CS544 EA

Integration

REST: sending / receiving XML

Send / Receive XML

- Enabling XML is simply adding a dependency

```
<dependency>
     <groupId>com.fasterxml.jackson.core</groupId>
     <artifactId>jackson-dataformat-xml</artifactId>
</dependency>
```

- With this added our webservice will be able to do both JSON and XML

Content Negotiation

- HTTP headers are used to indicate what the incoming and outgoing data types should be
  - **Content-Type** indicates what the browser sends
  - **Accept** indicates what the browser wants

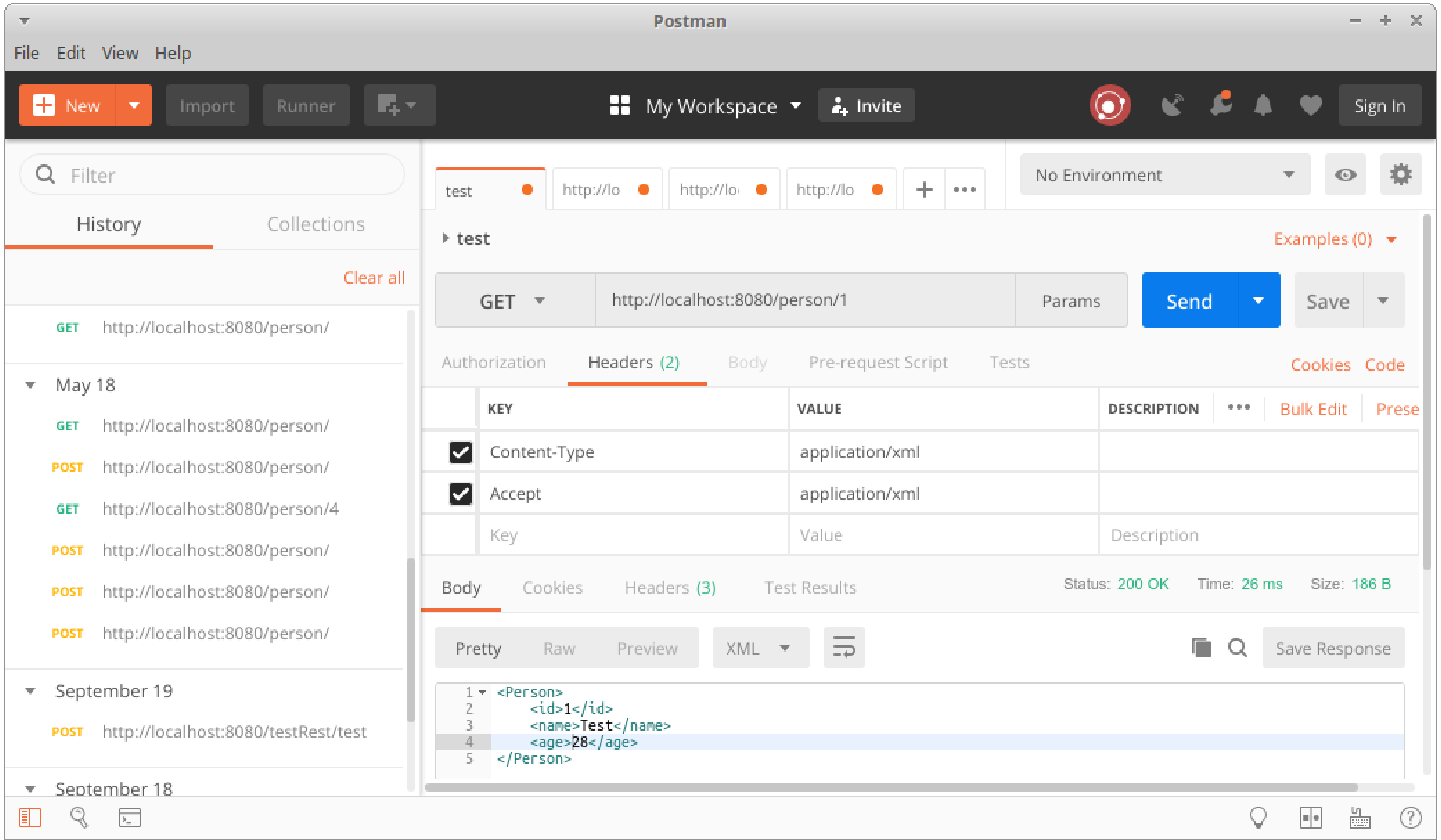
- The mime type for XML is:
  - application/xml

