Mapping Payload Data to Objects

Andrejs Doronins

```
// using JsonPath
int val = body.jsonPath().get(“some.path”);

// using the Fluent Interface
.rootPath(“some.path”)
    .body(“fieldX”, equalTo(y))
```

Overview

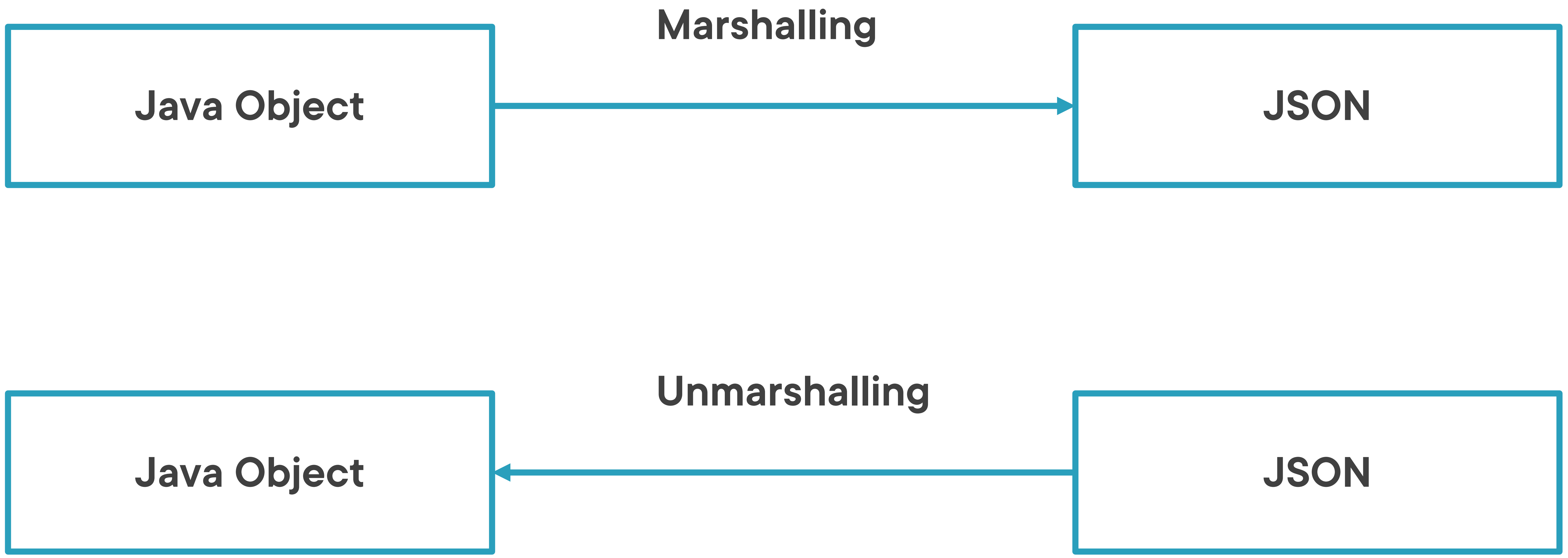
Converting JSON into Java objects

Unmarshalling

Marshalling

The process of transforming the memory representation of an object to a data format suitable for storage or transmission

