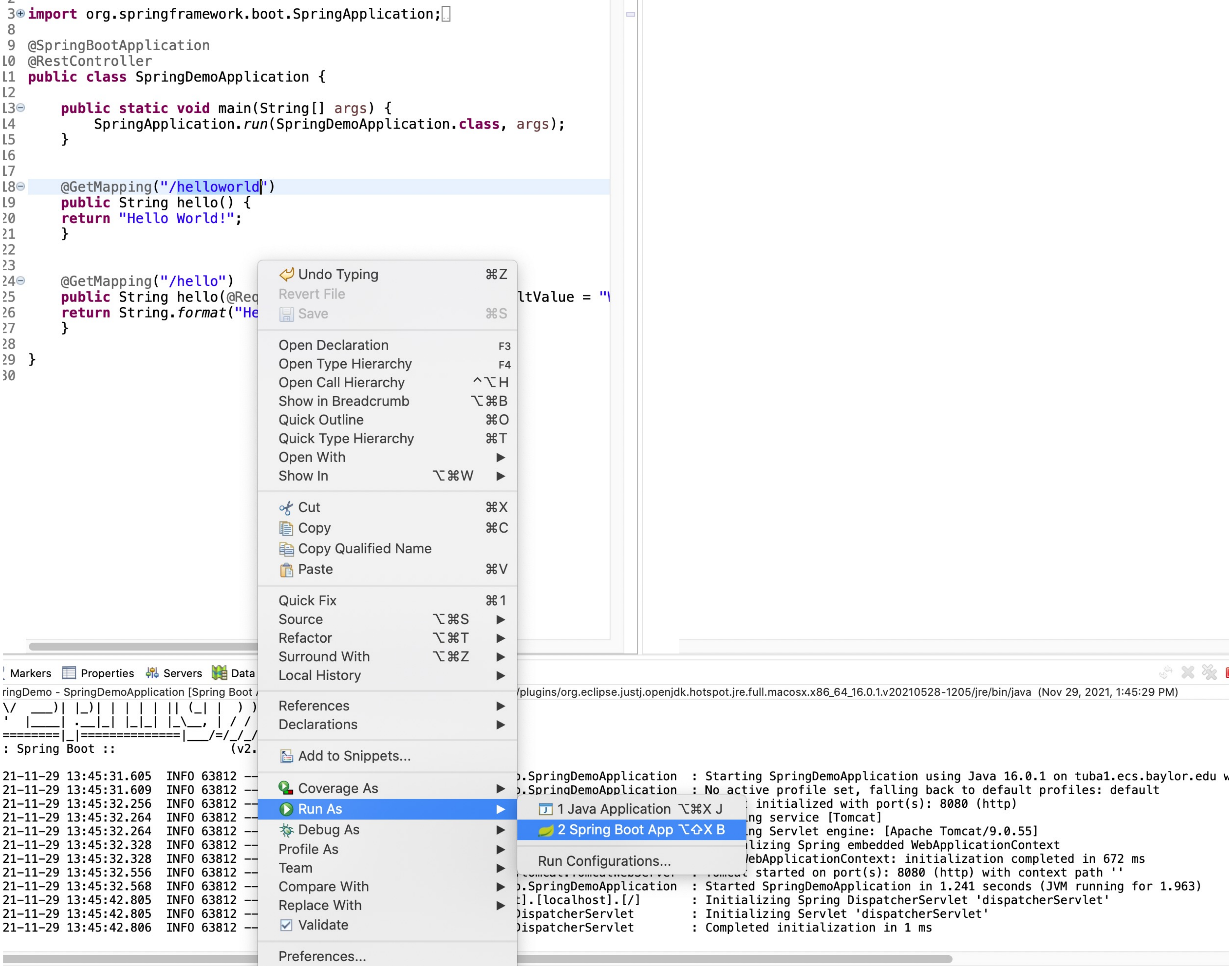
Project Metadata

Group	edu.baylor.cs
Artifact	SpringDemo
Name	SpringDemo
Description	Demo project for Spring Boot
Package name	edu.baylor.cs.SpringDemo
Packaging	● Jar ○ War
Java	● 17 ○ 11 ○ 8

Dependencies

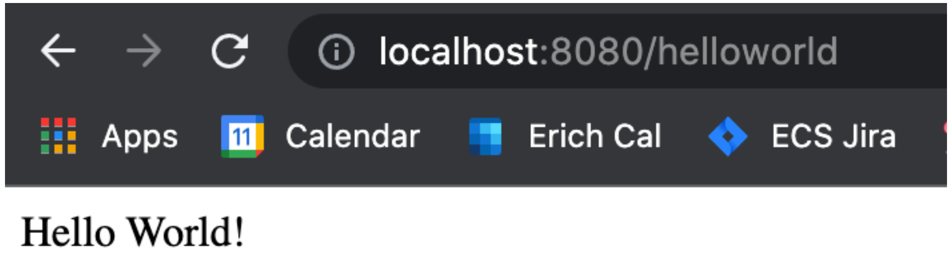
Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apac
default embedded container.

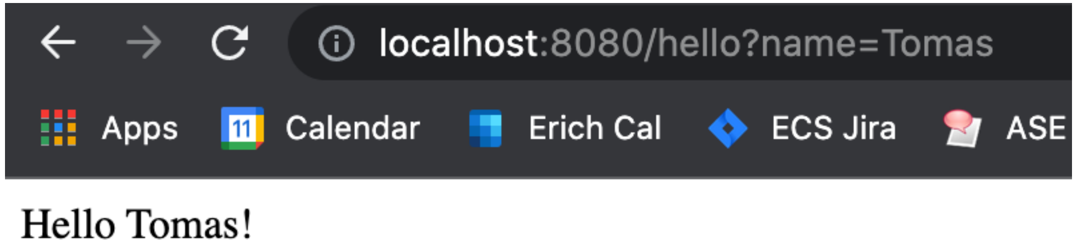


```java
import org.springframework.boot.SpringApplication;

@SpringBootApplication
@RestController
public class SpringDemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringDemoApplication.class, args);
    }


    @GetMapping("/helloworld")
    public String hello() {
    return "Hello World!";
    }


    @GetMapping("/hello")
    public String hello(@Req                                ltValue = "\
    return String.format("He
    }

}
```

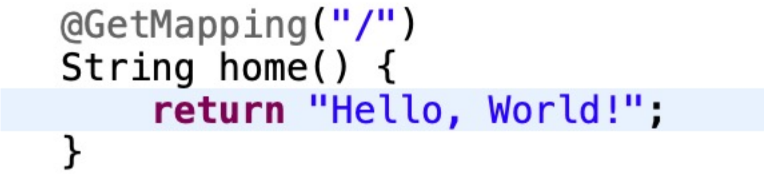
```
ringDemo - SpringDemoApplication [Spring Boot /            /plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_16.0.1.v20210528-1205/jre/bin/java  (Nov 29, 2021, 1:45:29 PM)
\/  ___)| |_)| | | | | || (_| |  ) )
'  |____| .__|_| |_|_| |_\__, | / /
=========|_|==============|___/=/_/_/
: Spring Boot ::                (v2.

21-11-29 13:45:31.605  INFO 63812 --    ).SpringDemoApplication   : Starting SpringDemoApplication using Java 16.0.1 on tuba1.ecs.baylor.edu w
21-11-29 13:45:31.609  INFO 63812 --    ).SpringDemoApplication   : No active profile set, falling back to default profiles: default
21-11-29 13:45:32.256  INFO 63812 --                               initialized with port(s): 8080 (http)
21-11-29 13:45:32.264  INFO 63812 --                              ng service [Tomcat]
21-11-29 13:45:32.264  INFO 63812 --                              ng Servlet engine: [Apache Tomcat/9.0.55]
21-11-29 13:45:32.328  INFO 63812 --                              lizing Spring embedded WebApplicationContext
21-11-29 13:45:32.328  INFO 63812 --                              ebApplicationContext: initialization completed in 672 ms
21-11-29 13:45:32.556  INFO 63812 --                               started on port(s): 8080 (http) with context path ''
21-11-29 13:45:32.568  INFO 63812 --    ).SpringDemoApplication   : Started SpringDemoApplication in 1.241 seconds (JVM running for 1.963)
21-11-29 13:45:42.805  INFO 63812 --    t].[localhost].[/]        : Initializing Spring DispatcherServlet 'dispatcherServlet'
21-11-29 13:45:42.805  INFO 63812 --    DispatcherServlet         : Initializing Servlet 'dispatcherServlet'
21-11-29 13:45:42.806  INFO 63812 --    DispatcherServlet         : Completed initialization in 1 ms
```



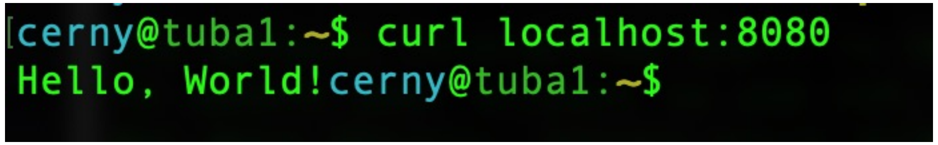
Hello World!



Hello Tomas!



