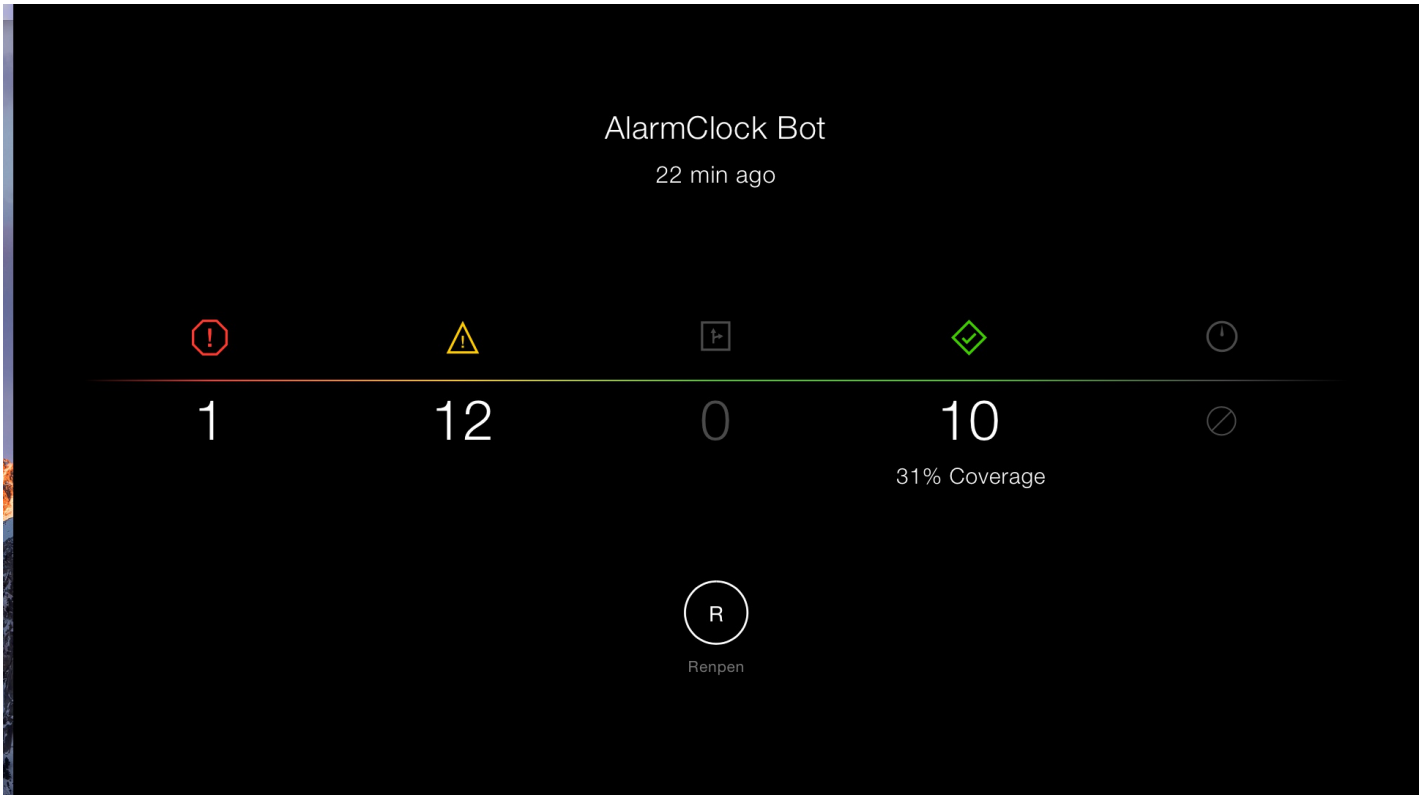
Continous Integration and Testing

Hi guys

we started with Continuous Integration of our App. As always in the Apple World we can´t use something like Jenkins for this task. Apple offers a solution called xCode Server that is a feature of the Mac OS Server App. This App can run on every Mac and turns it into a Server with additional features.

The xCode Server for instance provides the possibility to create something called xCode Bot. These bots can created on every Development Mac and can then be deployed at the Mac OS Server. The bot have several properties that can be manipulated by the user.It needs a remote repository from where the bot will checkout the code. The bot can be configured that it always runs when a new commit arrives at the remote git repository. The Ui Tests and the xCodeTests will be triggered every time someone pushes something into the Repository. Furthermore, there are several options to define post or pre scripts.

