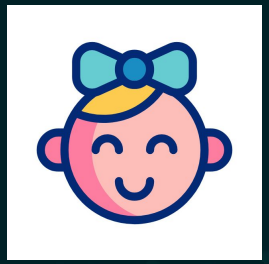
From zero to API hero with Java and REST-Assured

TechChamps



TUTORIAL

WHOA!

TechChamps

- Arjan Assink
- Enschede The Netherlands
- 10+ years in test automation
- Currently working at TechChamps

Topics

TechChamps

- APIs
- REST-Assured (incl. Key features)
- Apply DRY (Don’t repeat yourself)
- Use Helper Class
- Authorization
- Request chaining
- DTOs (Data Transfer Objects)
- Property files

You will have readable and maintainable code

Goal

TechChamps

- Start as zero
- Learn:
  - the basics with verbose code
  - to optimize code for readability and maintainability
  - to parameterize code for scalability
  - to test E2E flows
  - to use DTOs
- End as API hero

What are APIs

TechChamps

- Application Programming Interface
- APIs are like digital contracts that allow different software applications to communicate and work together.
- APIs are everywhere

What are APIs

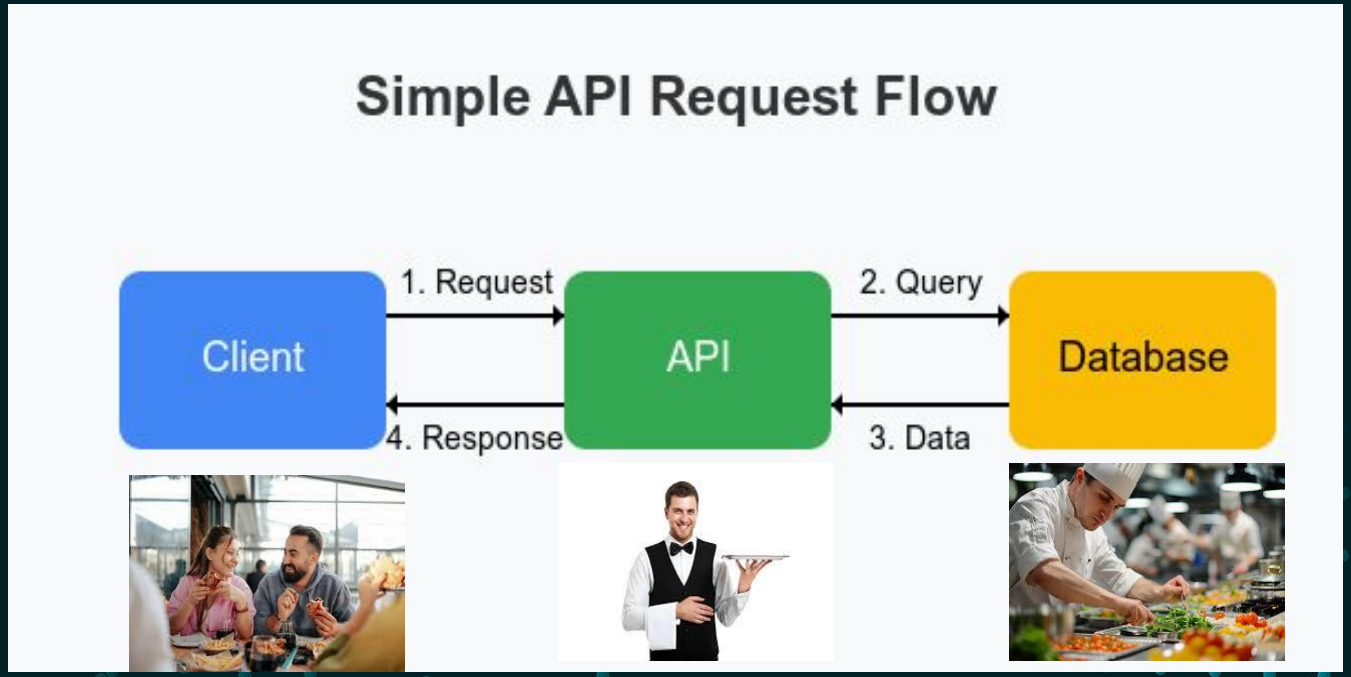
TechChamps

Different types of APIs:

- REST (Representational State Transfer)
  - Resource-based with standard HTTP methods (GET, POST, PUT, DELETE)
  - Stateless and cacheable
  - Uses standard media types (JSON, XML)
  - Most common type for web services
- SOAP (Simple Object Access Protocol)
- GraphQL (Query language for APIs)
- And more

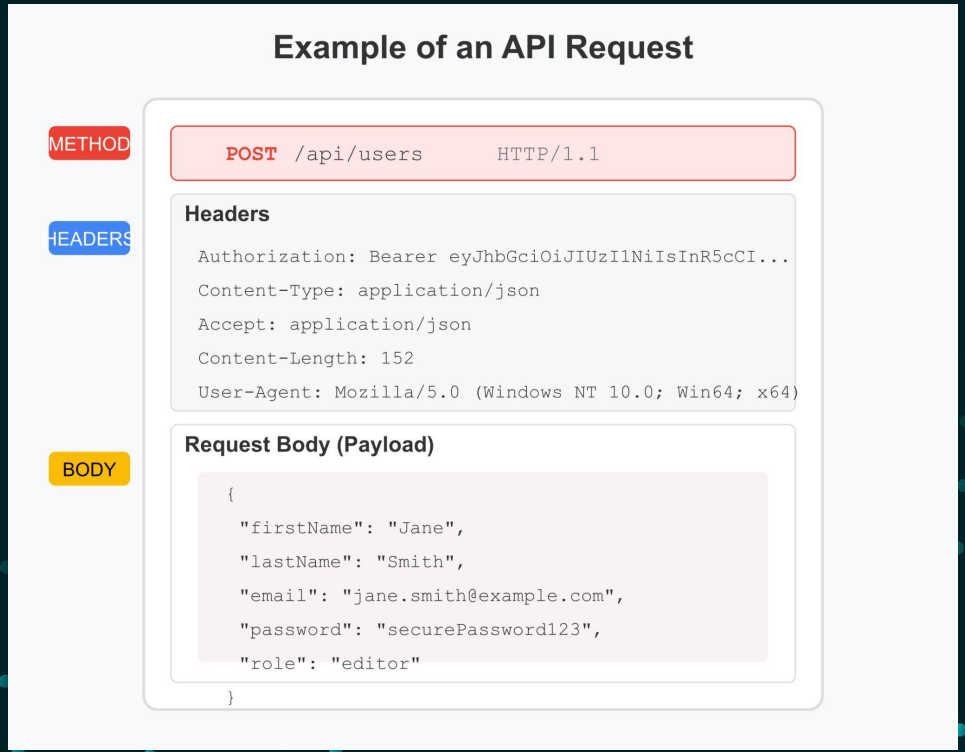
API example

TechChamps



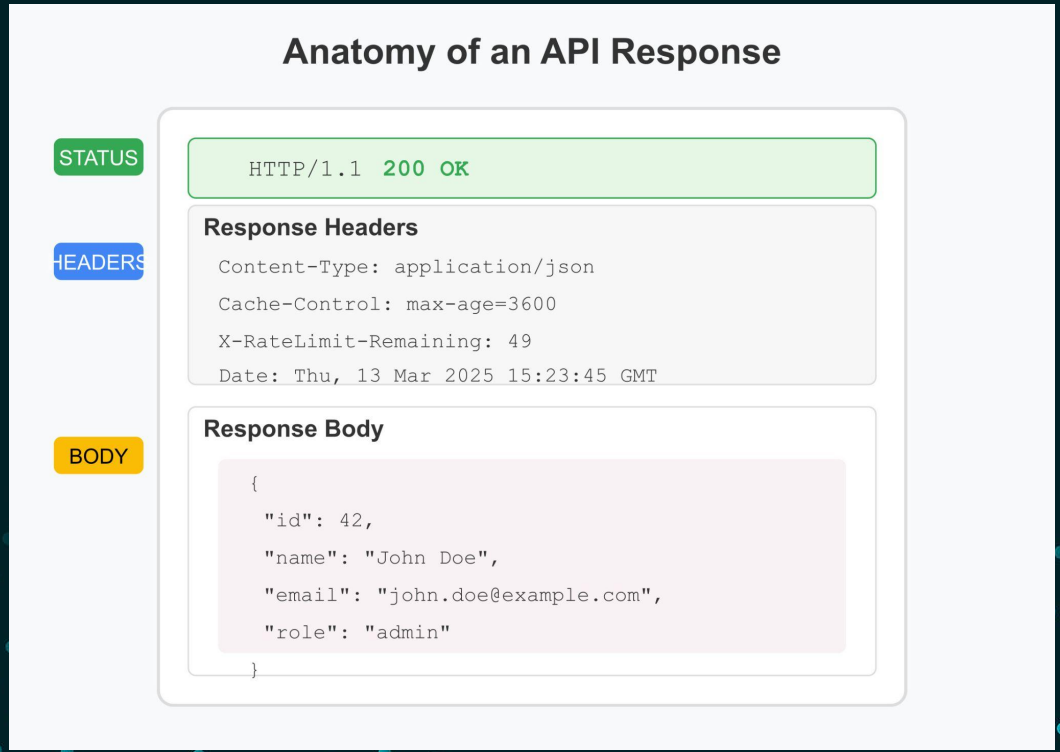
Example of a request

TechChamps



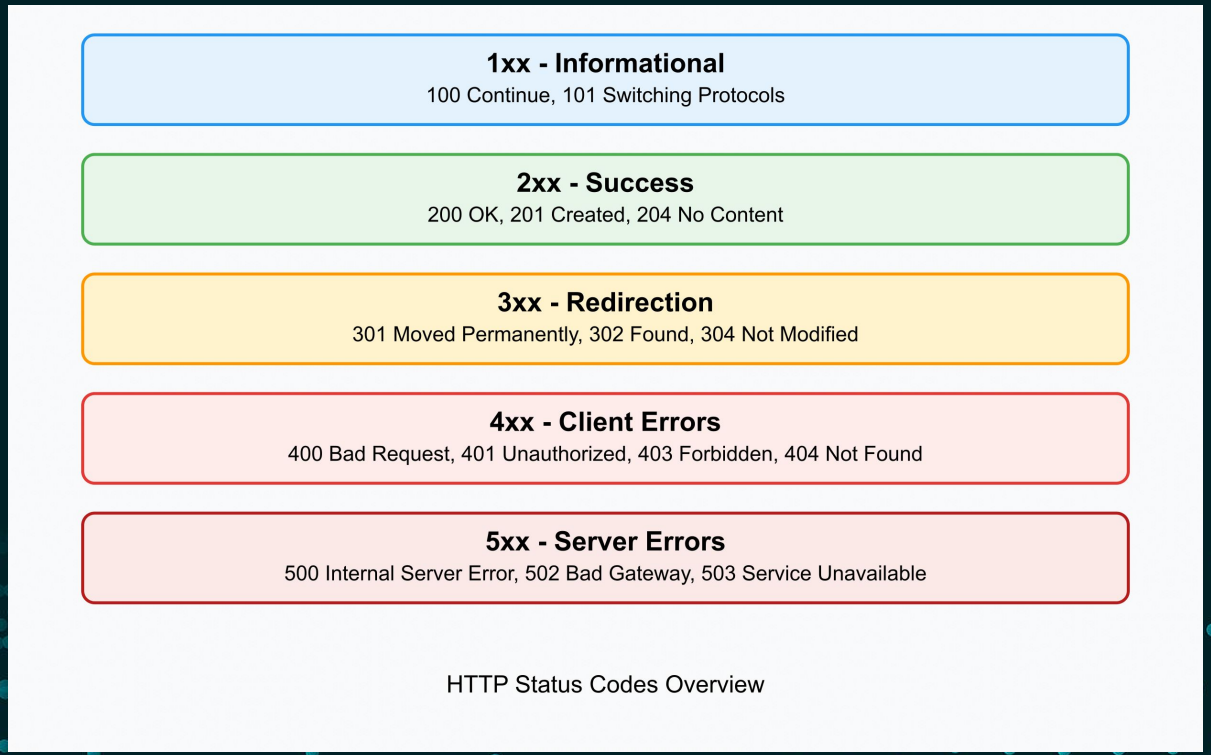
Example of a response

TechChamps



Status Codes

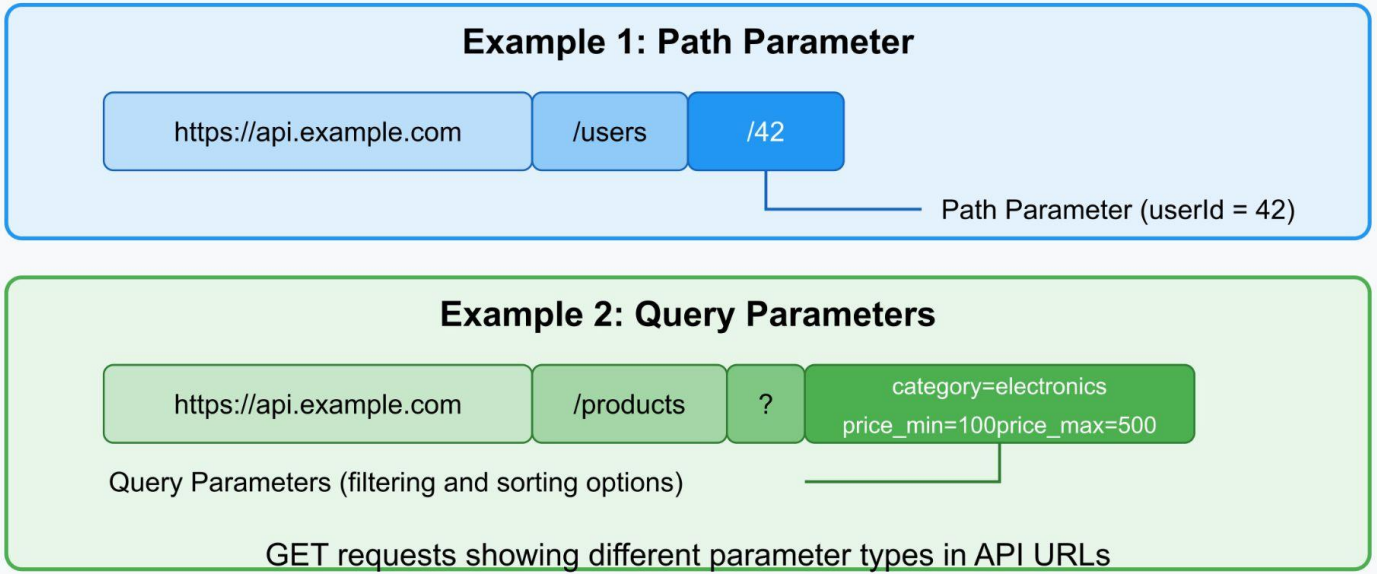
TechChamps



HTTP Status Codes Overview

Path parameters and Query parameters

TechChamps

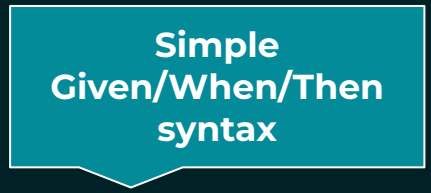


Key benefits of API testing