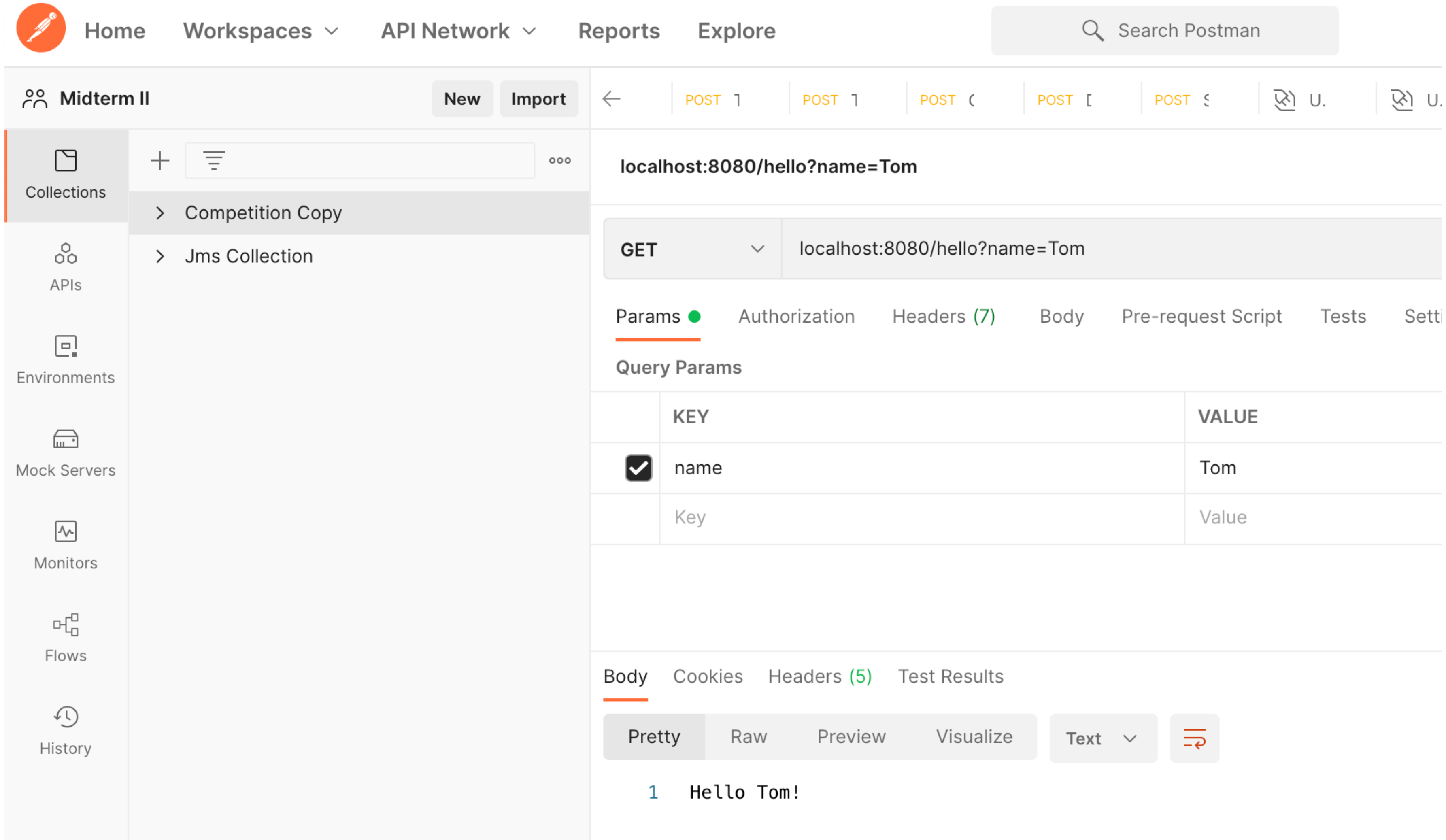
```java
@GetMapping("/")
String home() {
    return "Hello, World!";
}
```



```
cerny@tuba1:~$ curl localhost:8080
Hello, World!cerny@tuba1:~$
```

Postman project to test endpoints

- https://www.postman.com/



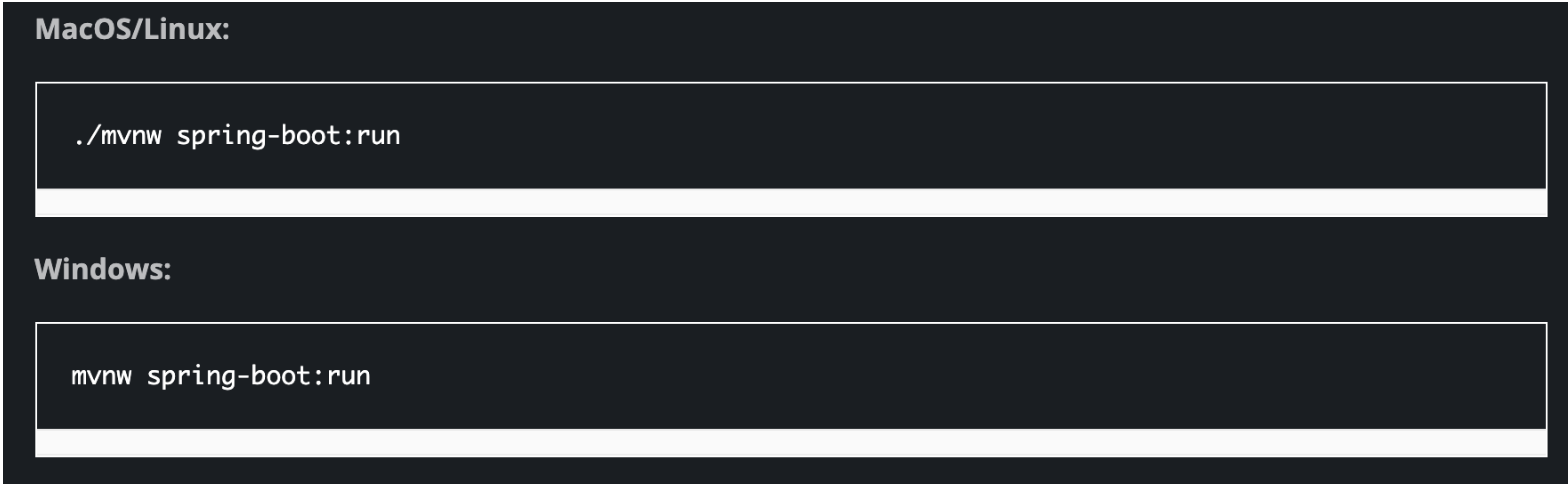
localhost:8080/hello?name=Tom

GET localhost:8080/hello?name=Tom

	KEY	VALUE
☑	name	Tom
	Key	Value

```
Hello Tom!
```

Maven



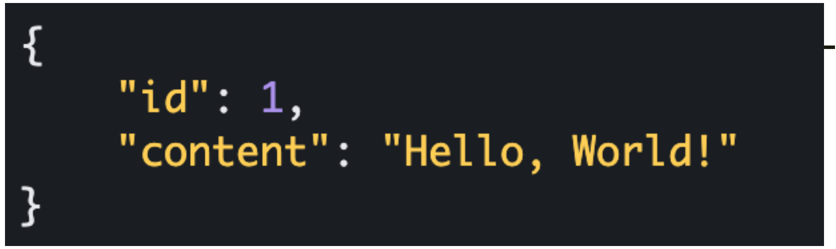
MacOS/Linux:

```
./mvnw spring-boot:run
```