The current status of the Bot/App can be checked on the Mac OS Server Dashboard for Bots:



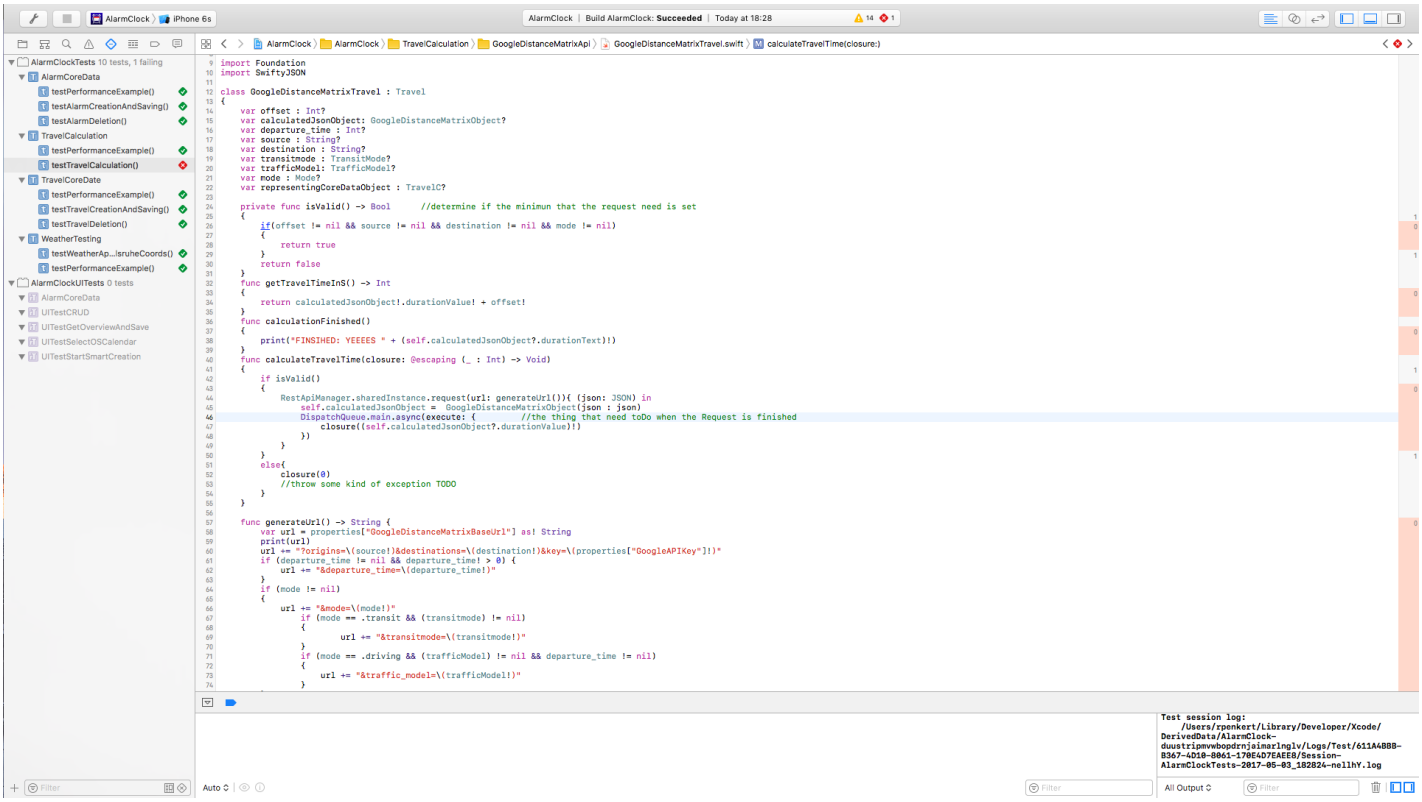
In Addition, we implemented Testing for our iOS App. In xCode the Tests are called “xcTest” that obviously mean Xcode Test.

The Tests for iOS and all other Operating Systems from Apple like tvOS,watchOS, macOS, are written in Swift or objective C. According to the fact that our App is written in Swift we developed the Testcases in Swift, too. Find more about Testing with xCode at the Apple Homepage: Apple Developer Documentation “About Testing with xCode” . Furthermore, the xcTests provide some APIs to make testing easier. For example there are two methods to separate the test code and the test data. “setUp()” is called before the test is called. With this method you can load data and or create new data before the actual test begins. “tearDown()” is called after the test is finished. It is meant for cleaning up the mess the test left behind 😉. The two methods results in a good separation of test data and the actual test code.