TechChamps

- Fast execution
- Easier to test backend logic
- More reliable
- Most business logic is handled by APIs
- Earlier detection of issues
- Easy to integrate in CI/CD pipeline
- Independent of UI framework

What is REST-Assured



REST-Assured simplifies the process of testing RESTful APIs by providing a readable and fluent syntax for writing tests.

Why REST-Assured

TechChamps

- Documentation with good examples (Usage Guide)
- Easy integration with e.g. Wiremock, Cucumber, JUnit
- Developers within the team onboard with Java
- Frequently used tool
- Open source

Key features of REST-Assured

TechChamps

- Simple Syntax for HTTP Requests
- Request and Response logging
- Support for JSON and XML
- Parameterization
- Integration with testing frameworks
- Built-in Assertions
- Support for file upload and downloads
- Custom header and request specification
- Support for authentication
- Custom (de)serialization

Basic example of REST-Assured

TechChamps

RestAssured API Test

```
Given().baseUri("https://api.example.com")

When().get("/users/1")

Then().statusCode(200)

.body("name", equalTo("John Doe"))
```

RestAssured test for validating an API response

Request specification example

TechChamps

RequestSpecification

- baseURI / basePath / port
- headers / contentType / cookies
- parameters / authentication

given()

- spec(requestSpecification)
- additional headers
- request body

when()

- HTTP method (get, post)
- Endpoint path
- Executes the request

then()