Controller

```
@RestController
public class PersonController {
      @Autowired
      private PersonService personService;

      @GetMapping("/person/")
      public List<Person> getAll() {
            return personService.getAll();
      }

      @GetMapping("/person/{id}")
      public Person get(@PathVariable long id) {
            return personService.get(id);
      }

      @PostMapping("/person/")
      public RedirectView post(@RequestBody Person person) {
            long id = personService.add(person);
            return new RedirectView("/person/" + id);
      }

      @PutMapping("/person/{id}")
      public void put(@PathVariable long id, @RequestBody Person person) {
            if (id != person.getId()) { throw new IllegalArgumentException(); }
            personService.update(person);
      }

      @DeleteMapping("/person/{id}")
      public void delete(@PathVariable long id) {
            personService.delete(id);
      }
}
```

By not specifying produces / consumes a controller method can be used for both XML and JSON

Example Receiving



History:

GET http://localhost:8080/person/

May 18
- GET http://localhost:8080/person/
- POST http://localhost:8080/person/
- GET http://localhost:8080/person/4
- POST http://localhost:8080/person/
- POST http://localhost:8080/person/
- POST http://localhost:8080/person/

September 19
- POST http://localhost:8080/testRest/test

September 18

Tabs: test | http://lo | http://lo | http://lo

test

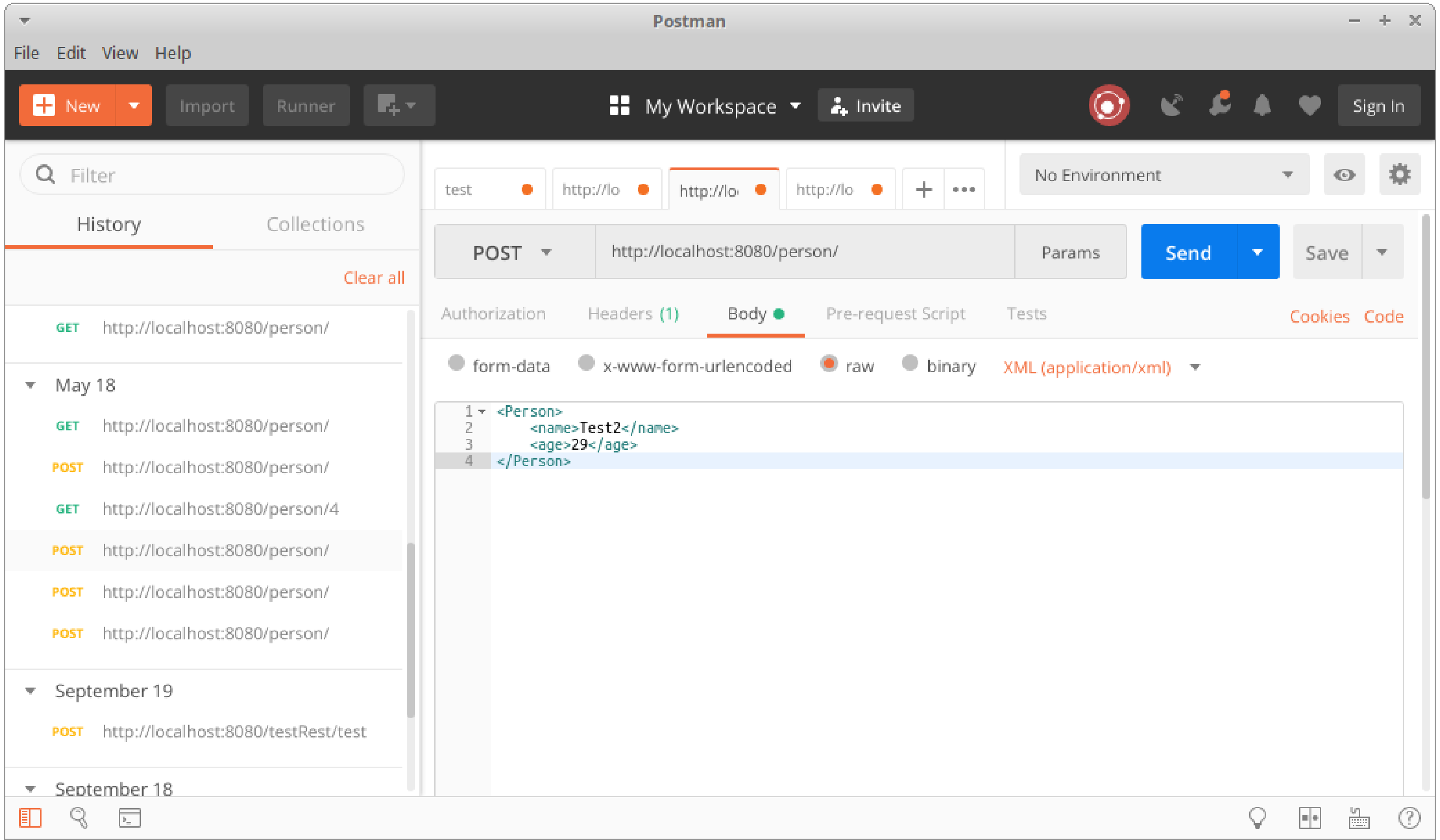
GET http://localhost:8080/person/1

	KEY	VALUE	DESCRIPTION
✓	Content-Type	application/xml	
✓	Accept	application/xml	
	Key	Value	Description

