

InWincible_shashank

Name: Shashank Shetkar

Education: B.E. (EXTC)

Profession: IT-Tech

Role: Principal QA @ Pepperfry

<https://inwincibleshashank.com/>

<https://crater.club/video/1100>

<https://www.facebook.com/testingadvices>

<https://www.udemy.com/instructor/courses>



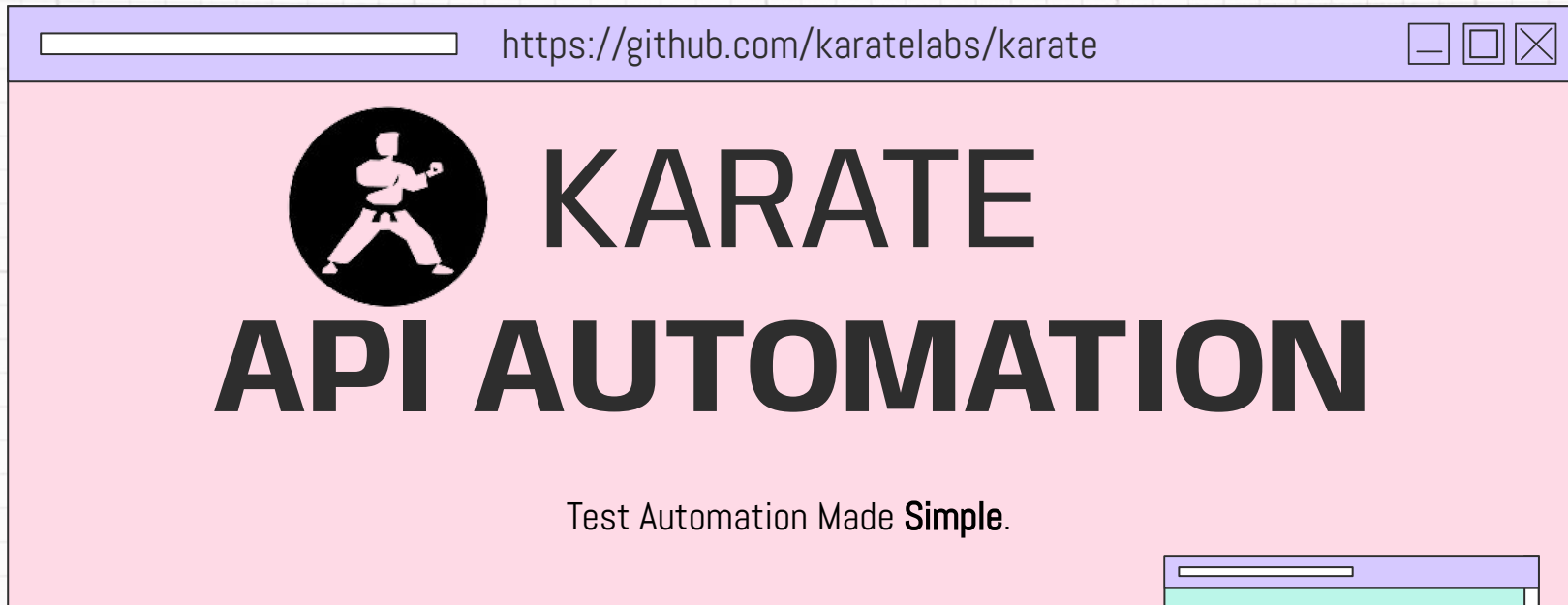
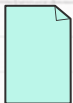
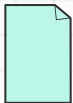
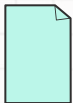
Start



Table of contents

← Back

Next →



Start



Table of contents

← Back

Next →

Table of contents

01

INTRO

What is API?
What is karate on high-level?