See here some example Tests:

- AlarmClockTest
- WeatherAPITest
- TravelCalculationTest

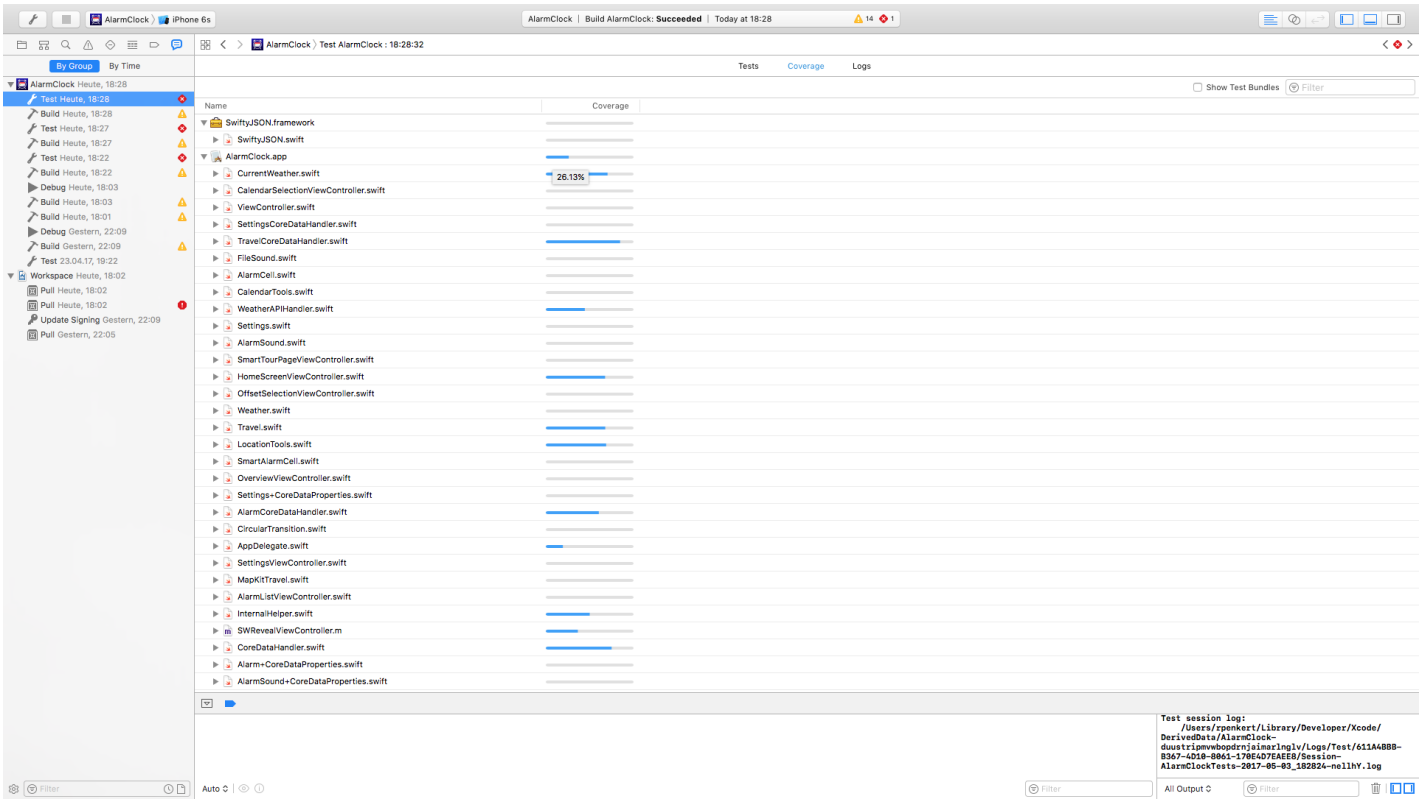
In the following Screenshot you can see on the left hand side the success of the Tests:



As you can see they are some succeed tests and one failed test.

In the Screenshot there is opened the Editor on the Source Code. You can see on the right side if your Source Code is covered by a test. If there is a red area with a zero the code in the editor is not covered. The number shows how often the Codeblock will be executed within all written tests.

Xcode have a Code Coverage Engine implemented. In the following Screenshot you can see how that is presented in Xcode:



You see all your Projectfiles with a bar that symbolizes how much of the file is covered by a test. When you scroll over a bar you can see a percentage. Our current overall Code Coverage is 26%.

We do not have a link for a maven file or something similar because for our project it is automatically build in our IDE “xCode”.

We provided a Code Coverage Badge on our Github Repo.

We documented everything in our Test Plan

← Function Points

Refactoring – Fowler →

2 thoughts on “Continous Integration and Testing”

Torben Krieger says:

4. May 2017 at 07:13

Hi Team,

glad to see that you managed to set up CI for the Apple Universe.

Do you know where the warning come from?

Your work fulfills all grading criteria.

Best regards,

Torben

Reply

Dominik Wunderlich says:

11. May 2017 at 06:48

Hi Benedikt and René,

it’s always very interesting to see how everything is done in the wo called Apple world.
I’m very happy to see that you have 31% code coverage as it is crucial to the quality of your iOS app.
Can you manage to get rid of the 12 warnings?
I think it’s cool that you already managed to get a github badge. How did you do that?

i wish you all the best for your application and continued fun with software engineering homeworks 😉
Dominik from team SAM

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT