

# Getting Started with Appium and SpecFlow

---



**Marko Vajs**

Software Development Engineer in Test



# Module Overview



**Get to know Appium and SpecFlow a bit more**

**Introduce demo applications**

**Set up a development environment:**

- Run both applications locally
- Create a new project
- Do the initial setup

**Formulate and automate a scenario using SpecFlow and Appium**



# Introducing Appium and SpecFlow

---





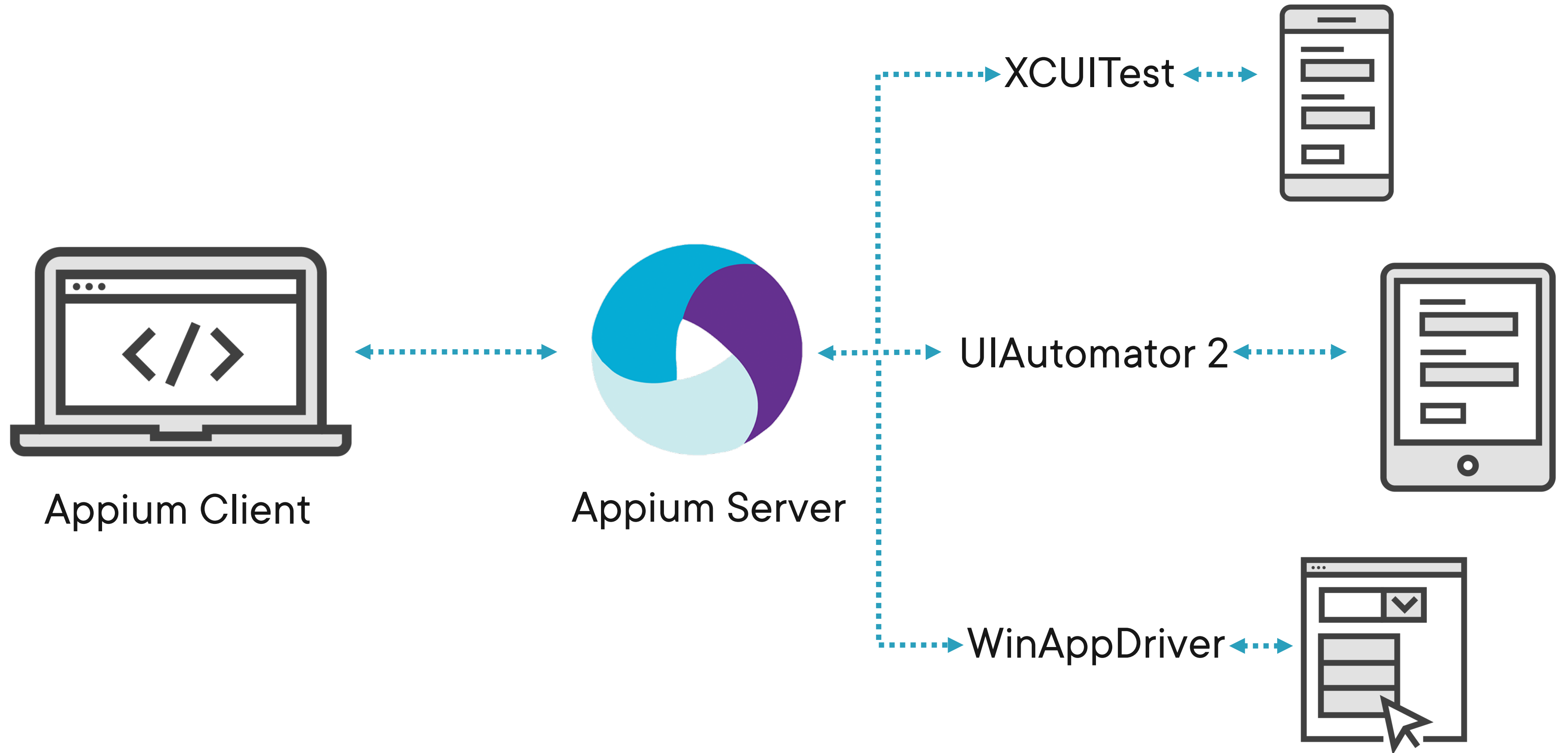
**Open-source test automation framework for automating native, hybrid, and mobile web apps**