Wikipedia.org


Marshalling
Java Object
JSON
Unmarshalling
Java Object
JSON

Course

Working with XML in Java Using JAXB

Jesper De Jong

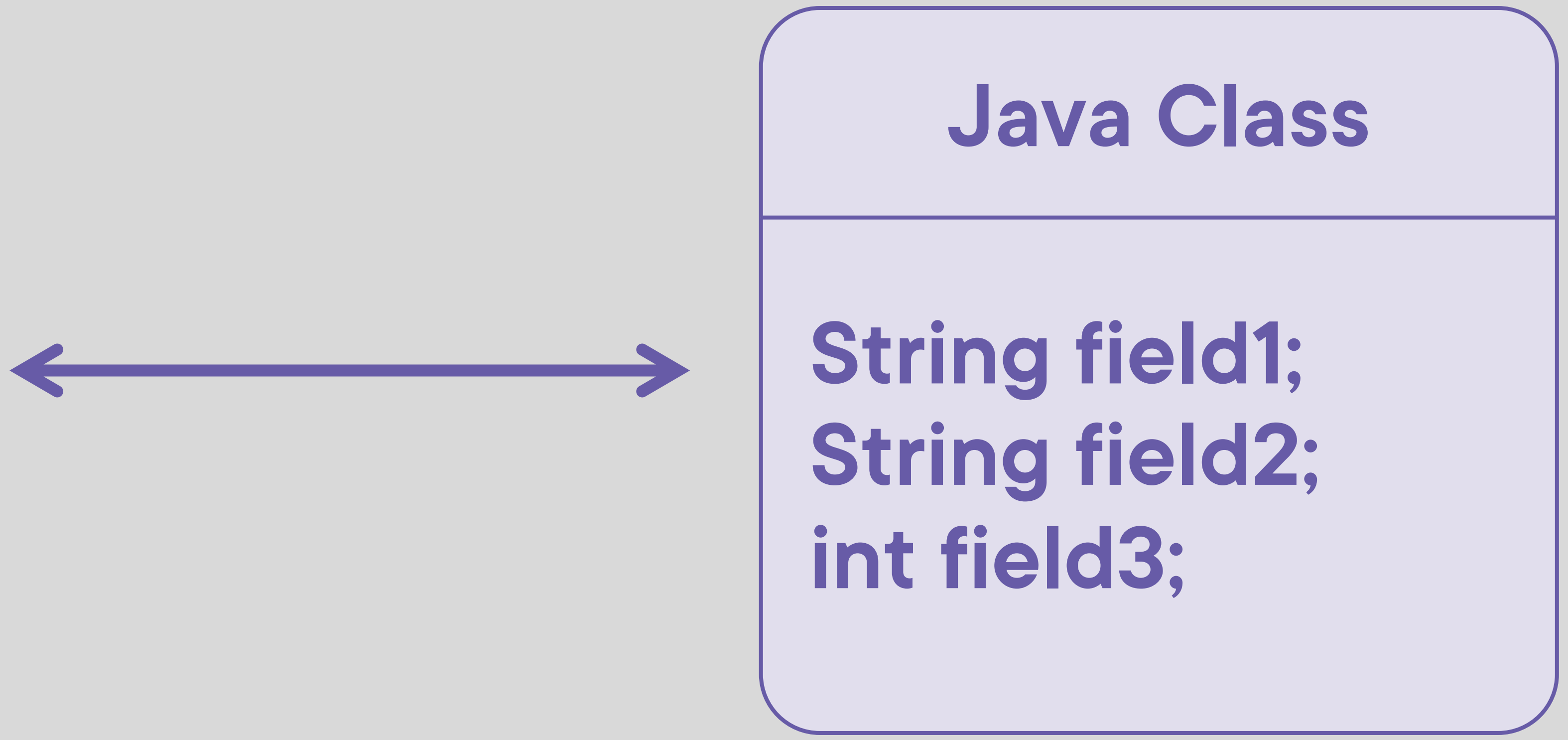
JSON Libraries

Jackson

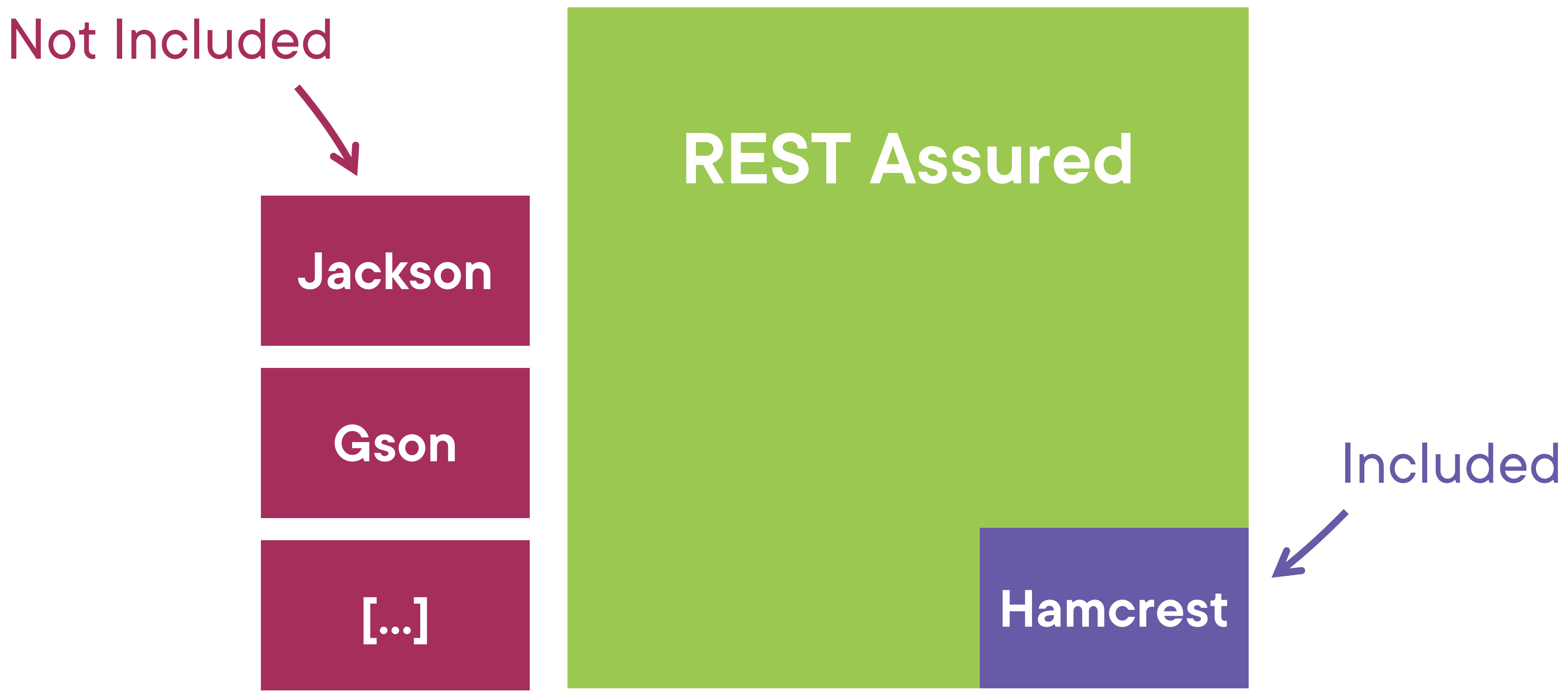
Gson

Mapping

```
{
    field1: value1
    field2: value2
    field3: value3
}
