

# QA AUTOMATION ENGINEER INTERVIEW GUIDE



# Welcome to theScore

This guide is meant to help prepare you for the technical challenge (take-home) + technical interview. Feel free to reach out to your recruiter if you have any additional questions.

Please send us an estimate of how long it will take you to complete the challenge as soon as you can.

## Solution Stack

You have the option to use a stack that you're most comfortable with, however our preference would be for you to use the Appium Framework. You can choose the programming language of your choice, but if you need a suggestion go with Kotlin.



## Purpose

The purpose of this challenge is for us to get a better understanding of your abilities to automate a test on a native mobile application. After you submit your code we'll run it on our end and perform a code review. We will evaluate you on your general coding style and testing practices.

We know coding can be stressful in an interview setting, so we want to let you know exactly what we are looking for.

- Scope - Does your code implement all required features?
- Correctness - Do the tests execute correctly & pass?
- Design - Is your code easy to follow and well organized?

# Requirements

Write an automated test that finds a league, team, or player within **theScore app** and verify the specified steps. Assume the project you create will be the foundation for an automation suite, so please build appropriate abstractions as necessary.

Automate the following steps:

1. Open a league, team, or player page of your choice (bonus points for using a data-driven or parameterized approach).
2. Verify that the expected page opens correctly.
3. Tap on a sub-tab of your choice, eg: league table / standings / leaders, or stats tab of the league, team, or player.
4. Verify that you are on the correct tab and that the data is displayed correctly and corresponds to the league, team, or player from step 1.
5. Verify that back navigation returns you to the previous page correctly.

## Prerequisites

Please include a gradle or maven build file using repository dependencies (no jar files) and in your docs describe which tasks should be run for initialization. The test host will have Java 11 and Appium 1.22.3. If your requirements vary, please specify the required versions in your documentation.

Feel free to use a free trial to a cloud testing platform if you don't want to run your tests on a local emulator or device, but you can also assume that we will have an emulator or physical device handy to run your tests. Make sure you document the required device setup steps for us to perform.

To get the apk file, you may use the Google Play link provided above or another service such as [apkpure.com](https://apkpure.com) or [apkmirror.com](https://apkmirror.com). Please include the apk file in your submission so we can test on our end using the same build.

## Document your submission

Ensure the repository has a well formatted README.md file with documentation for setting up and running your submission from the command line on a macOS or unix/linux environment.

If your submission doesn't already have a test specification separate from the implementation, please include a human readable test specification or test run report for the scenario you implemented and provide a link to it in your README.

**Bonus:** Write a short description of the rationale you used to decide your test approach and provide a coverage assessment of the feature your tests targeted.

## Completion



Create a new public **GitHub** repository to push your submission to; aim to use proper git workflow with meaningful commit messages. Once it is ready, please send a link to the project repo in an email to **qachallenge@thescore.com**. If you don't want to publicize completing the challenge, email a patch instead.

## Checklist

- ☐ Automated test
- ☐ Prerequisites
- ☐ Documentation

If you have any questions regarding requirements, please do not hesitate to email **qachallenge@thescore.com** for clarification.

Good luck!