**Follows client/server architecture**

**Uses vendor-provided automation frameworks under the hood**

**Supports client libraries for multiple programming languages**







Used for defining, managing, and executing scenarios in .NET

Kicks-in in the Formulation phase

Supports different test execution frameworks such as NUnit and xUnit

Living documentation generator



# Introducing Demo Application

---



# Globotickets

iOS





10:29

## Globotickets

GUEST INFORMATION

Number of Guests

VENUE INFORMATION

Venue City Hall >

Service Level Basic >

Promotion Included ☐

PAYMENT INFORMATION

Upfront Payment ☐

% Upfront Payment

RESULT

Calculate

Clear

### Three venues:

- City Hall (200 guests, \$2,500)
- Main Building (400 guests, \$5,000)
- Retro Lounge (500 guests, \$7,000)

**The number of guests cannot be higher than the venue's capacity**

### 2 service levels:

- Basic (\$100 per guest)
- Premium (\$150 per guest)

**The promotion costs \$500**

**Discount does not apply to the promotion**



10:29

## Globotickets

GUEST INFORMATION

Number of Guests

VENUE INFORMATION

Venue City Hall >

Service Level Basic >

Promotion Included ☐

PAYMENT INFORMATION

Upfront Payment ☐

% Upfront Payment

RESULT

Calculate

Clear

### Upfront payment discount:

- 10-49% upfront payment is 10%
- 50-100% upfront payment is 20%

**Calculate button triggers cost calculation**  
**If there are validation errors, they will be displayed in the results field instead of the total cost**

**Clear button resets the form to its initial state**



Total cost is calculated using the formula:

$$(\text{venue price} + \text{number of guests} \times \text{service level price}) \\ \times \text{discount \%} + \text{promotion fee}$$

Clicking on the calculate button



```
graph LR; A[Clicking on the calculate button] --> B[If the form is valid, then the total cost is displayed.]; A --> C[If the form is NOT valid, validation message is displayed.];
```

If the form is valid, then the total cost is displayed.

If the form is NOT valid, validation message is displayed.



**Rule that says that the discount does not apply to the promotion fee**

**Rule that describes what happens when the user clicks on the calculate button**





If you cover all acceptance criteria with examples, what would you put under the rule that describes the calculate action?



You cannot always map  
one-to-one rules with examples.

As long as you are able to  
generate examples considering  
all available rules, discovery is  
successful.



10:29

## Globotickets

GUEST INFORMATION

Number of Guests

VENUE INFORMATION

Venue City Hall >

Service Level Basic >

Promotion Included ☐

PAYMENT INFORMATION

Upfront Payment ☐

% Upfront Payment

RESULT

Calculate

Clear

Jim has opened the Globotickets application.

Jim selects the venue option.

Displayed options: City Hall, Main Building, and Retro Lounge.

Jim has opened the Globotickets application. City Hall is selected.

Jim enters 200 for the number of guests and clicks calculate.