Windows:

```
mvnw spring-boot:run
```

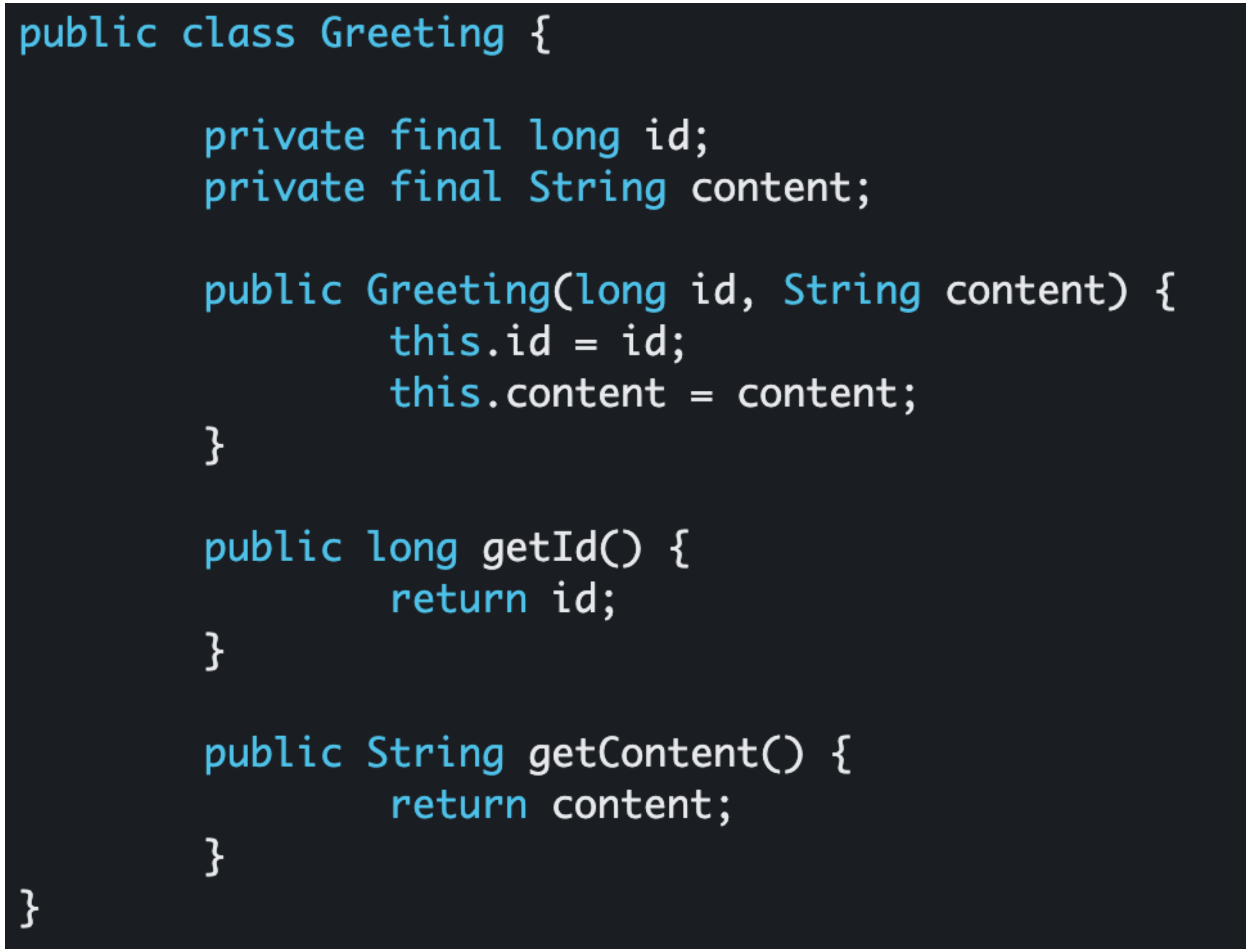
REST API (JSON)



```
{
    "id": 1,
    "content": "Hello, World!"
}
```

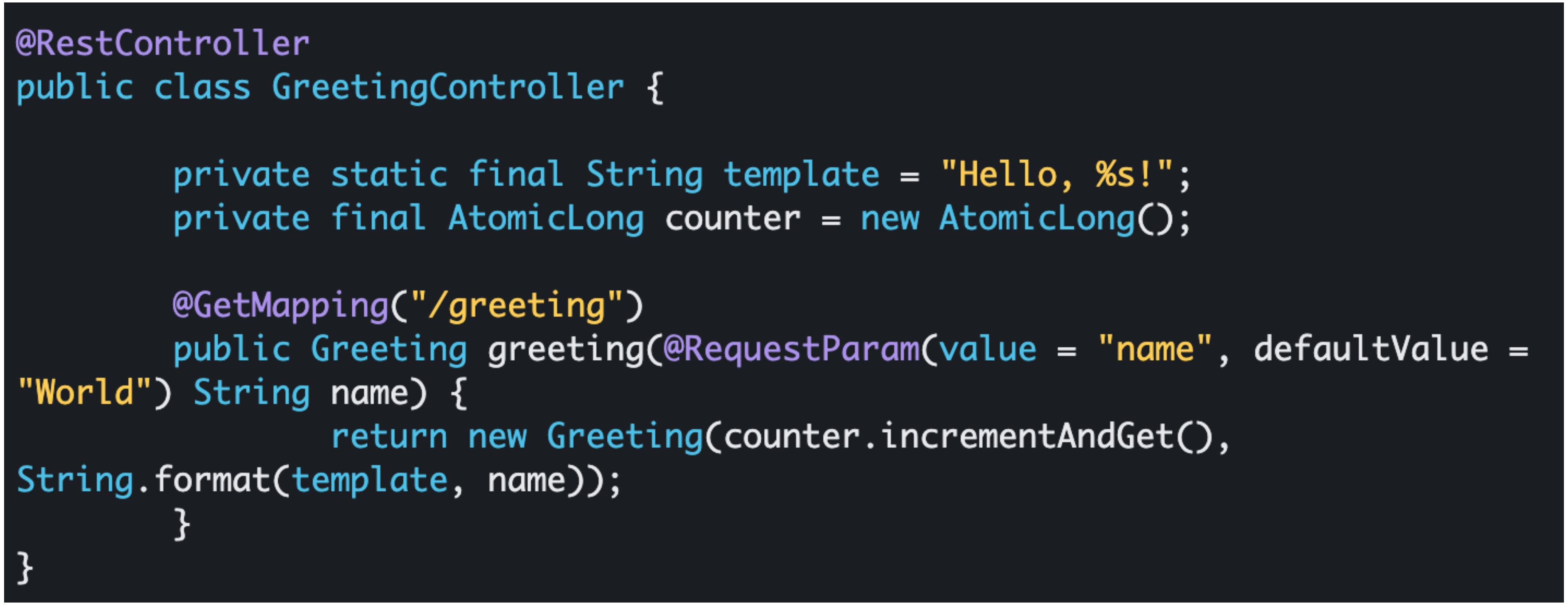
-service/

Adding model / entity Greeting



```
public class Greeting {

        private final long id;
        private final String content;

        public Greeting(long id, String content) {
                this.id = id;
                this.content = content;
        }

        public long getId() {
                return id;
        }

        public String getContent() {
                return content;
        }
}
```

Adding controller / endpoint



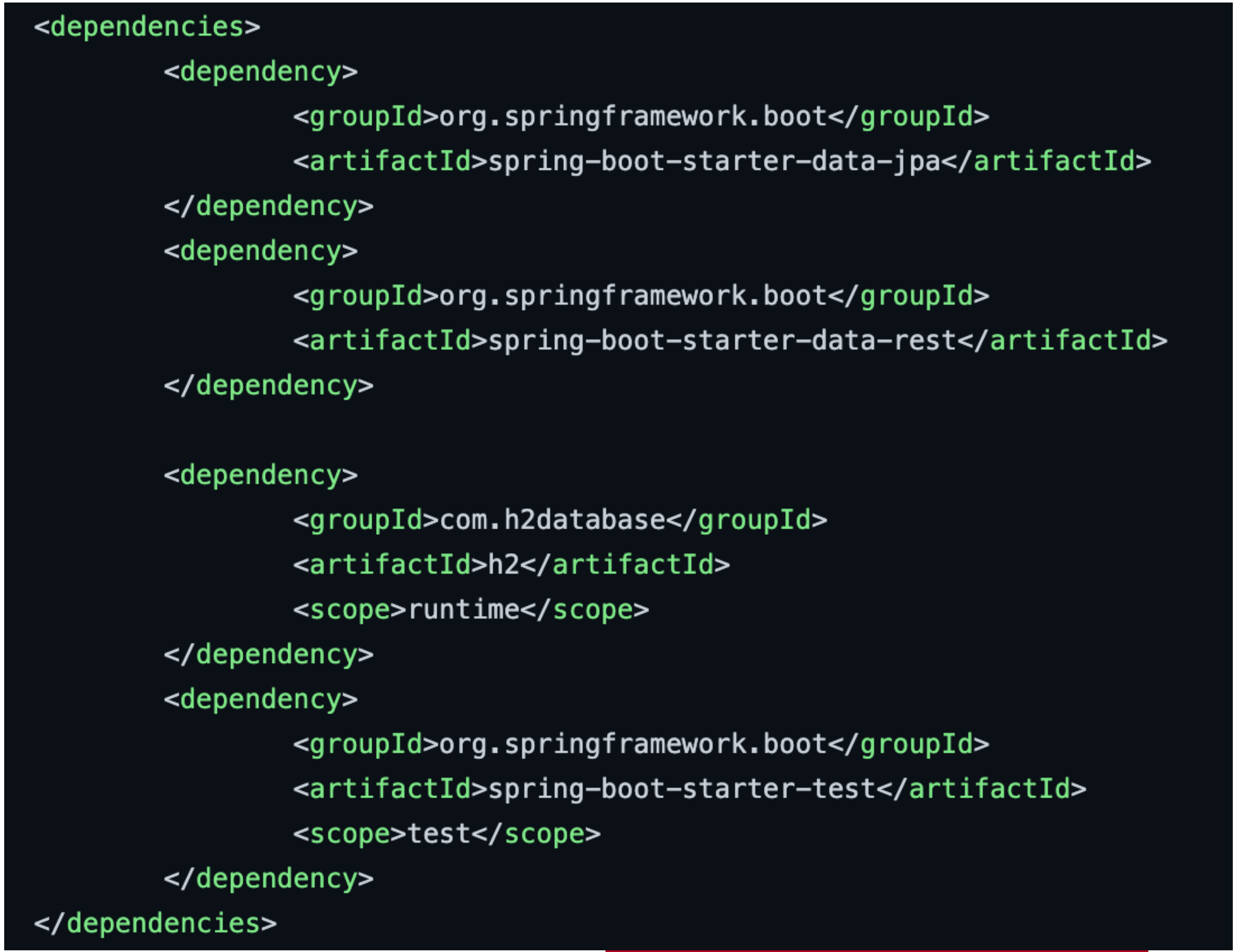
```
@RestController
public class GreetingController {

        private static final String template = "Hello, %s!";
        private final AtomicLong counter = new AtomicLong();

        @GetMapping("/greeting")
        public Greeting greeting(@RequestParam(value = "name", defaultValue =
"World") String name) {
                return new Greeting(counter.incrementAndGet(),
String.format(template, name));
        }
}
```