02

Karate Quickstart

Importing & Designing karate
framework

03

Feature

Designing feature files for API
testing

04

Validations

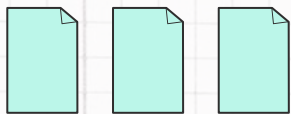
Validating response using
various functions

05

Bonus - UI

<https://github.com/karatelabs/karate/tree/master/karate-core>

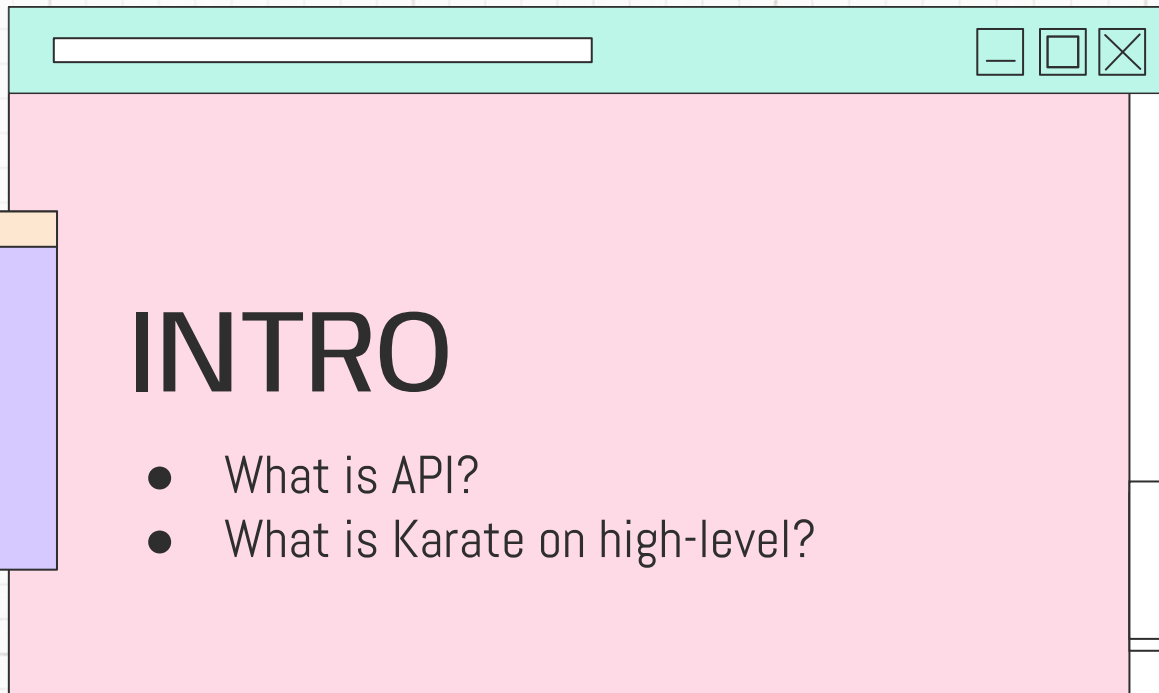




01

INTRO

- What is API?
- What is Karate on high-level?





API

What is API?

Examples

- ❏ QA engineers early insight into defects and errors so that developers can resolve the issues before they impact the GUI.
- ❏ Data is exchanged via XML and JSON formats, all languages can be used during API testing



Start

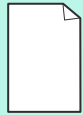
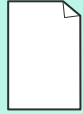
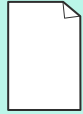


Table of contents

← Back

Next →

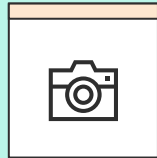
KARATE is open-source tool to combine Into single, unified framework



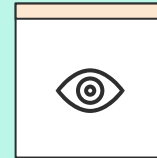
**API
test-automation**



Performance-
testing



mocks



UI automation



Features

- Java knowledge is not required and even non-programmers can write tests
- Test can be written in plain-text, readable syntax, no need to glue with step def.
- IDE Support - V.S.C., IntelliJ, Eclipse
<https://marketplace.visualstudio.com/items?itemName=karatelabs.karate>
- Write expected results as readable, well-formed JSON. For validating response we can use match operation which ignores white-space, and the order of keys.
- Fuzzy Matching
- Fully featured debugger that can step backwards, re-play a step while editing it

