

REST Assured Fundamentals

Introducing Web API Testing



Andrejs Doronins

Why learn REST Assured?



Why should I learn it?
To gain API testing skills



Is there demand?
Yes!



We could test the
API manually...



OK, but which tool
do we use?

Speed and efficiency
matter!

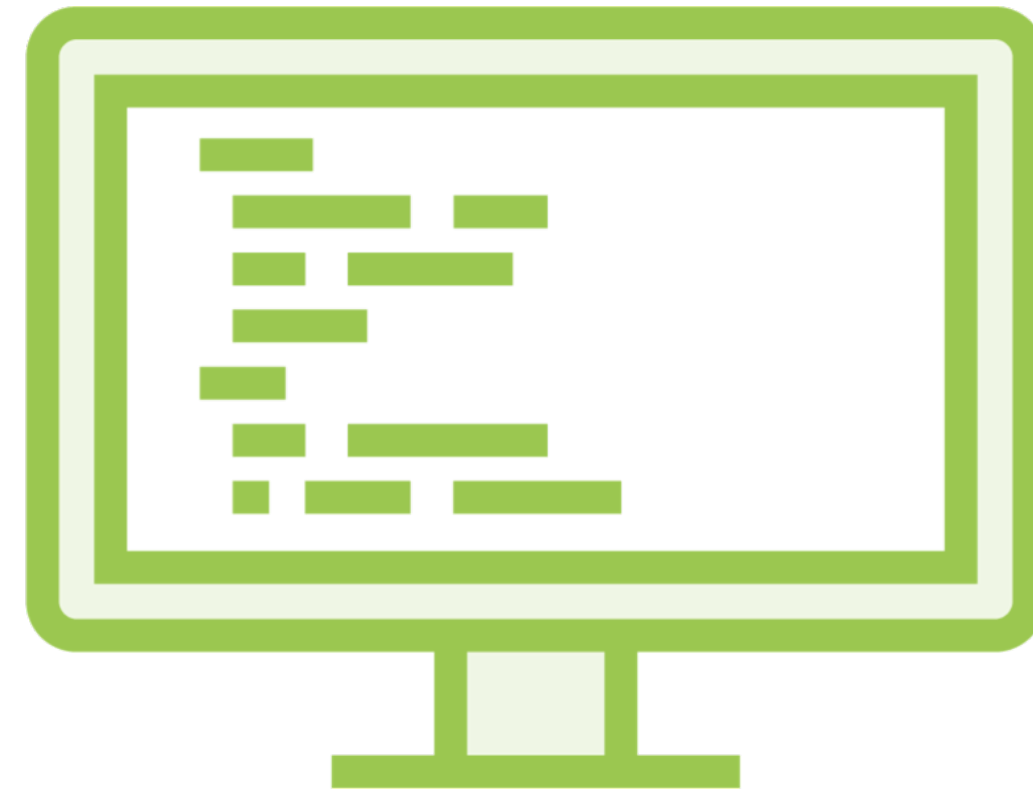


API Testing Tools



GUI-based

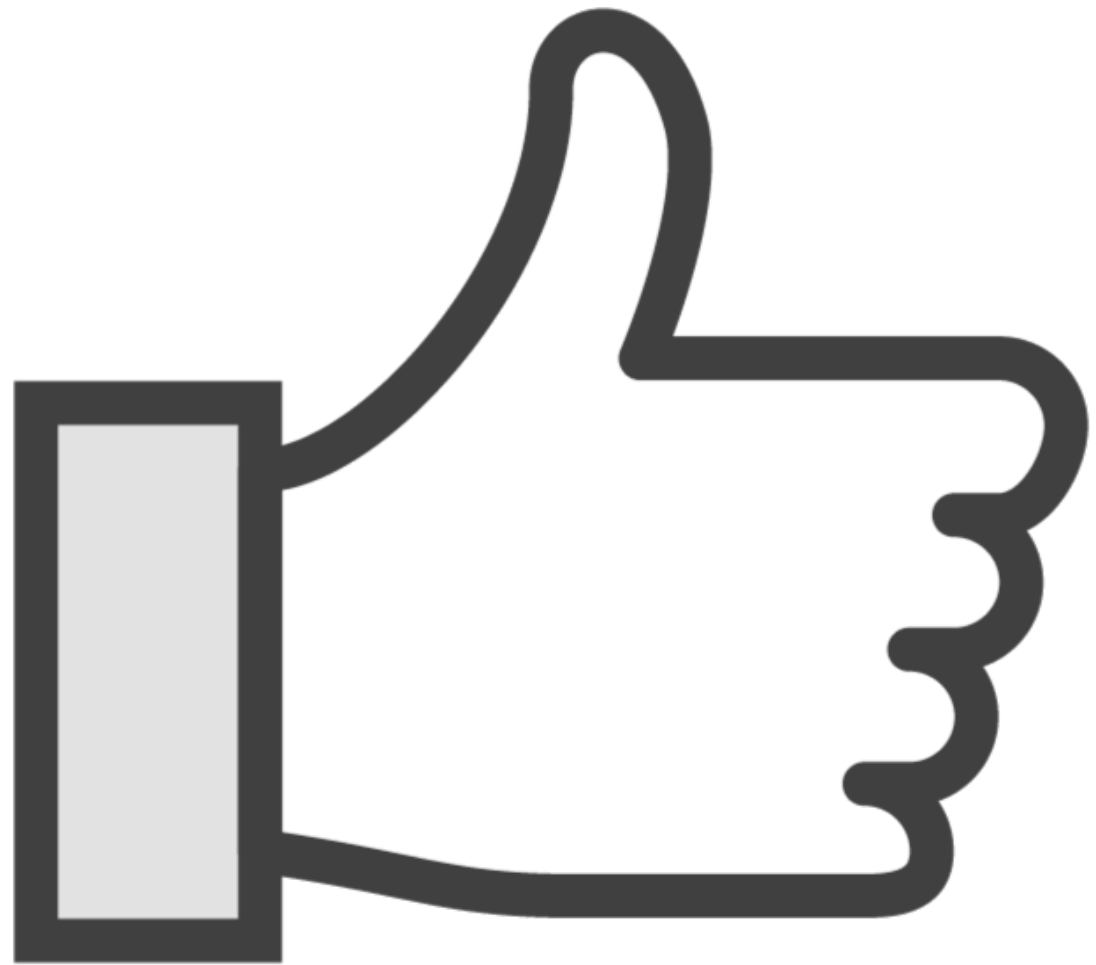
(e.g. SoapUI or Postman)



Code-based

(frameworks and libraries)

Advantages of GUI Tools



- **Lower learning curve**
- **Works out-of-the-box**
- **Suitable for both simple and complex APIs**

