**AUTOMATED UNIT TESTING**

**BROWSER TEST AUTOMATION**

**MOBILE TEST AUTOMATION**

**USING**

**PYTHON / PYTEST/ SELENIUM/ APPIUM AND**

**OTHER EXTENSIONS**

Table of Contents

[1 Introduction 3](#_Toc26082876)

[2 Glossary 3](#_Toc26082877)

[3 Audience 3](#_Toc26082878)

[4 Making sense of the overall ecosystem 3](#_Toc26082879)

[5 Different approaches to solve the same problem 3](#_Toc26082880)

[6 Ecosystem for the training course 3](#_Toc26082881)

# Introduction

# Glossary

# Audience

# Making sense of the overall ecosystem

# Different approaches to solve the same problem

# References

# Ecosystem for the training course

## Mocking

provides extremely powerful and useful means by which to mock and stub out these undesired side-effects.

example: perhaps we’re writing a finance app and want to test out our new ‘Post to Accounts feature’, but don’t want to actually post to Accounts every time we run our test suite

## Pytest

### Good Practices

### Fixtures

|  |  |  |  |
| --- | --- | --- | --- |
|  | A software test fixture sets up the system for the testing process by providing it with all the necessary code to initialize it, thereby satisfying whatever preconditions there may be. An example could be loading up a database with known parameters from a customer site before running your test. |  |  |
|  | The purpose of test fixtures is to provide a fixed baseline upon which tests can reliably and repeatedly execute. |  |  |

## Selenium

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

## Web drivers

### Chrome

### Firefox

### Edge

### Safari