- statusCode verification
- body assertions
- header validations

API used for this tutorial

TechChamps

- Spring Boot
- OpenApi (Swagger)
- Maven for dependency management
- Authentication with JSON Web Token
- Roles (admin,user) for authorisation
- H2 in memory database

Testing:

- REST-Assured
- JUnit
- Maven for dependency management

TechChamps

Let’s look a basic sign in test

Summary

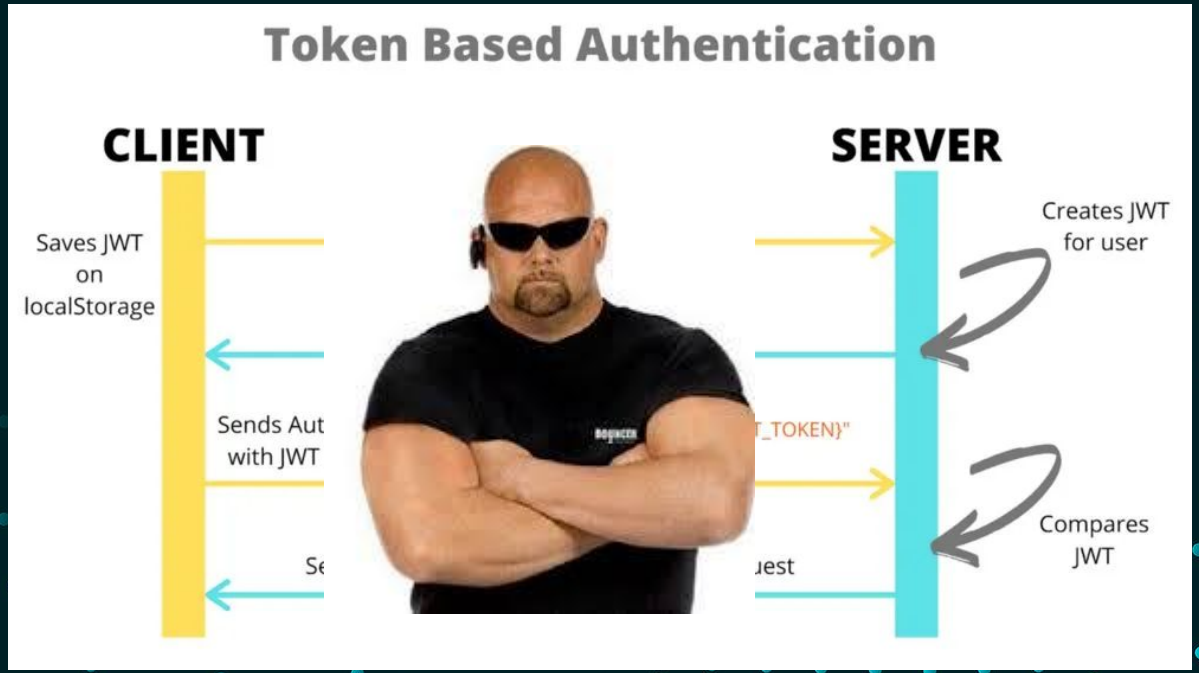
TechChamps

- Applied DRY (Don't Repeat Yourself)
- Introduced a Helper class with a RequestSpecBuilder
- This helper class will be enhanced later on

Authentication

TechChamps

How does JWT (JSON Web Token) authentication work?