```



```
String field1;
String field2;
int field3;
```


Not Included
Jackson
Gson
[...]
REST Assured
Hamcrest
Included

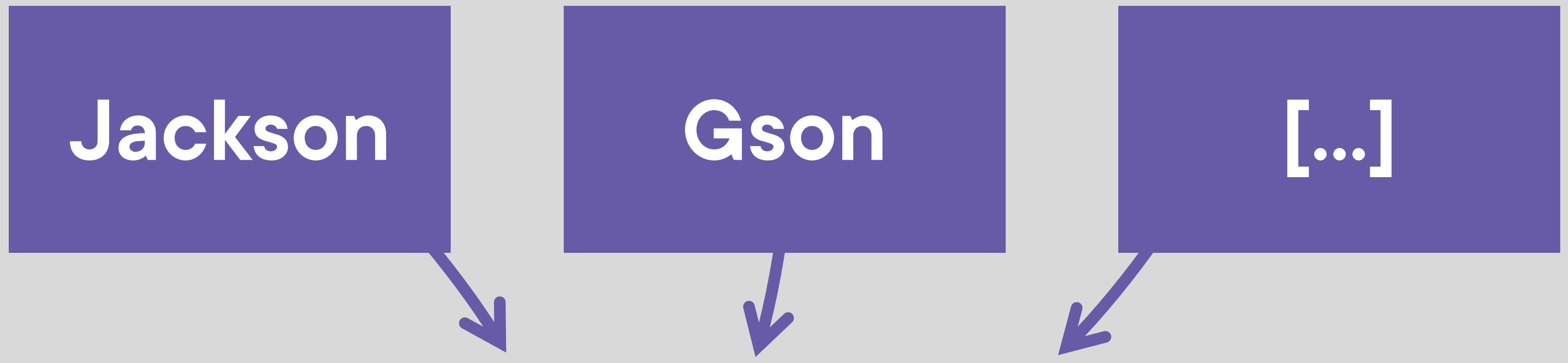
Headers:

- header(String s, String s)
- header(String s, Matcher m)
- […]

Body:

- body(Matcher m, Matcher ms)
- body(String s, ResponseAwareMatcher m)
- […]