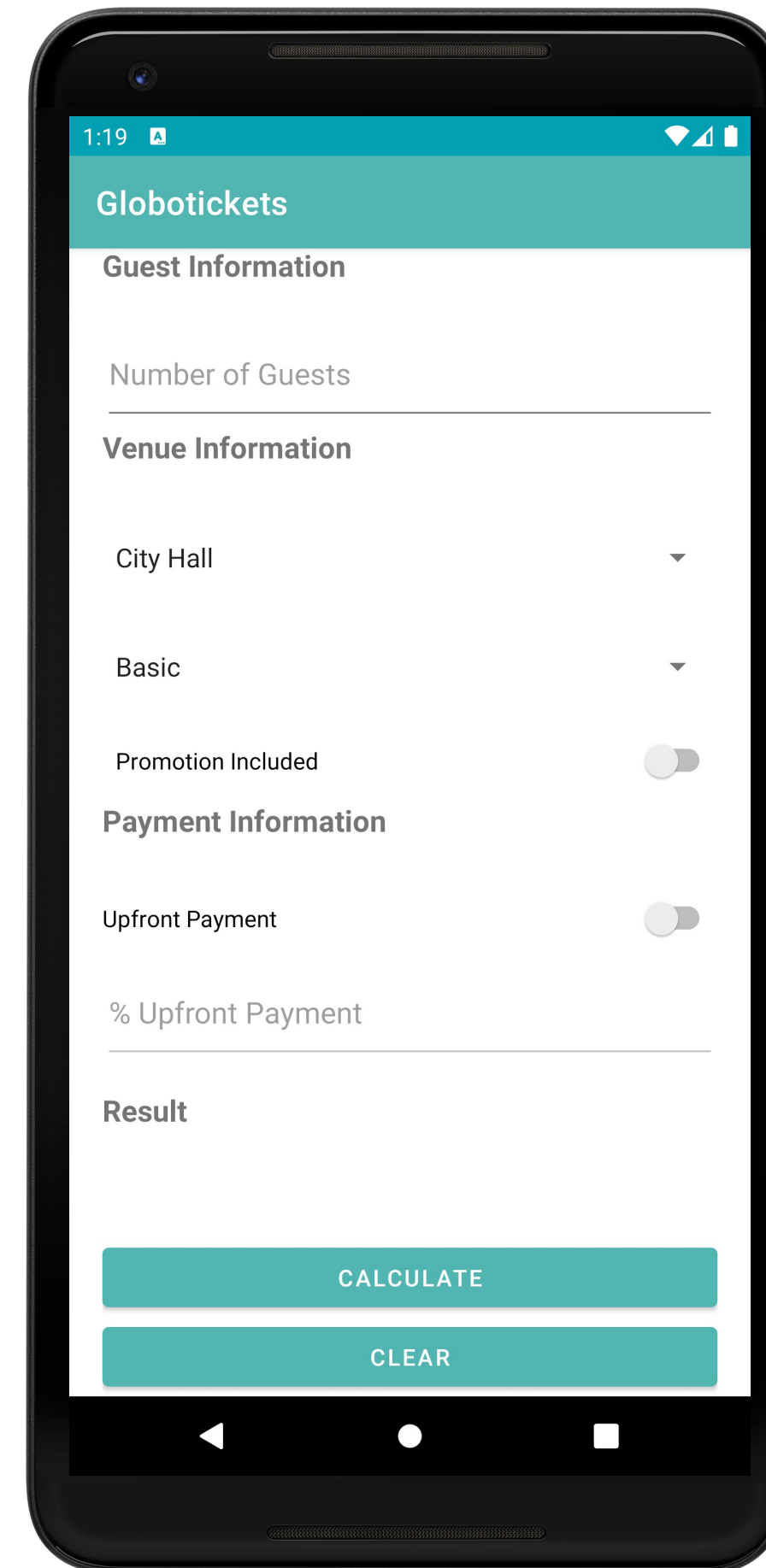
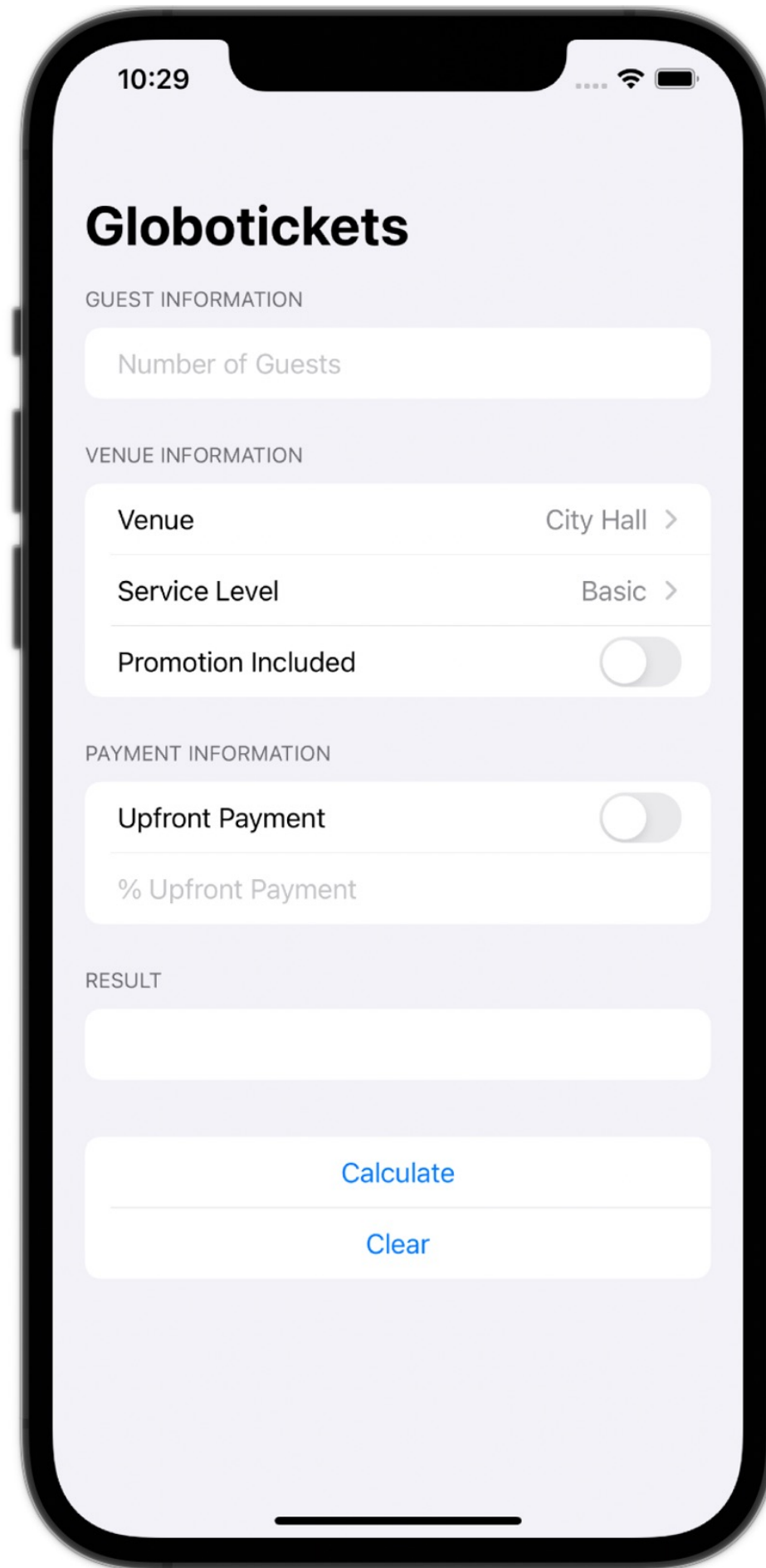
The result is: 22500 ( $2500 + 200 \times 100 = 22500$ )