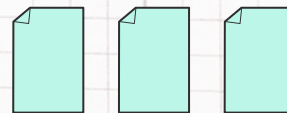
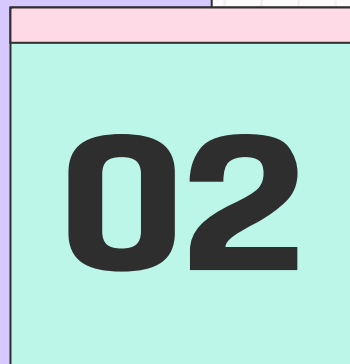
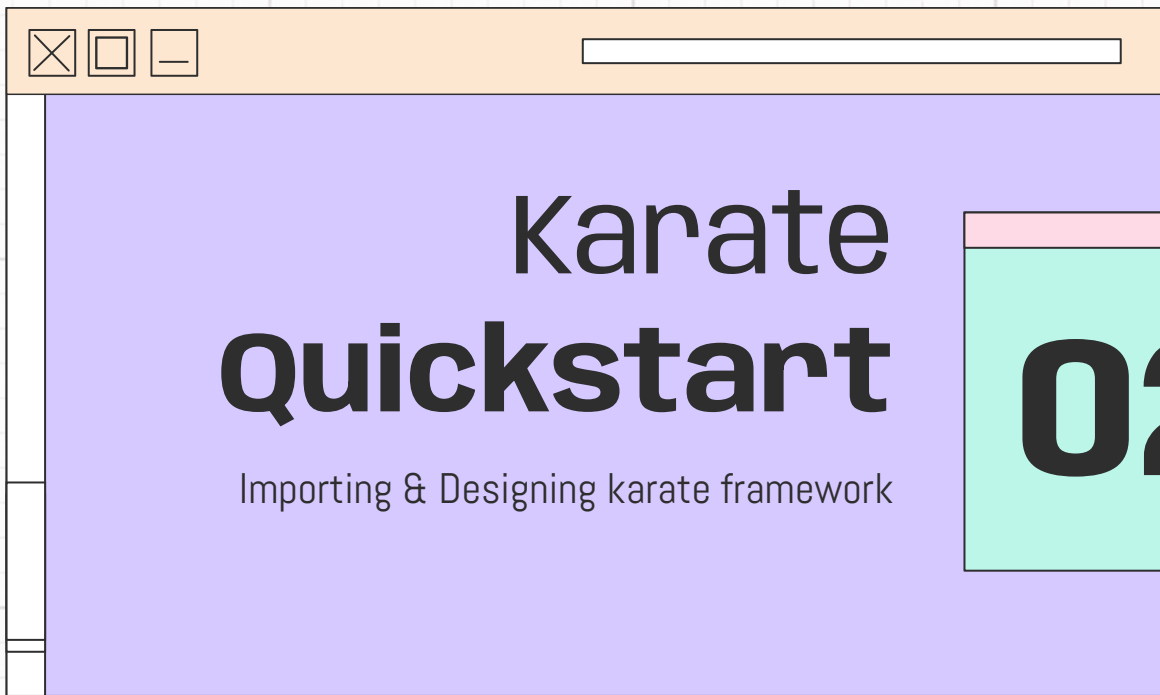
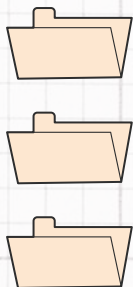
Installation

- Editor - Visual Studio Code
- Java (*to check use cmd >> java -version*)
- Maven (*to check use cmd >> mvn -version*)

Plugin

- Cucumber (Gherkin) Full Support - Alexander Krechik
- Karate Runner - Kirk Slota
- Extension Pack for Java - Microsoft
- Open In Default Browser - peakchen90





Step-1:



Maven quickstart Link

mvn clean

mvn clean install

Maven test



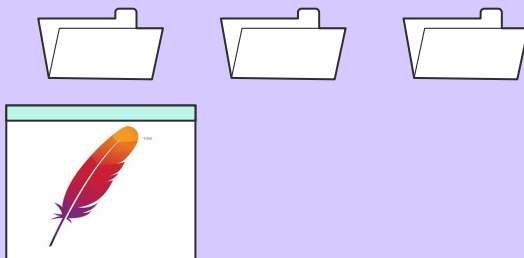
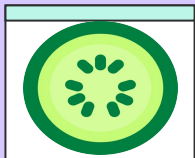
Replace below in pom.xml:

```
<groupId>com.storeapiapp</groupId>  
<artifactId>StoreApiApp</artifactId>
```

```
<properties>  
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
  <java.version>1.8</java.version>  
  <maven.compiler.version>3.8.1</maven.compiler.version>  
  <maven.surefire.version>2.22.2</maven.surefire.version>  
  <karate.version>1.2.0</karate.version>  
</properties>
```



Step-2:



- Create a file with extension ".feature"
- Will use - 'Feature', 'Scenario', 'Given', 'When', 'And', 'Then'
- Runner file - Karate testAll()
mvn test
- Karate testTags()
mvn test -Dtest=StoreApiAppTest#testTags
- mvn test "-Dkarate.options=--tags @debug"