GUI Tools May Not Offer Everything

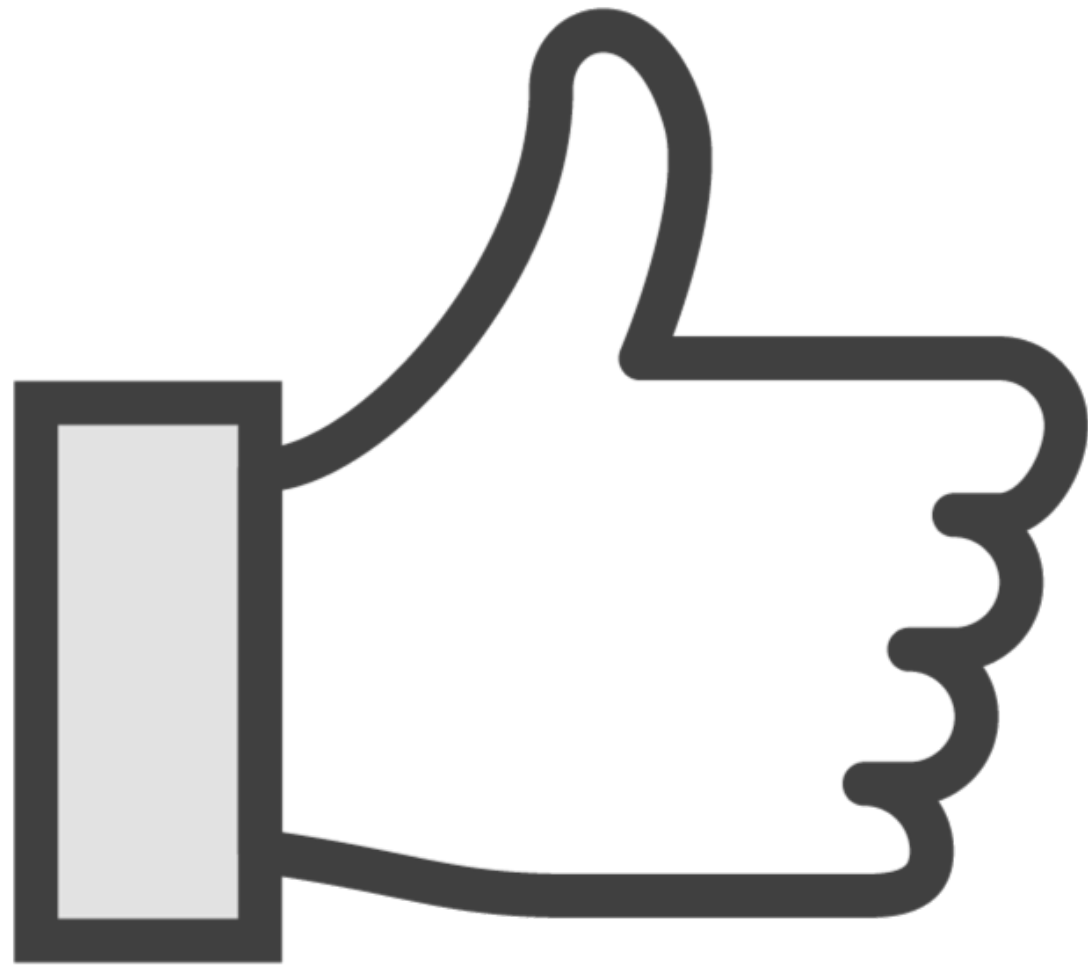
But I also need it do X!



GUI Tool:

**has 100+ features,
but not what you
need**

Code-based Solutions



“Semi-DIY”