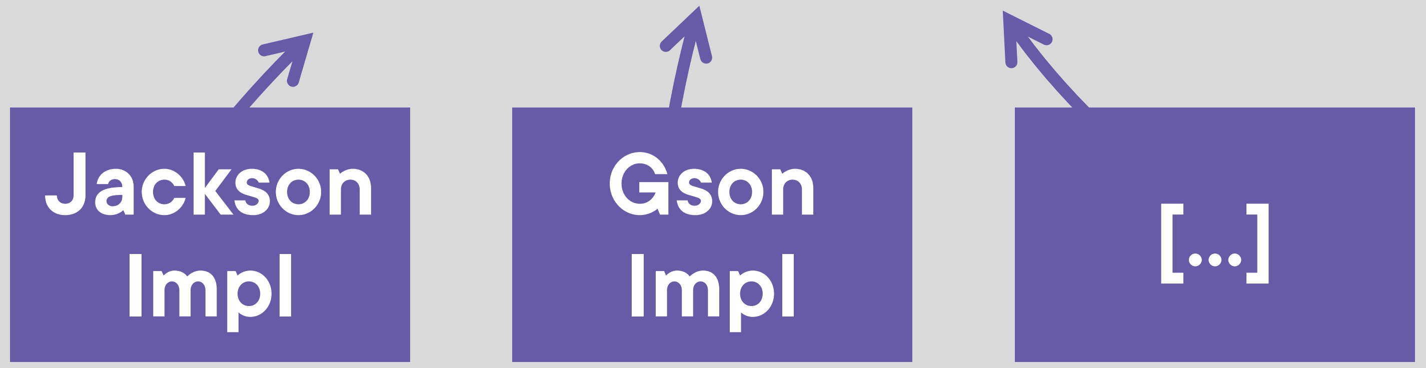
```
.as(Class<T> cls)
```



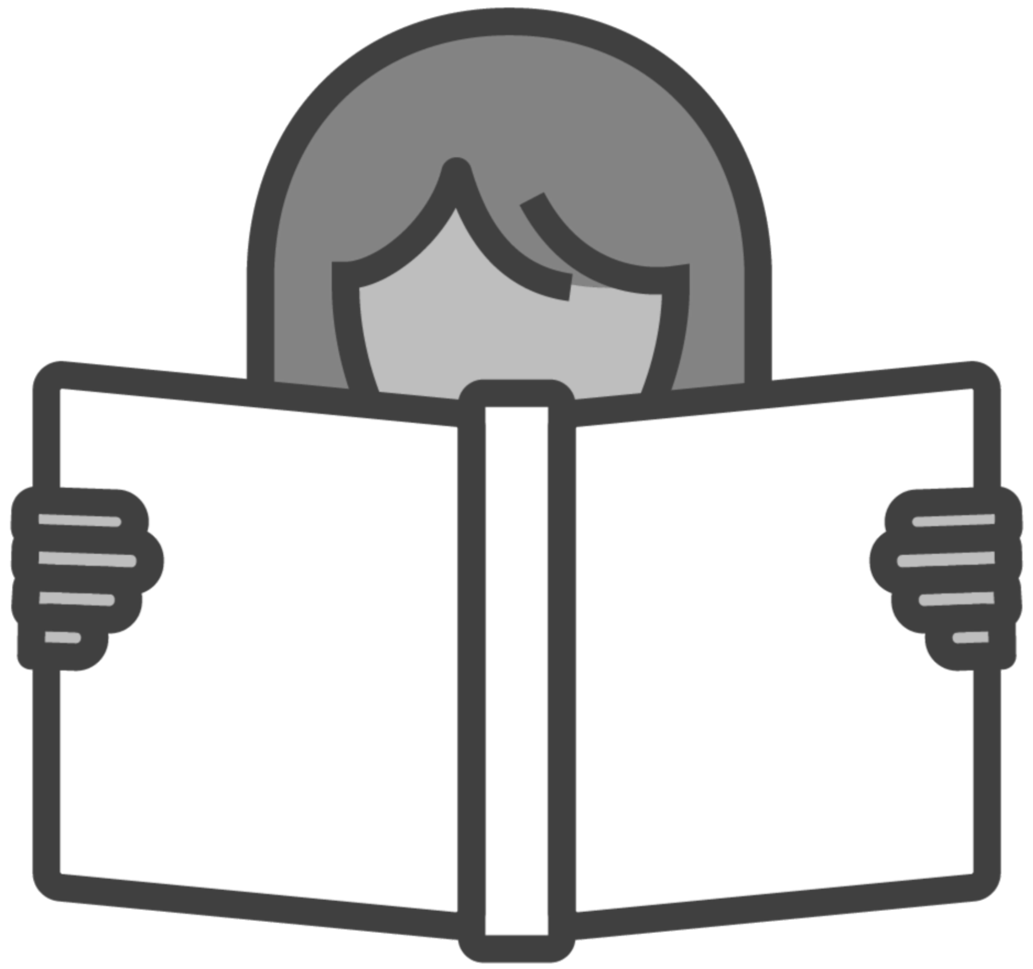
```
.as(Class<T> cls, ObjectMapperType mapperType)

.as(Class<T> cls, ObjectMapper mapper)
```

```
com.fasterxml.jackson.databind.ObjectMapper
```

```
io.restassured.mapper.ObjectMapper
```



Courses



Working with XML in Java Using JAXB

Java: JSON Databinding with Jackson

Getting Started with Web API Test Automation in Java (Module: Writing Advanced GET Tests for Response Payload)

Summary

Unmarshalling concept

Jackson and other libraries

Up Next:
Working with Other HTTP Requests