Adding Database and ORM/JPA

- https://spring.io/guides/gs/accessing-data-rest/

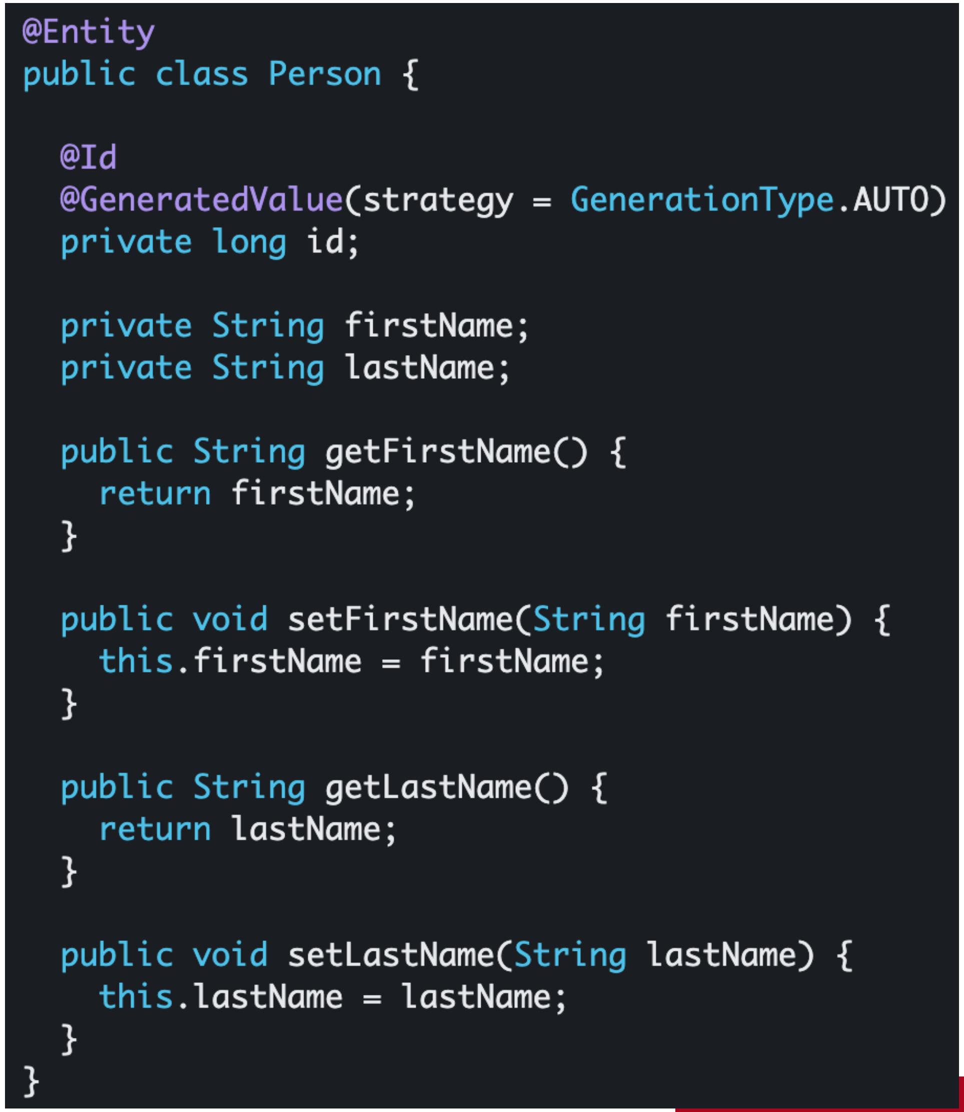
- ORM - object relational mapping
- Automated mapping from Objects to Relations in the database
- H2 Database

Maven dependency



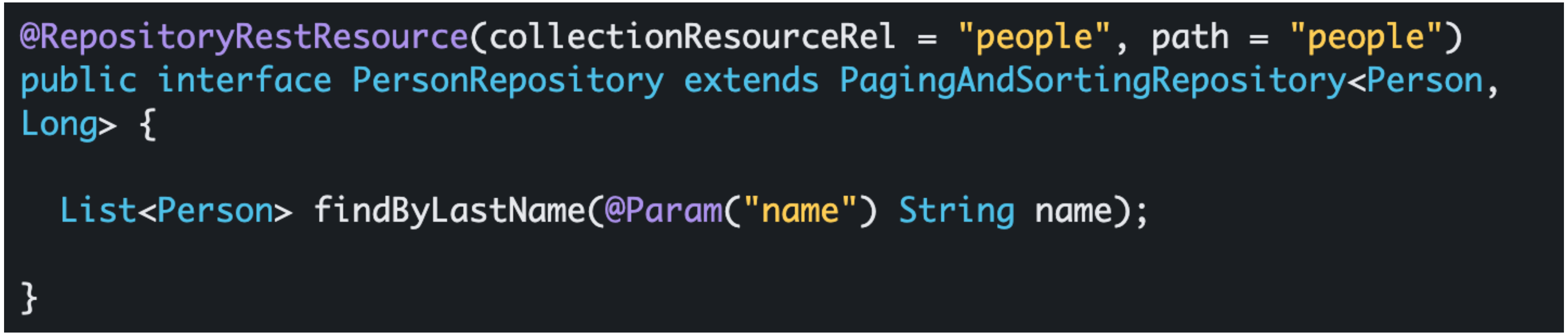
```
<dependencies>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-data-rest</artifactId>
        </dependency>

        <dependency>
                <groupId>com.h2database</groupId>
                <artifactId>h2</artifactId>
                <scope>runtime</scope>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
        </dependency>
</dependencies>
```

Data/entity with @Entity annotation

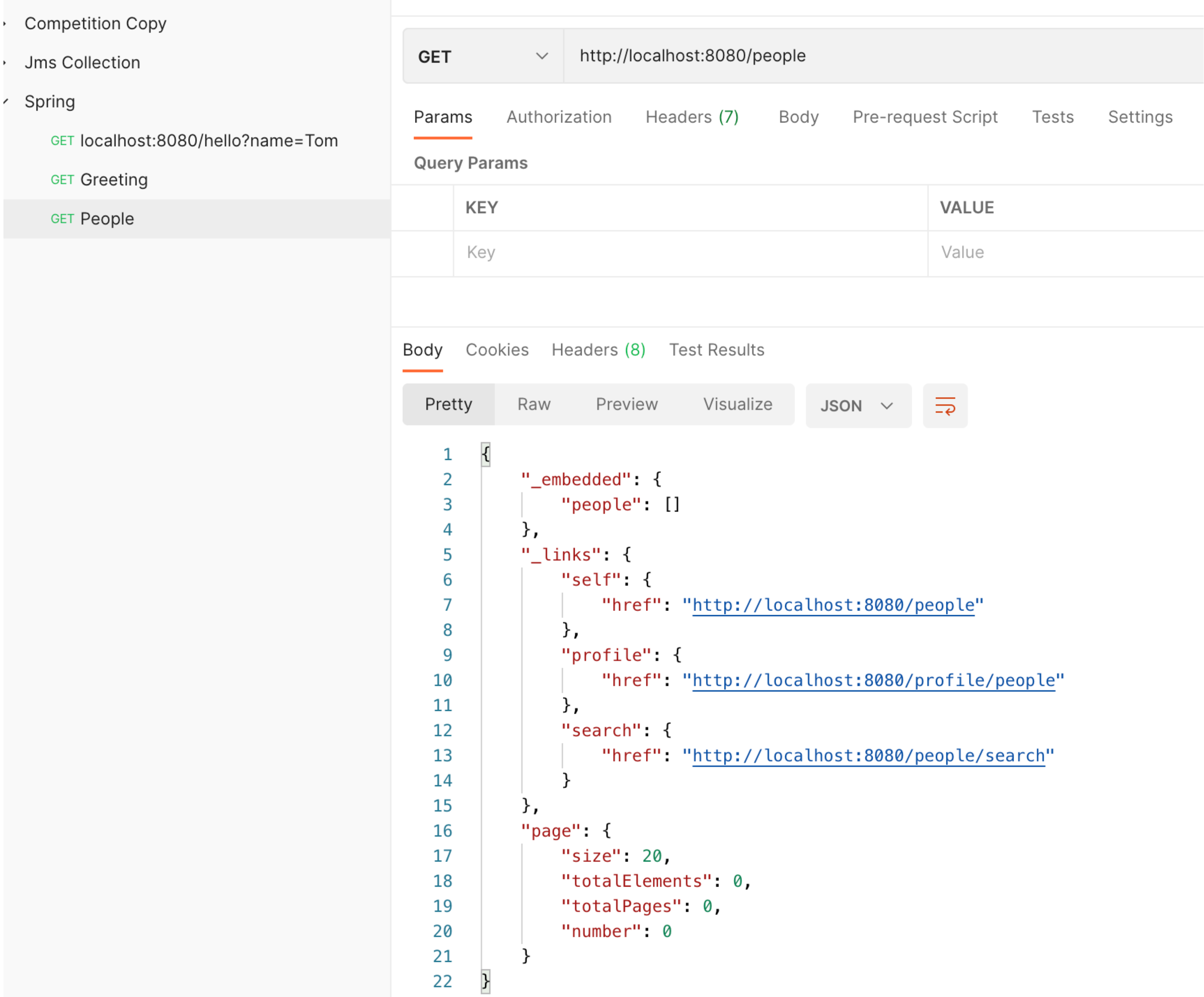


```
@Entity
public class Person {

  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  private long id;

  private String firstName;
  private String lastName;

  public String getFirstName() {
    return firstName;
  }

  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }

  public String getLastName() {
    return lastName;
  }

  public void setLastName(String lastName) {
    this.lastName = lastName;
  }
}
```