03

Feature File

Designing feature files for API testing



Start



Table of contents

← Back

Next →

Feature file keywords



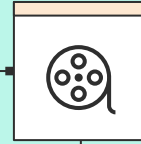
Feature

Specify feature
to test



Background

Common reusable
functions,
variables



Scenario

Test case



Examples

D.D.T.
(*Scenario Outline*)



Scenario



Keywords

Statement always start with 'Given', 'And', 'When', 'Then'

URL

To call a specific url use keyword 'url' and specify the path

method

To call a method use keyword 'method' and specify 'method name' - 'Get', 'Post', 'Put'

Status Code

To validate status code use keyword 'status' and specify 'status code' (*as number*)

header

To add header use keyword 'header'

request

To pass request body use double quotes (3x) and inside that add json request body

Print response

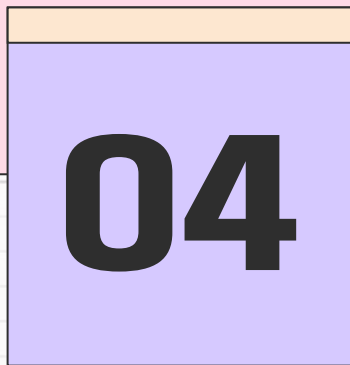
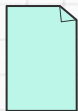
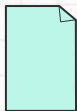
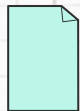
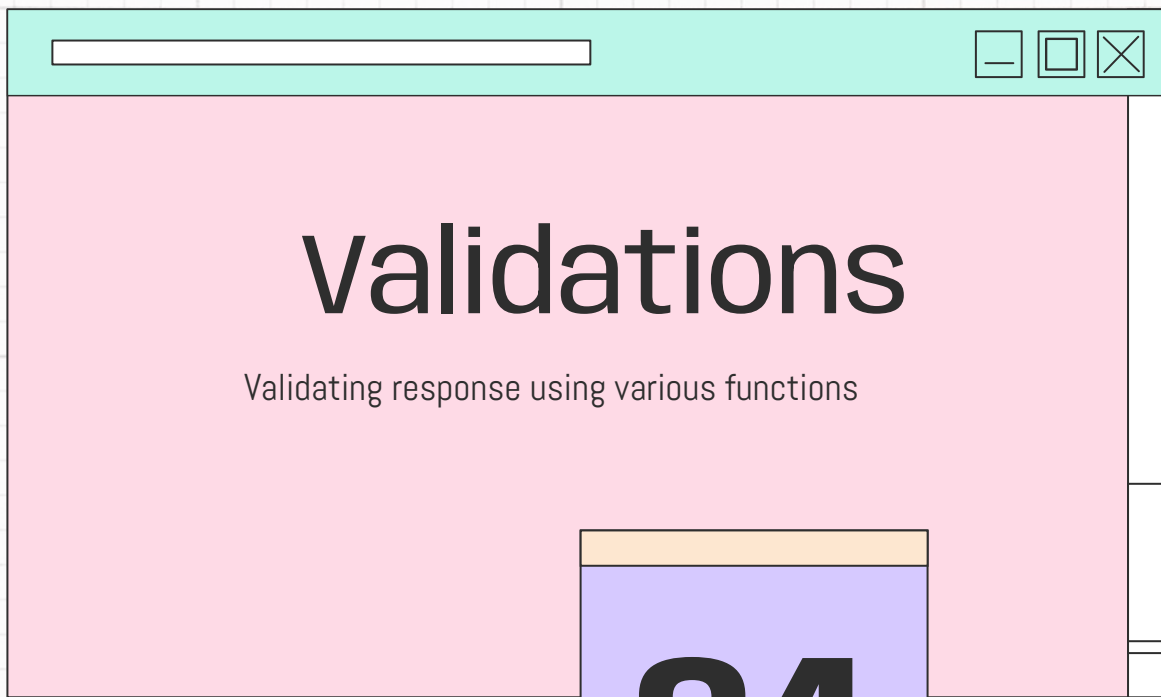
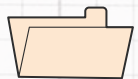
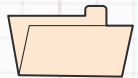
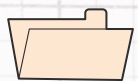
To print response use keyword 'print' and keyword 'response' for entire json response