TechChamps

Let’s look at an example and how to improve that

Summary

TechChamps

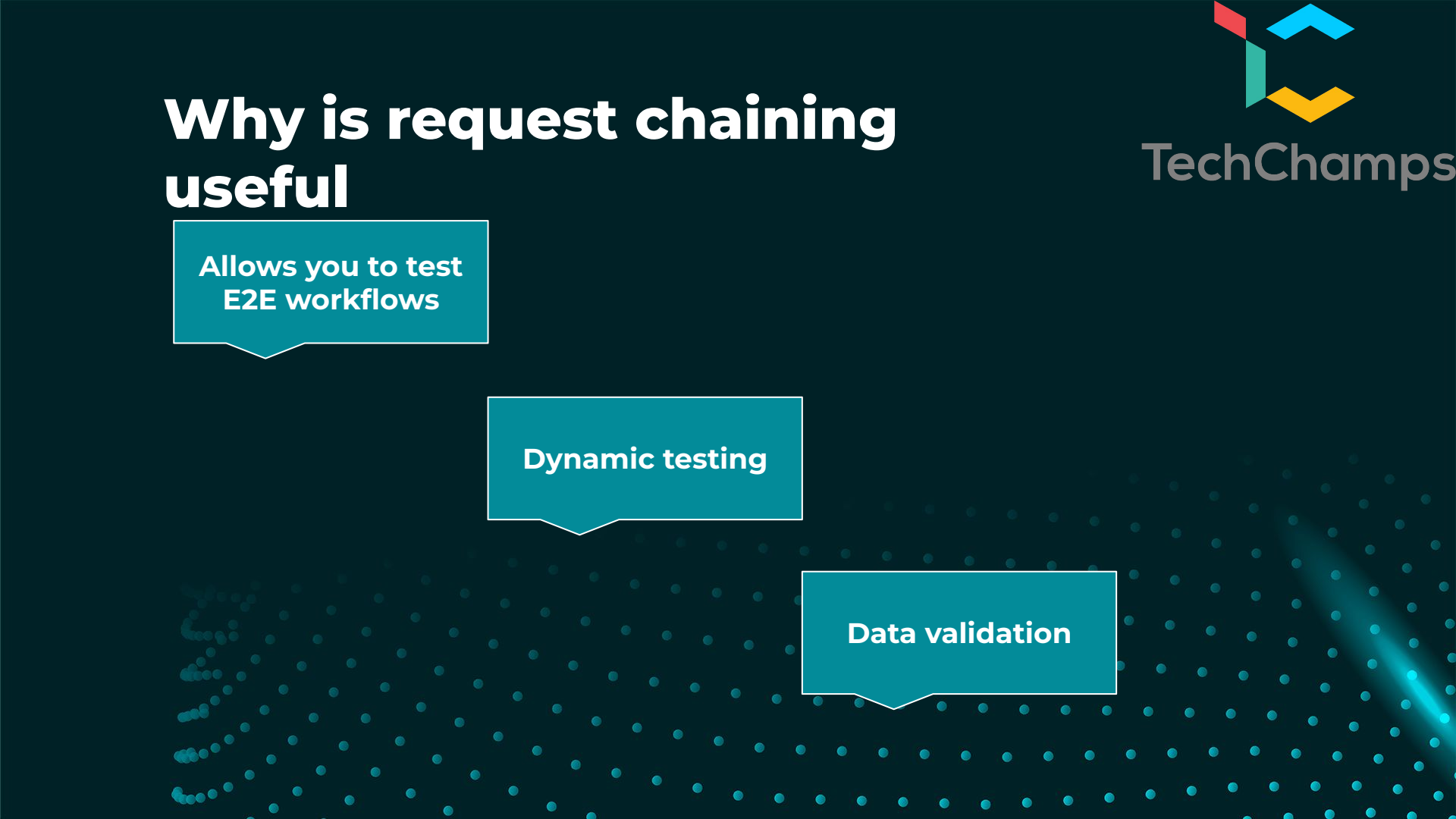
- Used authentication
- Example of request chaining
- Put the authentication part in the Helper class to make the tests more readable and maintainable

What is request chaining

TechChamps

Request chaining is a technique where multiple API requests are executed in sequence, with each subsequent request depending on data received from previous responses.