Access to Data / Repository (DAO)



```
@RepositoryRestResource(collectionResourceRel = "people", path = "people")
public interface PersonRepository extends PagingAndSortingRepository<Person,
Long> {

  List<Person> findByLastName(@Param("name") String name);

}
```



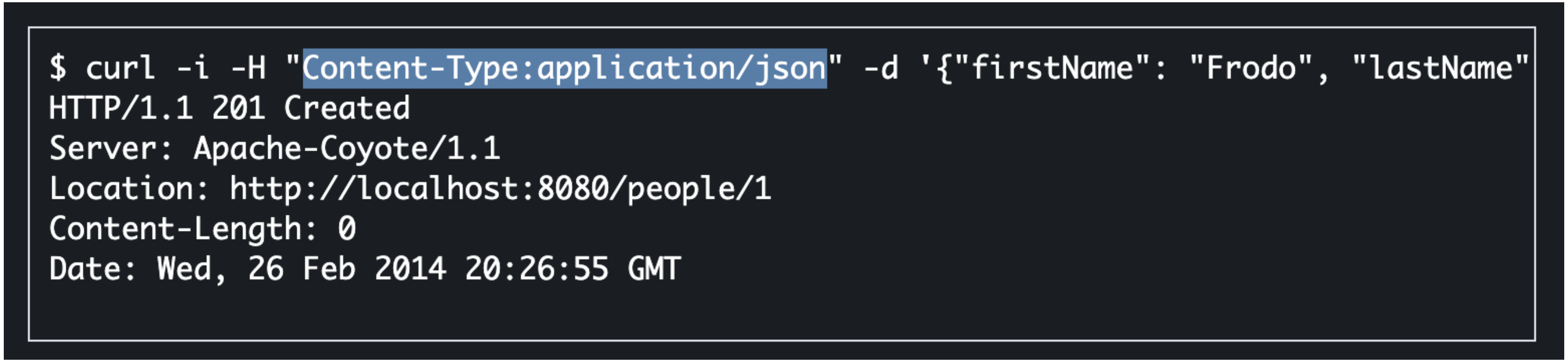
GET http://localhost:8080/people

Query Params

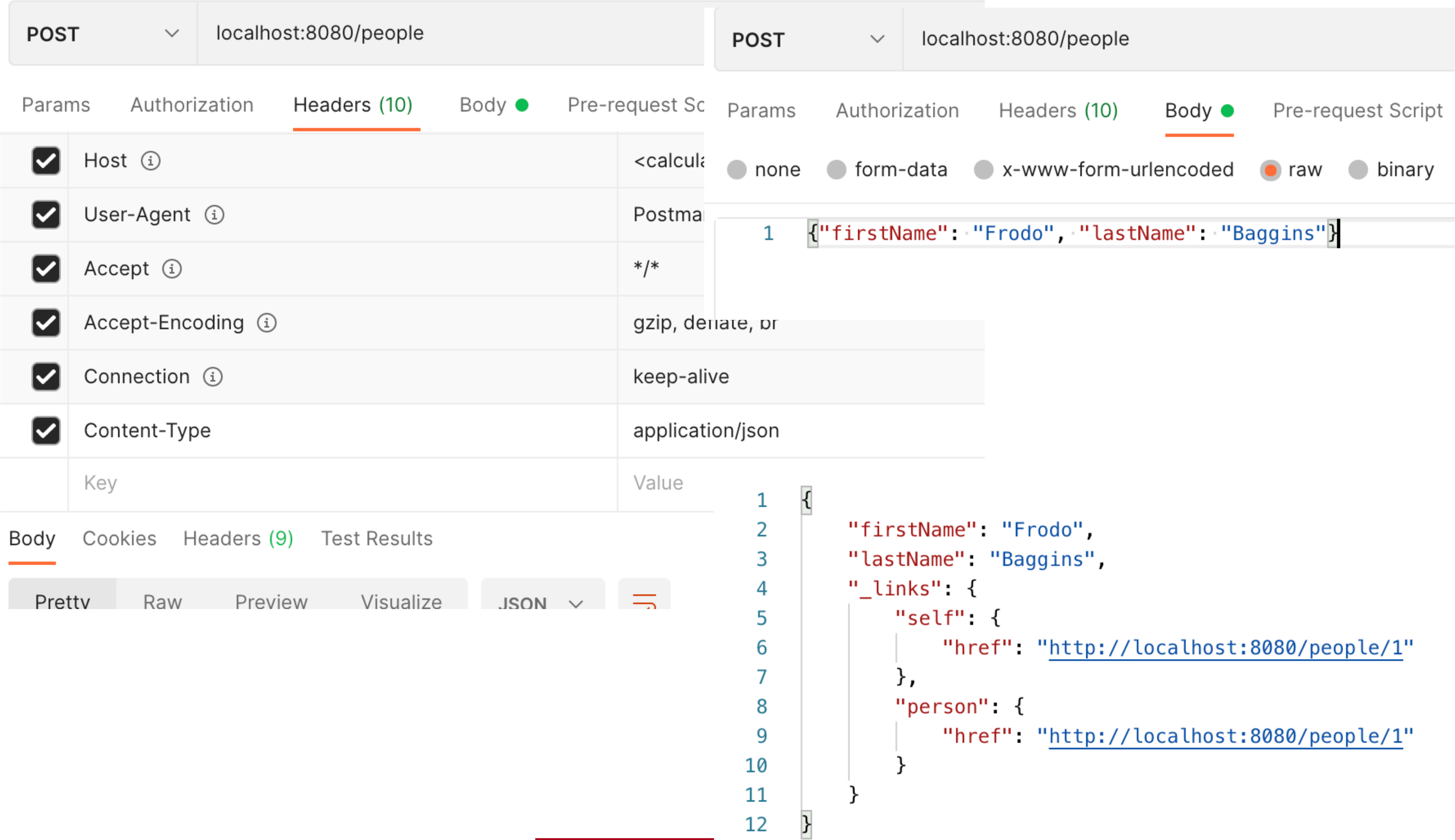
KEY	VALUE
Key	Value

```
{
    "_embedded": {
        "people": []
    },
    "_links": {
        "self": {
            "href": "http://localhost:8080/people"
        },
        "profile": {
            "href": "http://localhost:8080/profile/people"
        },
        "search": {
            "href": "http://localhost:8080/people/search"
        }
    },
    "page": {
        "size": 20,
        "totalElements": 0,
        "totalPages": 0,
        "number": 0
    }
}
```

Post Data



```
$ curl -i -H "Content-Type:application/json" -d '{"firstName": "Frodo", "lastName"
HTTP/1.1 201 Created
Server: Apache-Coyote/1.1
Location: http://localhost:8080/people/1
Content-Length: 0
Date: Wed, 26 Feb 2014 20:26:55 GMT
```



POST localhost:8080/people

	Key	Value
☑	Host	<calcula
☑	User-Agent	Postma
☑	Accept	*/*
☑	Accept-Encoding	gzip, deflate, br
☑	Connection	keep-alive
☑	Content-Type	application/json
	Key	Value