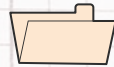
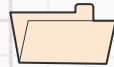
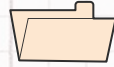
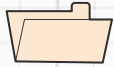
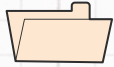
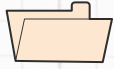
Use of Keywords inside Feature File



<u>AllProduct.feature</u>	<u>AllCategories.feature</u>	<u>Login.feature</u>	<u>User.feature</u>
Define: feature	Define: Background	Read: Another json file	Access: Token from another file
Scenario: Given, When, Then	Read: Another .js file within package	Define: variable to store file	Call & read: Another feature file
Method: GET	Define: reusable variable	Set: Token values	Define: header, headers
Validate: status Code	Call: variable inside scenario	Scenario Outline: with Examples (<i>D.D.T.</i>)	Get: Token value and re-use in header
	Match: basic response		Match: basic response





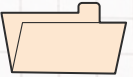
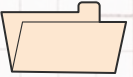
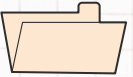
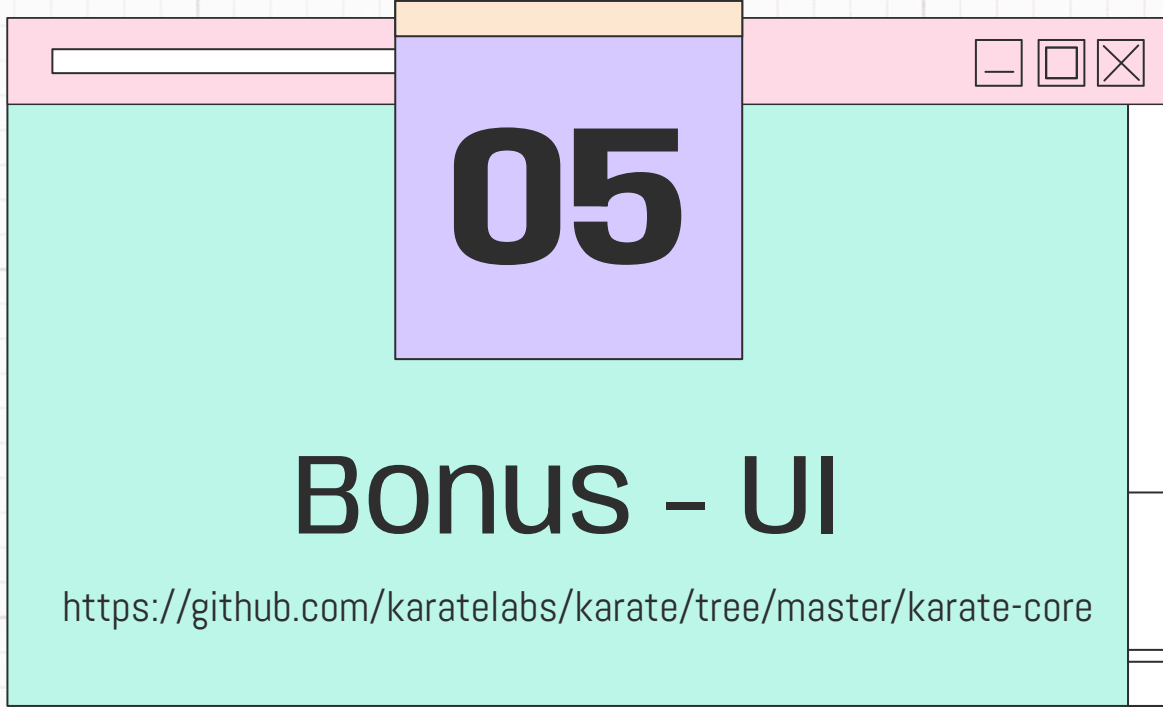
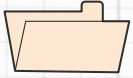
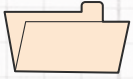
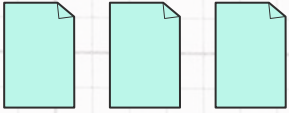


AllCategories.feature/Scenario/3

Fuzzy Match

- What? Validate variable type
- Which? Array, number, string, boolean, null/
non-mandatory
- How? '#array', '#number', '#string', '#boolean',





Start



Table of contents

← Back

Next →

Thanks!

Do you have any questions?

shetkarshashank@gmail.com

98-200-10741

<https://inwincibleshashank.com/>



Start



Table of contents

← Back

Next →