Body — Status: 200 OK Time: 26 ms Size: 186 B

```
<Person>
    <id>1</id>
    <name>Test</name>
    <age>28</age>
</Person>
```

Example Sending



GET http://localhost:8080/person/

May 18

GET http://localhost:8080/person/
POST http://localhost:8080/person/
GET http://localhost:8080/person/4
POST http://localhost:8080/person/
POST http://localhost:8080/person/
POST http://localhost:8080/person/

September 19

POST http://localhost:8080/testRest/test

September 18

POST http://localhost:8080/person/

raw XML (application/xml)

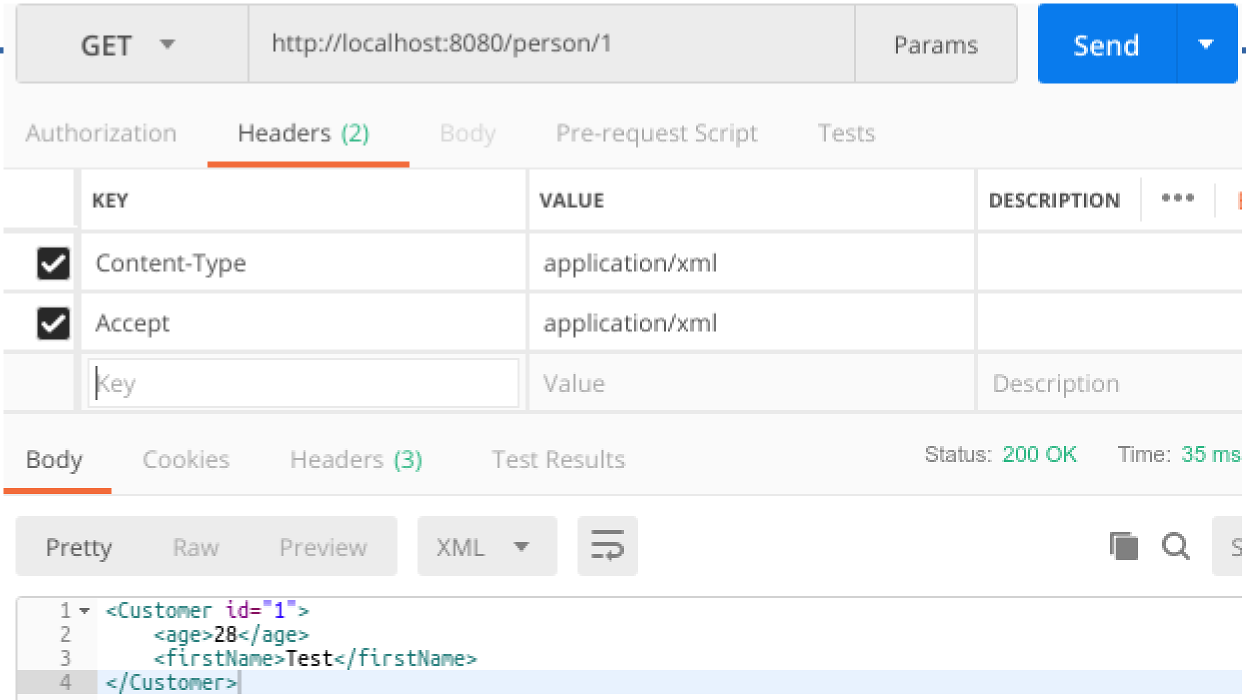
```xml
<Person>
    <name>Test2</name>
    <age>29</age>
</Person>
```

JAXB Mapping

- The Object to XML Mapping (OXM) defaults to:
  - Class get its own tag
  - Every property gets its own tag inside it
  - An object inside an object is nested
- If you want to change the name of a tag
  - Or use an XML attribute
  - Map them with annotations

Basic OXM Mapping

```java
@JacksonXmlRootElement(localName = "Customer")
public class Person {
    @JacksonXmlProperty(isAttribute = true)
    private Long id;
    @JacksonXmlProperty(localName = "firstName")
    private String name;
    private int age;
```



http://localhost:8080/person/1

KEY	VALUE	DESCRIPTION
Content-Type	application/xml	
Accept	application/xml	
Key	Value	Description

Status: 200 OK Time: 35 ms

```xml
<Customer id="1">
    <age>28</age>
    <firstName>Test</firstName>
</Customer>
```