Why is request chaining useful



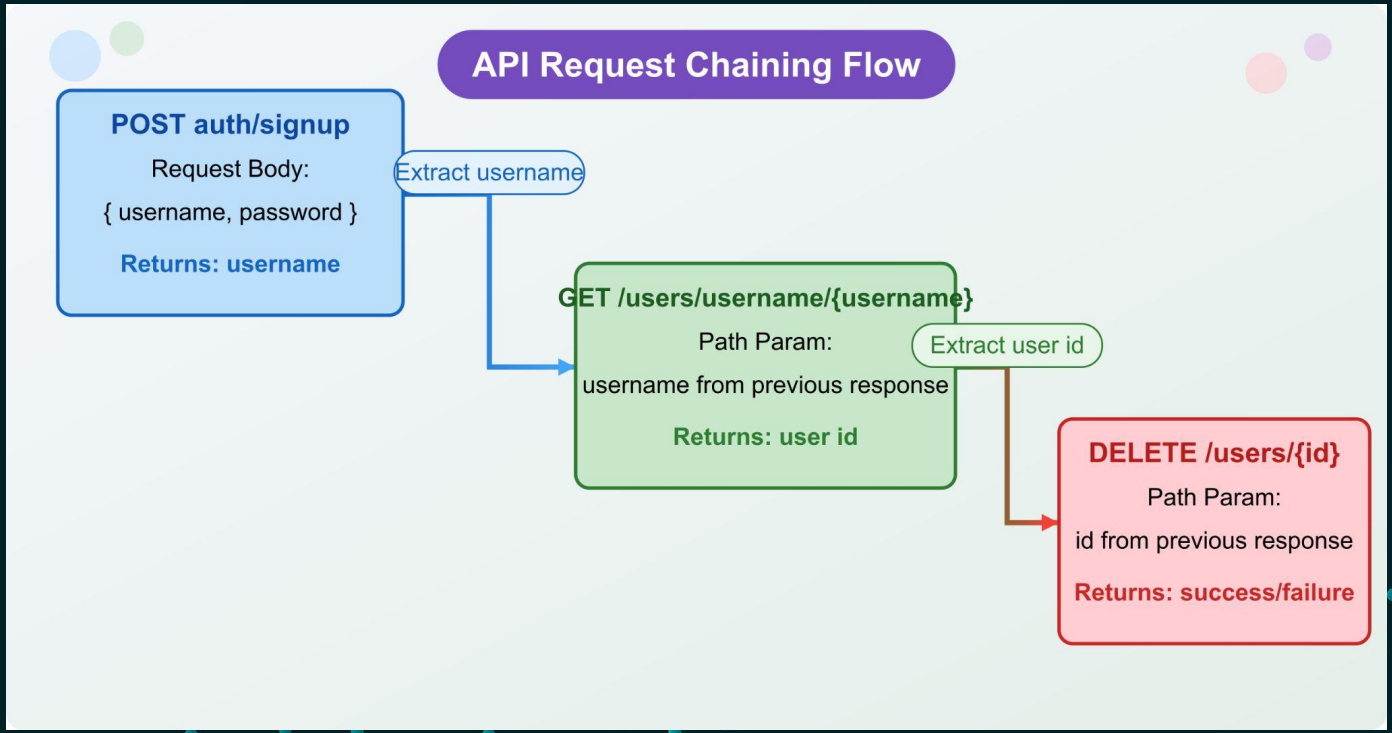
Allows you to test E2E workflows

Dynamic testing

Data validation

Let’s look at an example

TechChamps



TechChamps

Let’s implement it in our code

DTO (Data Transfer Object)

TechChamps

DTO (Data Transfer Object) is a design pattern used in APIs to transfer data between layers, systems, or modules. In the context of APIs, DTOs are objects that encapsulate data and serve as a structured way to pass information between the client and the server.

Why Use DTOs in APIs?

TechChamps

- Adaptability to client needs
- Reduce overhead
- Improved security
- Separation of concerns

Why Use DTOs in REST-Assured

TechChamps

Using DTOs (Data Transfer Objects) in REST-Assured instead of raw JSON has several advantages:

- **Type safety**
- **Increase readability**
- **Increase maintainability**
- **Easier to debug**
- **Reusability**
- **Autocompletion**

TechChamps

Let’s look at our API and use DTOs in our test

Don’t copy/create DTOs but auto generate them

TechChamps

- Code duplication
- Maintenance overhead
- Consistency

TechChamps

Let’s autogenerate our DTOs

- **Springdoc-openapi-maven-plugin**
- **Openapi-generator-maven-plugin**

Let's have a look

Why use Property files

TechChamps

- Separation of concerns
- Flexibility
- Environment - specific configurations
- Security
- Ease of Deployment
- Consistency
- Centralized Configuration Management

TechChamps

Let’s implement it in the code

Tips

TechChamps

- Integrated in CI/CD from the beginning
- Let developers review your code
- API testing is more than checking status codes and responses. Test E2E flows
- Start simple and then improve it
- Use generated DTOs
- Keep in mind DRY
- Don’t make it to abstract keep it readable for other testers

Questions?

TechChamps

- Repository
  - Main
  - Tutorial