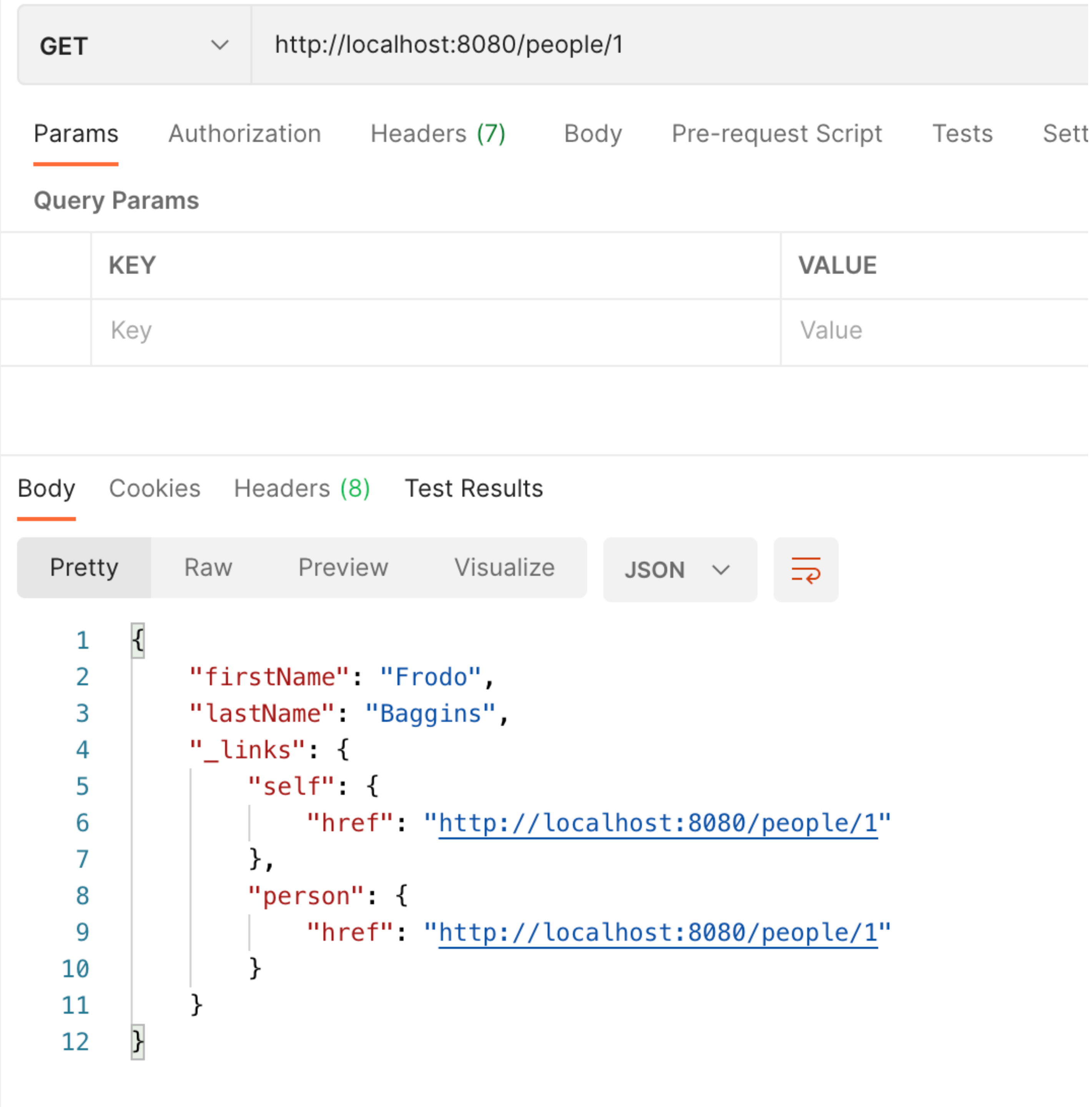
POST localhost:8080/people

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary

```
{"firstName": "Frodo", "lastName": "Baggins"}
```

```
{
    "firstName": "Frodo",
    "lastName": "Baggins",
    "_links": {
        "self": {
            "href": "http://localhost:8080/people/1"
        },
        "person": {
            "href": "http://localhost:8080/people/1"
        }
    }
}
```

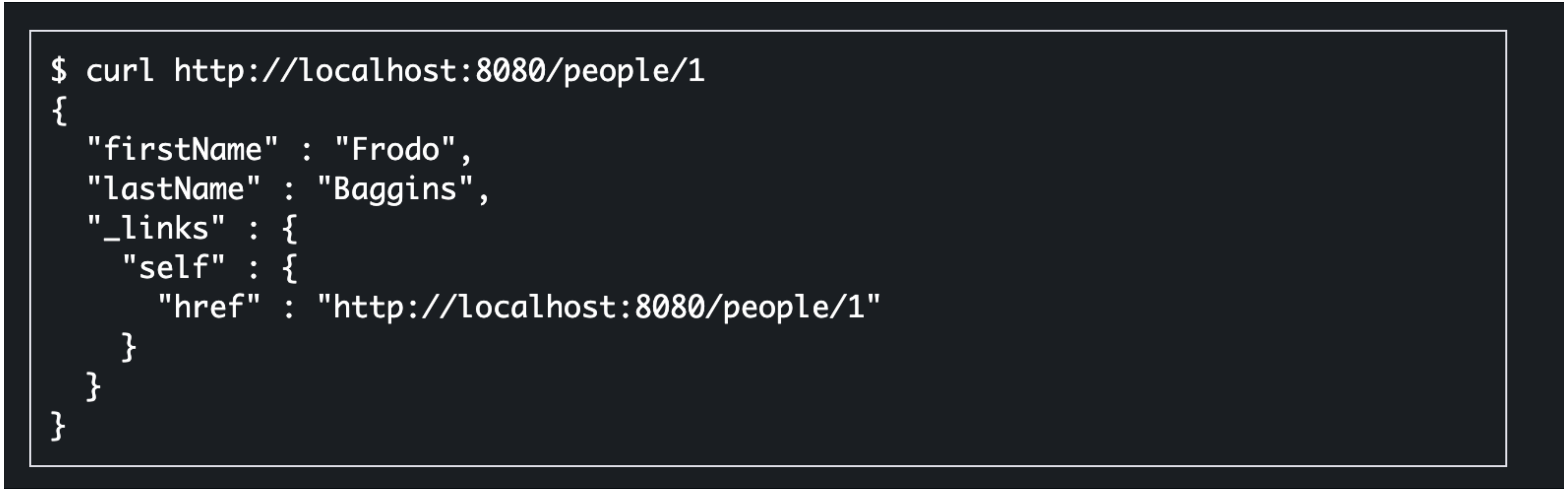
Read #1



GET http://localhost:8080/people/1

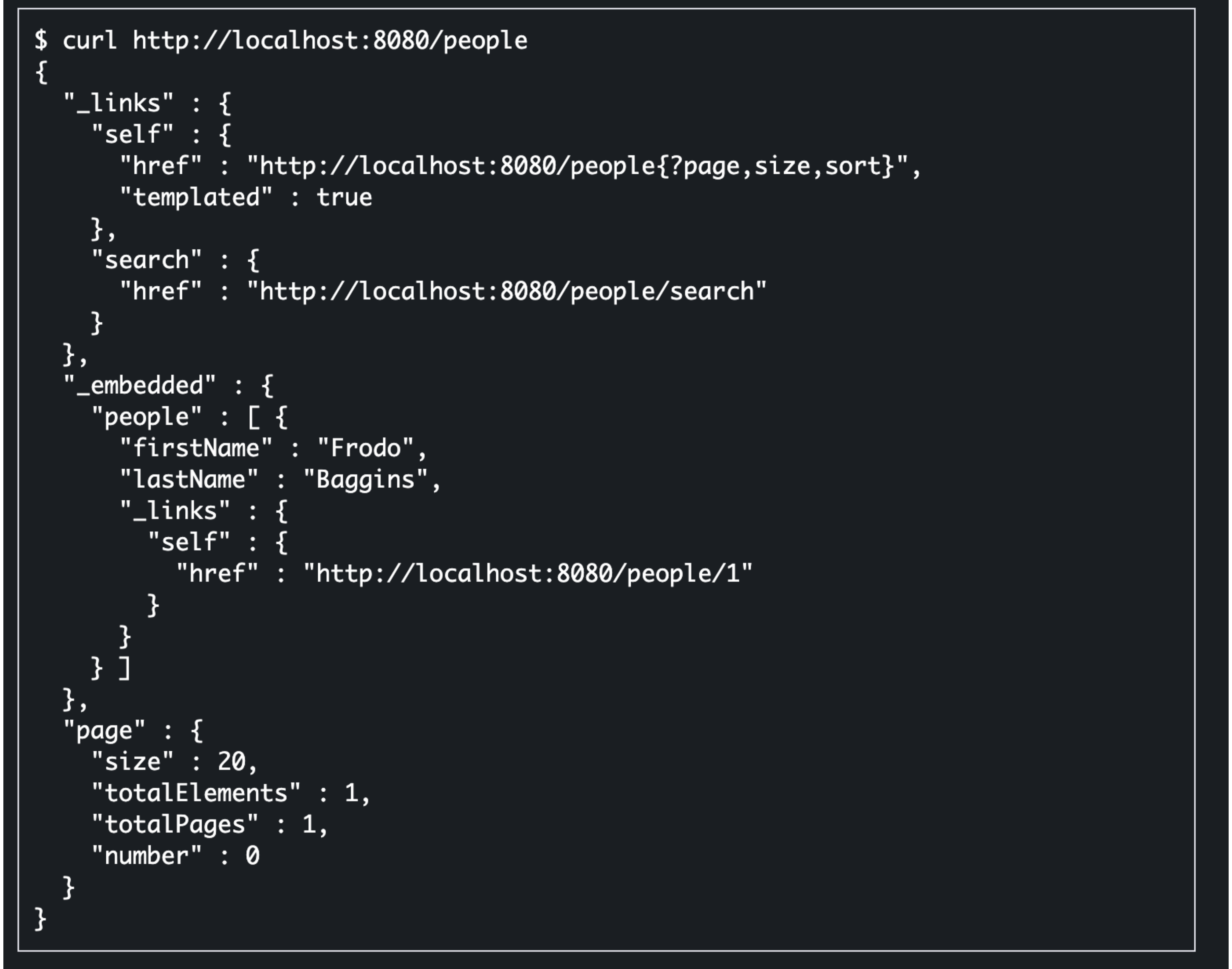
```
{
    "firstName": "Frodo",
    "lastName": "Baggins",
    "_links": {
        "self": {
            "href": "http://localhost:8080/people/1"
        },
        "person": {
            "href": "http://localhost:8080/people/1"
        }
    }
}
```