- **REST Assured**
- **Provides a Fluent DSL**

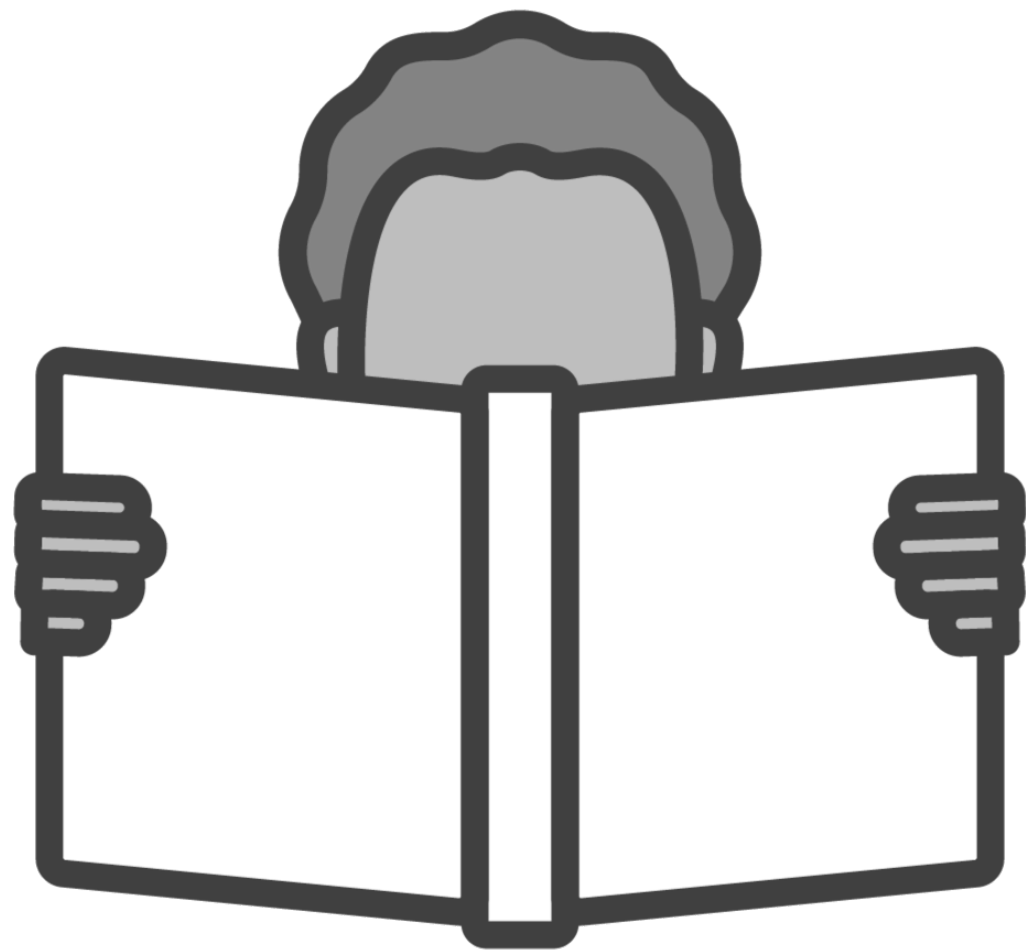
“Full DIY”

- **Plain HttpClient:**
 - **Apache Commons HttpClient**
 - **Java 11 HttpClient**


```
RestAssured.get("api/url...")  
    .then()  
    .assertThat()  
        .statusCode(200)  
    .and()  
        .contentType(ContentType.JSON);
```

```
CloseableHttpClient client = HttpClientBuilder.create().build();  
CloseableHttpResponse reponse = null;  
  
    try {  
        reponse = client.execute(new HttpGet("api/url..."));  
    } catch (IOException e) {  
        // handle  
    }  
  
int code = reponse.getStatusLine().getStatusCode();
```

Web API Testing Courses



Courses on Pluralsight:

- **Postman Fundamentals**
- **Getting Started with Web API Test Automation in Java**