Jim has opened the Globotickets application. City Hall is selected.

Jim enters 201 for the number of guests and clicks calculate.

Validation message Selected venue cannot accept the desired number of guests. is displayed in the results field.







# Running Demo Application

---



# Getting Started with UI Testing and Appium

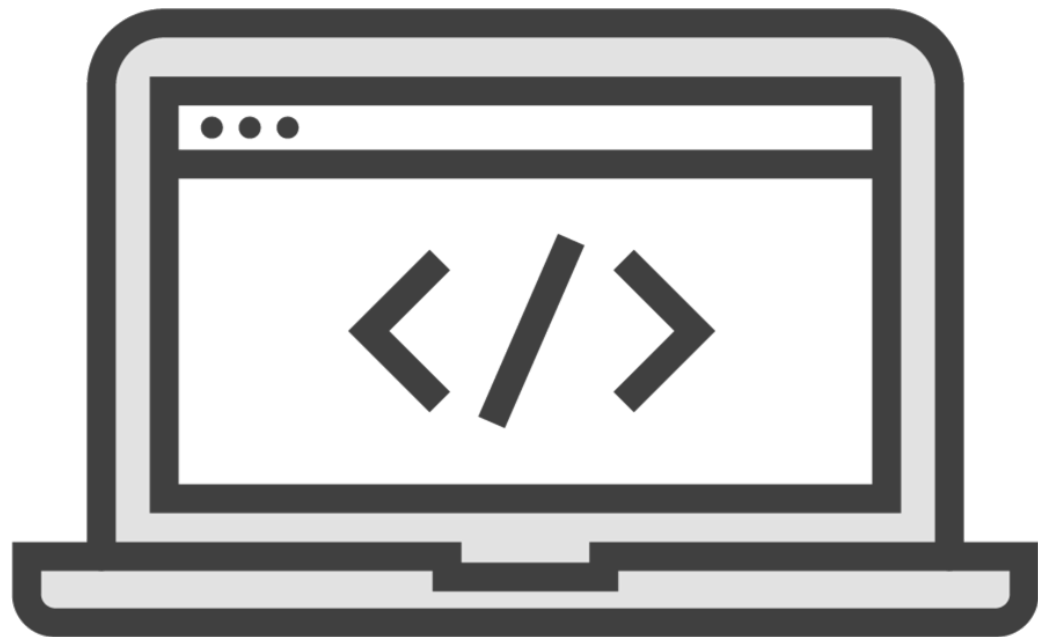
by Marcel de Vries

This course you will teach you how to write UI Test automation in the programming language you like for applications on iOS, Android, Mac, and Windows including UWP, WPF, and Windows Forms applications using the open source tool Appium