One



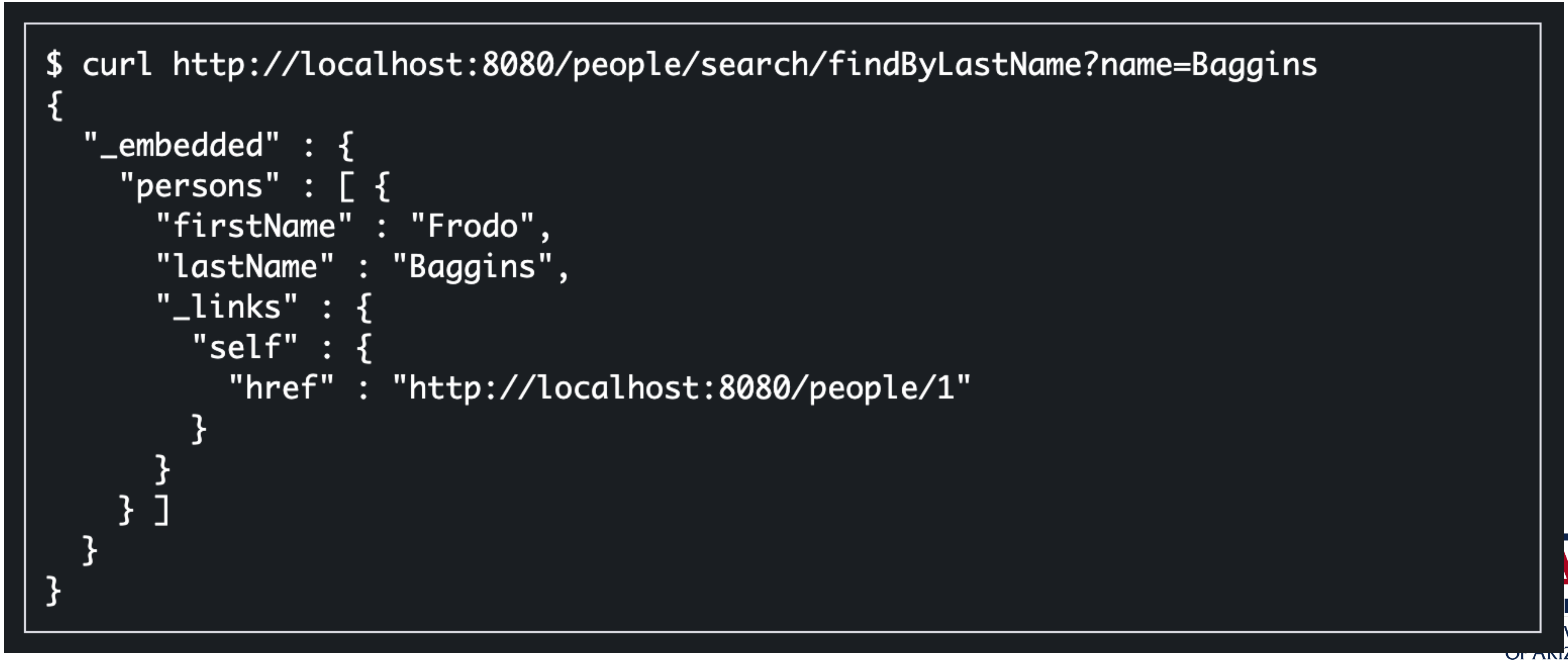
```
$ curl http://localhost:8080/people/1
{
  "firstName" : "Frodo",
  "lastName" : "Baggins",
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/people/1"
    }
  }
}
```

All



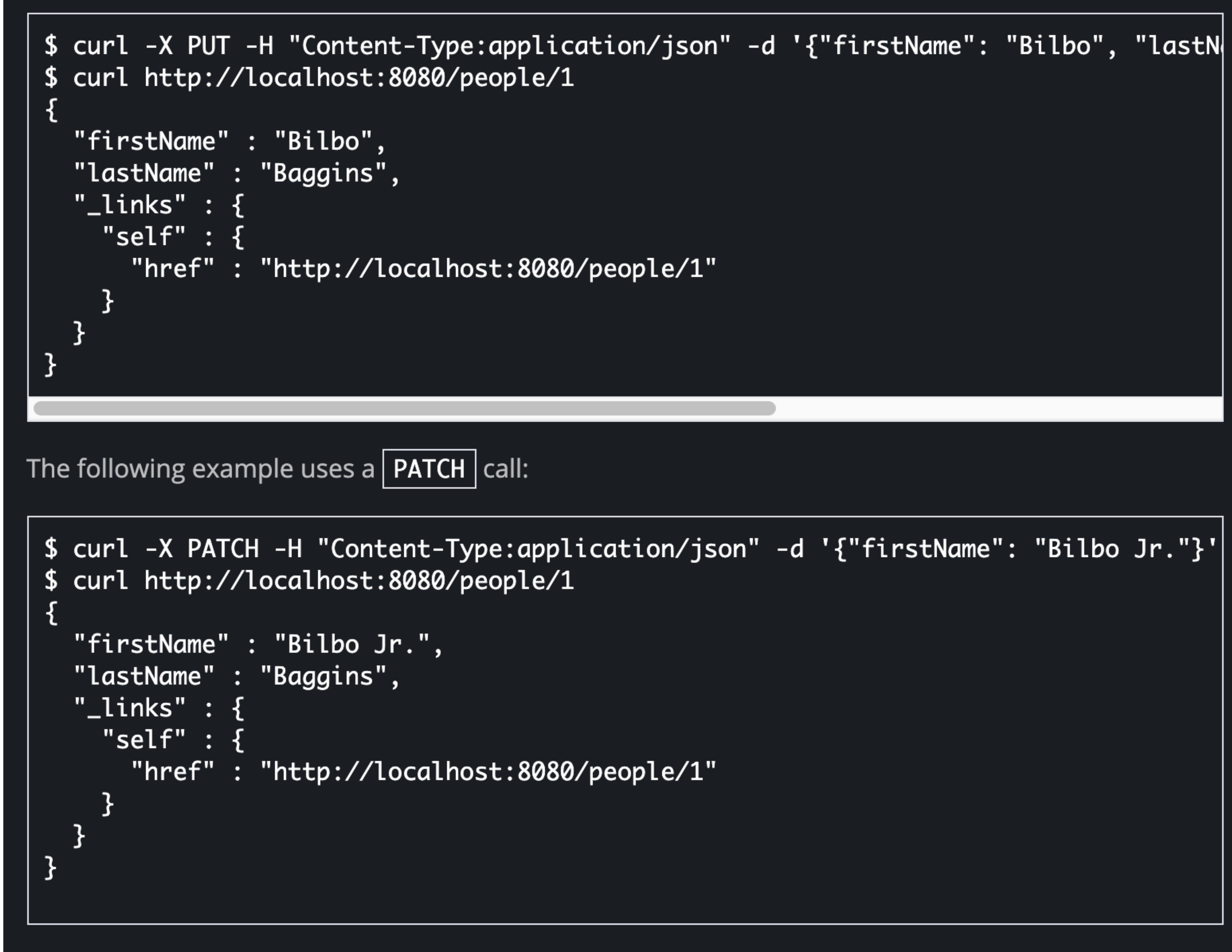
```
$ curl http://localhost:8080/people
{
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/people{?page,size,sort}",
      "templated" : true
    },
    "search" : {
      "href" : "http://localhost:8080/people/search"
    }
  },
  "_embedded" : {
    "people" : [ {
      "firstName" : "Frodo",
      "lastName" : "Baggins",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8080/people/1"
        }
      }
    } ]
  },
  "page" : {
    "size" : 20,
    "totalElements" : 1,
    "totalPages" : 1,
    "number" : 0
  }
}
```

Search



```
$ curl http://localhost:8080/people/search/findByLastName?name=Baggins
{
  "_embedded" : {
    "persons" : [ {
      "firstName" : "Frodo",
      "lastName" : "Baggins",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8080/people/1"
        }
      }
    } ]
  }
}
```