Who This Course Is For



Technical Testers



Automation Engineers



Developers

Prerequisites



Java basics

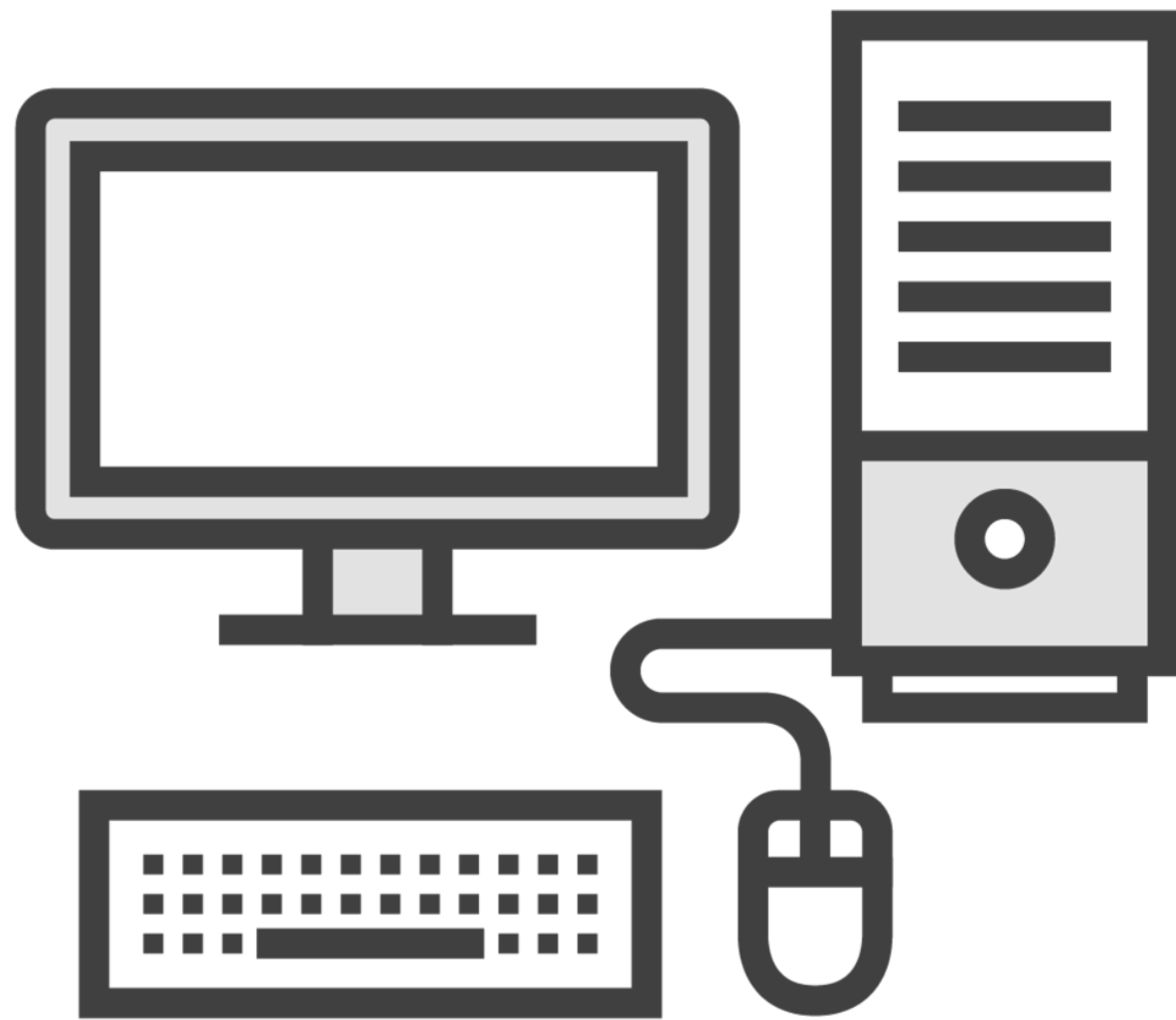
6+ months of working with Java

IDE basics, e.g. IntelliJ or Eclipse

Test Runner: JUnit or TestNG

Basics of the HTTP protocol

Technology in This Course



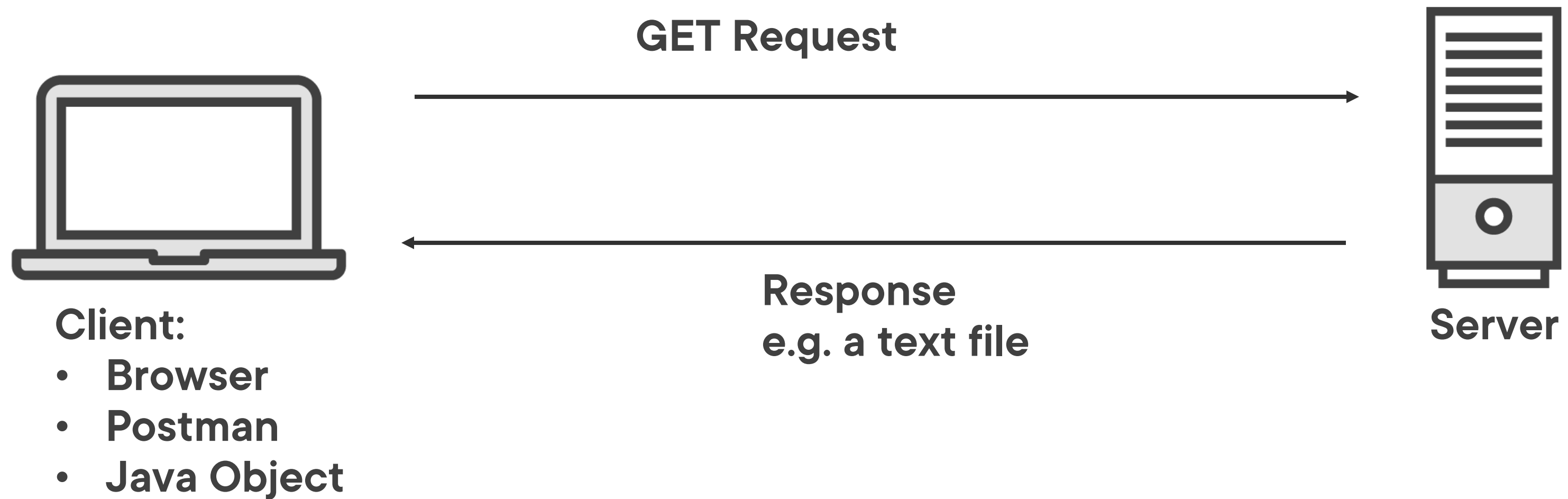
IntelliJ

TestNG

REST Assured dependency