 **Resume Course**



# Android Requirements



**Mac or Windows  
computer**

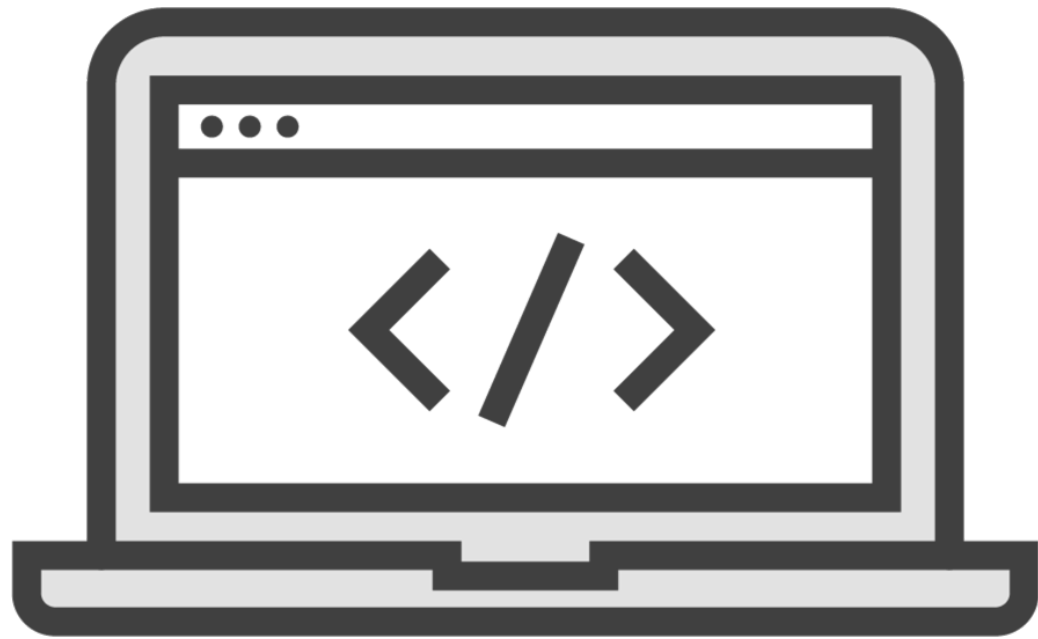


**android**

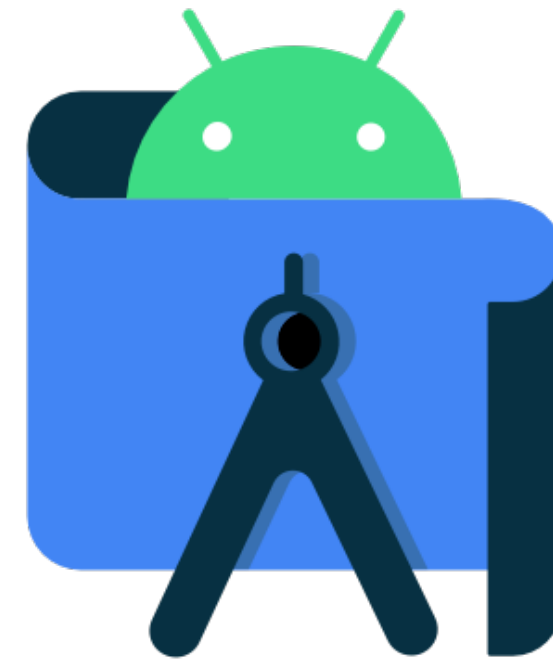
**Android SDK**



# Android Requirements



**Mac or Windows  
computer**



**Android Studio**

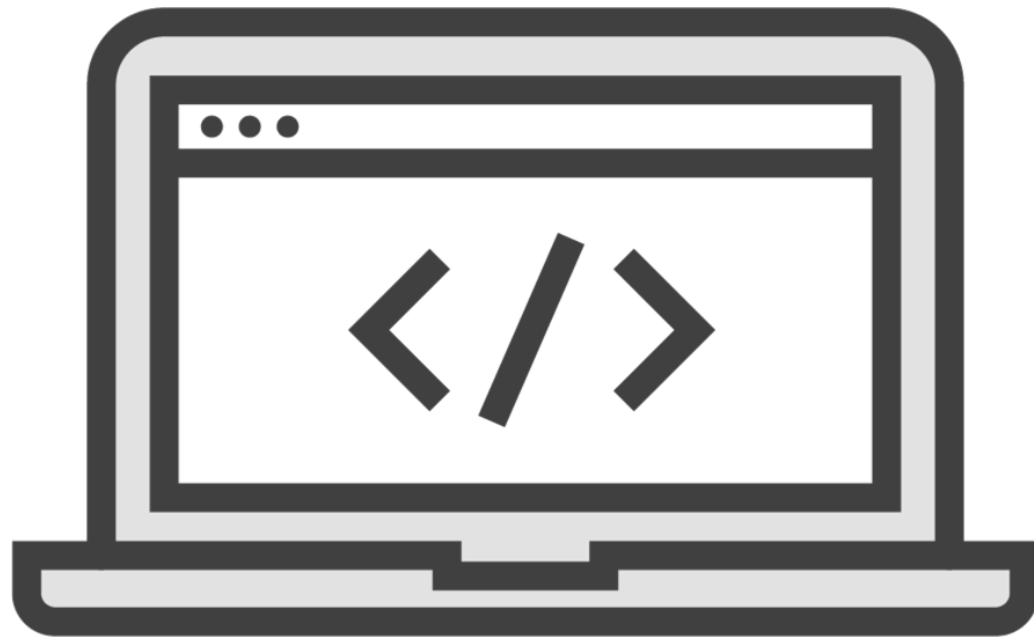
# Demo



## Running Android application locally



# iOS Requirements



**Mac computer**



**Xcode**

# Demo



## Running iOS application locally





# Installing Appium Desktop

---



# Appium Desktop



**Appium server**



**Appium inspector**

# Demo



## Installing Appium Desktop



# Demo



## Creating Appium with SpecFlow project



# Demo



## Adding configuration to the project

- Create `appsettings.json` and add properties to it
- Create a model for the settings file
- Implement deserialization



# Demo



## Handling Appium driver instance



# Introducing Feature Files and Formulating a Scenario Using Gherkin

---



# What Are Feature Files?



**Serve as a place to store features**

**SpecFlow knows how to interpret them**

**Need to be well structured since they serve as a documentation**

**Contain one or more scenarios that are connected to code from C# step classes**





# Demo



## Formulating a scenario



10:29

## Globotickets

GUEST INFORMATION

Number of Guests

VENUE INFORMATION

Venue City Hall >

Service Level Basic >

Promotion Included ☐

PAYMENT INFORMATION

Upfront Payment ☐

% Upfront Payment

RESULT

Calculate

Clear

Jim has opened the Globotickets application.

Jim selects the venue option.

Displayed options: City Hall, Main Building, and Retro Lounge.

Jim has opened the Globotickets application. City Hall is selected.

Jim enters 200 for the number of guests and clicks calculate.

The result is: 22500 ( $2500 + 200 \times 100 = 22500$ )

Jim has opened the Globotickets application. City Hall is selected.

Jim enters 201 for the number of guests and clicks calculate.

Validation message Selected venue cannot accept the desired number of guests. is displayed in the results field.



# Demo



## Automating a scenario on iOS



# Demo



## Automating a scenario on Android



# Module Summary

---



# Module Summary



## Technical requirements:

- IDE, preferably Visual Studio or Rider
- SpecFlow plugin
- Appium Desktop
- Android SDK and/or Xcode

**Try keeping feature files as structured and organized as possible because their purpose is to serve as documentation**

**Using the same code to automate multiple platforms requires additional efforts to keep that code readable, maintainable, and reusable**



Up Next:  
Exploring Capabilities of SpecFlow through  
Test Automation with Appium

---