Update

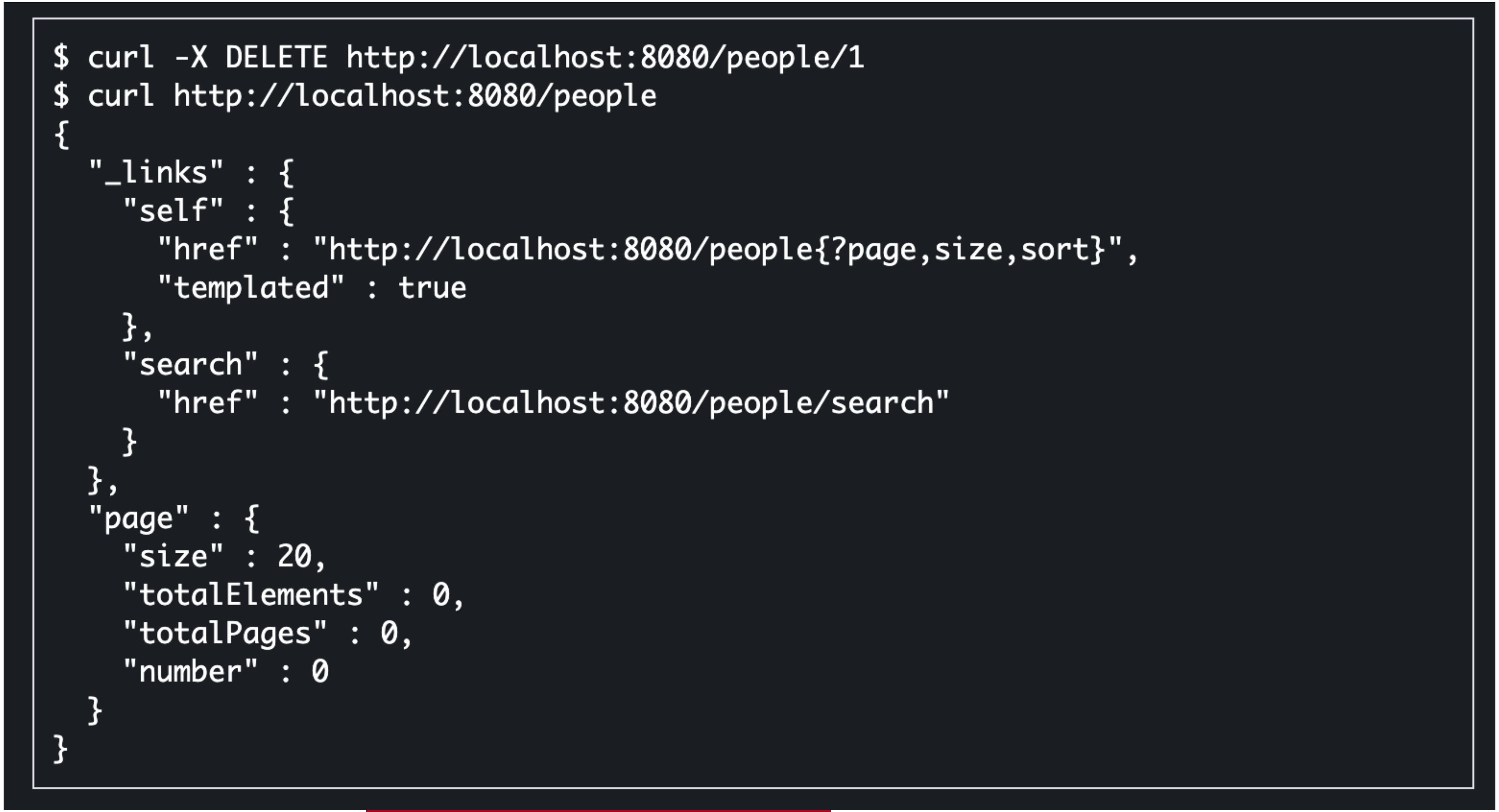


```
$ curl -X PUT -H "Content-Type:application/json" -d '{"firstName": "Bilbo", "lastN
$ curl http://localhost:8080/people/1
{
  "firstName" : "Bilbo",
  "lastName" : "Baggins",
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/people/1"
    }
  }
}
```

The following example uses a PATCH call:

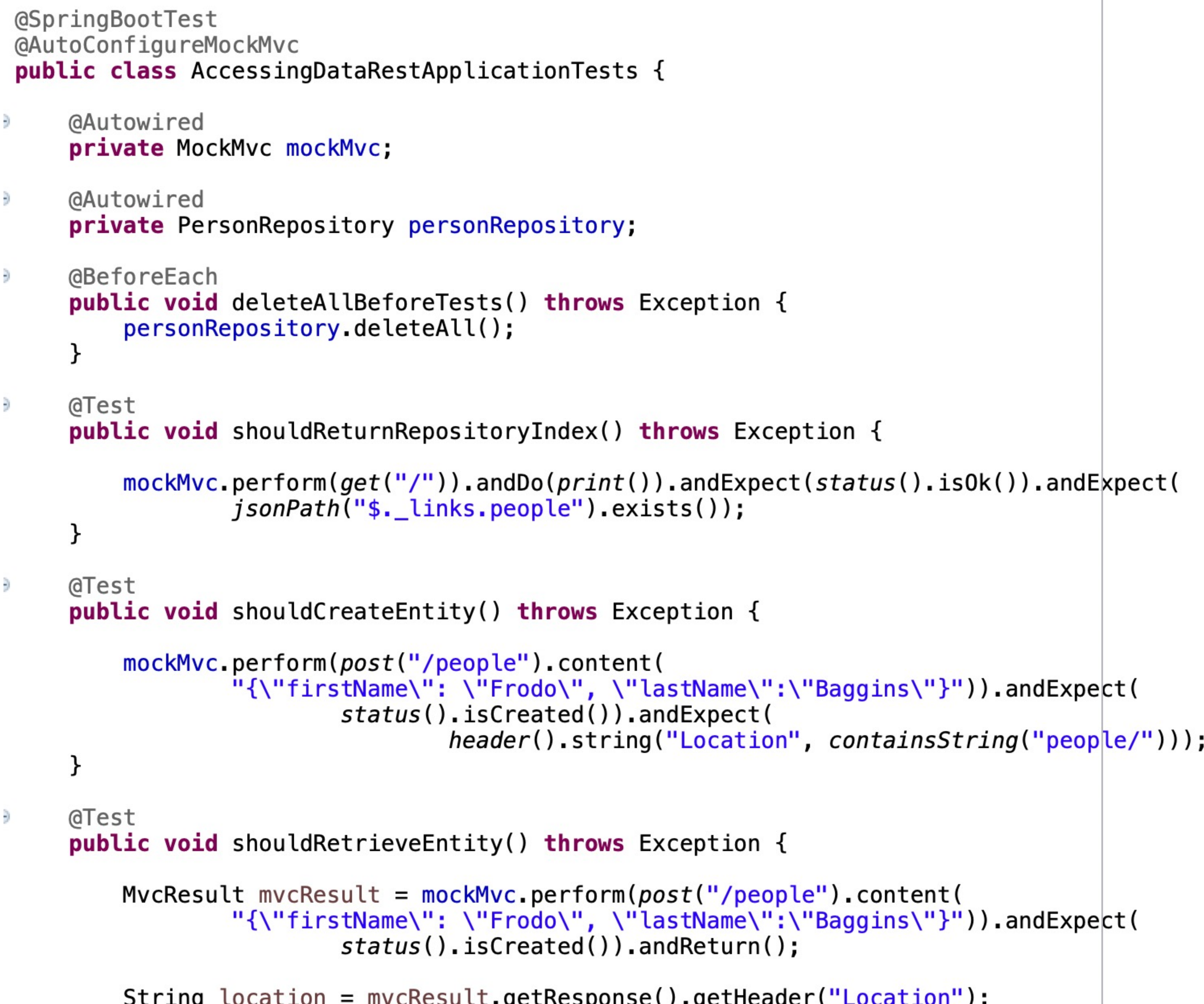
```
$ curl -X PATCH -H "Content-Type:application/json" -d '{"firstName": "Bilbo Jr."}'
$ curl http://localhost:8080/people/1
{
  "firstName" : "Bilbo Jr.",
  "lastName" : "Baggins",
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/people/1"
    }
  }
}
```

Delete



```
$ curl -X DELETE http://localhost:8080/people/1
$ curl http://localhost:8080/people
{
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/people{?page,size,sort}",
      "templated" : true
    },
    "search" : {
      "href" : "http://localhost:8080/people/search"
    }
  },
  "page" : {
    "size" : 20,
    "totalElements" : 0,
    "totalPages" : 0,
    "number" : 0
  }
}
```

Testing / JUnit



```java
@SpringBootTest
@AutoConfigureMockMvc
public class AccessingDataRestApplicationTests {

    @Autowired
    private MockMvc mockMvc;

    @Autowired
    private PersonRepository personRepository;

    @BeforeEach
    public void deleteAllBeforeTests() throws Exception {
        personRepository.deleteAll();
    }

    @Test
    public void shouldReturnRepositoryIndex() throws Exception {

        mockMvc.perform(get("/")).andDo(print()).andExpect(status().isOk()).andExpect(
                jsonPath("$._links.people").exists());
    }

    @Test
    public void shouldCreateEntity() throws Exception {

        mockMvc.perform(post("/people").content(
                "{\"firstName\": \"Frodo\", \"lastName\":\"Baggins\"}")).andExpect(
                        status().isCreated()).andExpect(
                                header().string("Location", containsString("people/")));
    }

    @Test
    public void shouldRetrieveEntity() throws Exception {

        MvcResult mvcResult = mockMvc.perform(post("/people").content(
                "{\"firstName\": \"Frodo\", \"lastName\":\"Baggins\"}")).andExpect(
                        status().isCreated()).andReturn();

        String location = mvcResult.getResponse().getHeader("Location");
```

Spring MVC for web apps

- https://www.baeldung.com/spring-mvc-form-tutorial
- https://www.baeldung.com/spring-mvc-tutorial

- https://spring.io/guides/gs/handling-form-submission/
- https://www.baeldung.com/thymeleaf-iteration

Spring Guides

- https://spring.io/quickstart
- https://spring.io/guides/gs/rest-service/
- https://spring.io/guides/gs/accessing-data-rest/

THE UNIVERSITY OF ARIZONA

Questions

THE UNIVERSITY OF ARIZONA