HTTP Refresher

HTTP Refresher

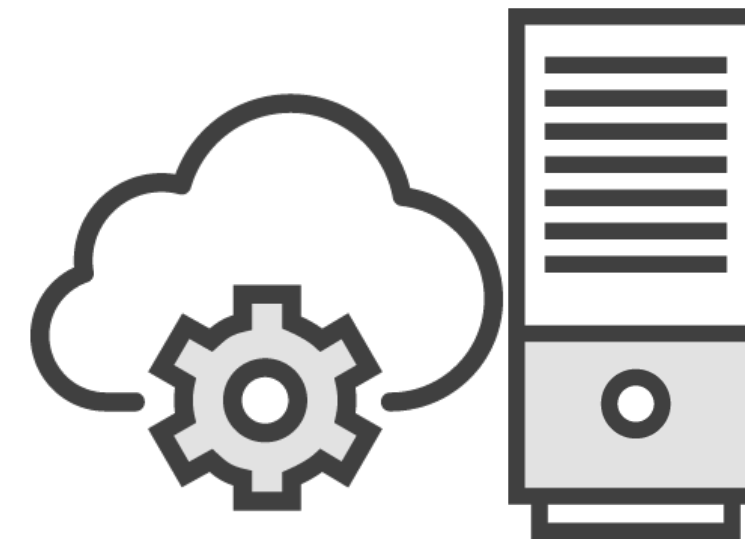


Endpoint

`https://api.github.com`

`https://api.github.com/users/{user}`

`https://api.github.com/users/{user}/repos`



Web API

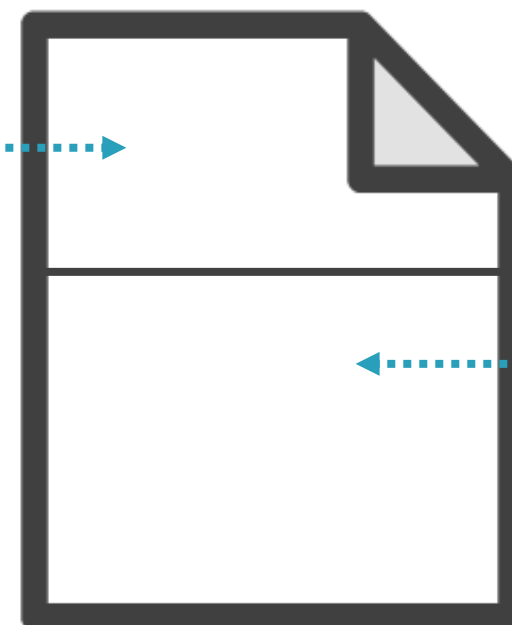
Response

Header (metadata)

e.g.

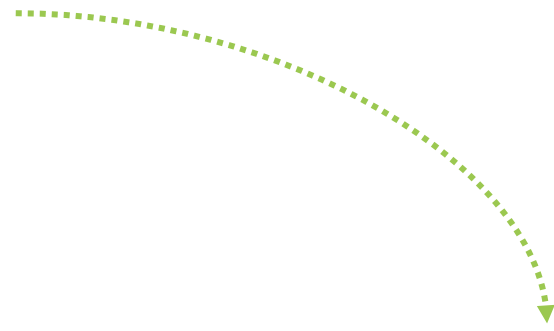
Status 200 OK

Status 404 Not Found



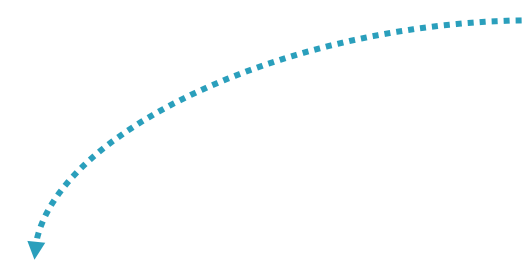
Body with data (payload)
Typically XML or JSON

XML



```
<root>
  <login>somename</login>
  <id>12345</id>
  <followers>14</followers>
</root>
```

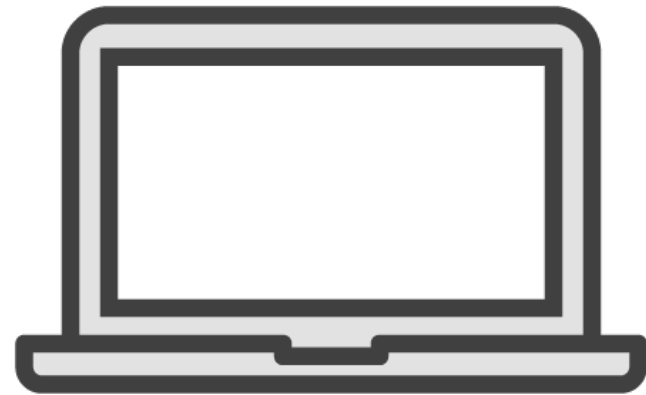
JSON



```
{
  login: "somename",
  id: 12345,
  followers: 14,
}
```

Other widespread HTTP methods:

GET
POST (to create)
PUT (to update)
DELETE



Client (e.g. web browser)



Server

Web API vs. REST API

Web API vs. REST API

Web API

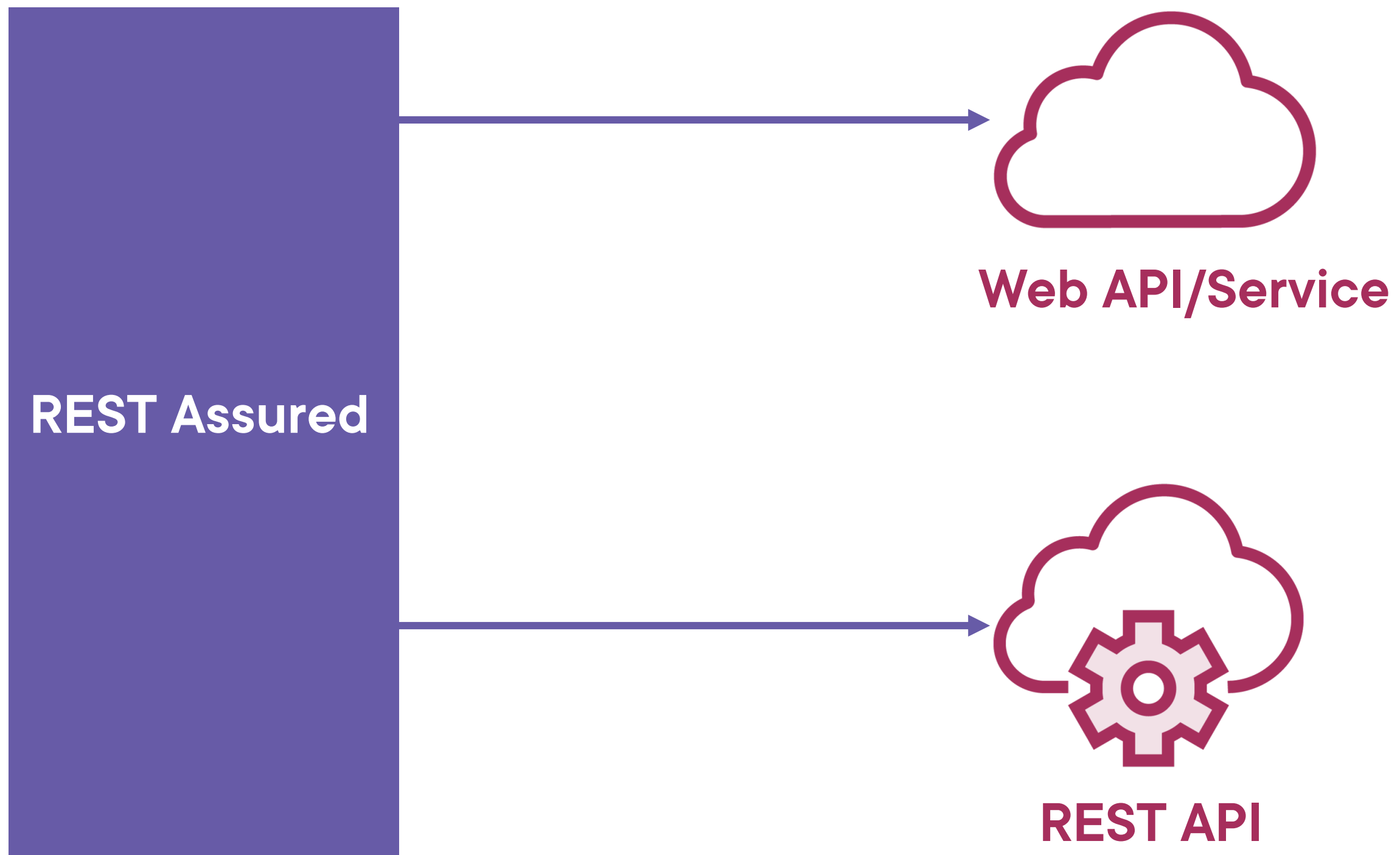
A more general term

An API that understands HTTP

REST API

A Web API...

Designed according to RESTful principles



Demo



Project setup

[Table of contents](#)

[Description](#)

[Transcript](#)

[Exercise files](#)

Demo APIs



<https://api.github.com/>

<https://jsonplaceholder.typicode.com/>

<https://reqres.in/>

Overview



- **DSL and Fluent Interface concepts**
- **GET: Validating Response headers**
- **GET: Validating Response body**
- **Unmarshalling to objects**
- **Other HTTP methods**
- **Config and Specifications**

Summary



- **Web APIs are everywhere, and we need to test them**
- **Prerequisites**
- **Review of Web APIs and HTTP**
- **Project setup**