INTEGRATION TEST RESEARCH

Kevin Oomen

FONTYS EINDHOVEN

Inhoud

What (What is the research question)

Our project group wants to know more about integration tests and how to implement them. If the results of this research are satisfactory, we will implement these tests in our project. The reason we want to investigate integration tests is because we want to test the communication between services. We have heard from our school learning environment that these kinds of tests can fulfil this need. That's the main reason for this research.

How (how will the research be performed)

The DOT-framework makes use of 5 research strategies:

Library

You use information that has already been made. You do this by checking if the information fits in the scope of your project and if it has any value to your project. If this is the case, you can use the information to further help you in your research.

Field

You use information about your users. You will have to make sure that your user's requirements get fulfilled. The way this is done is by doing interviews with your users and investigate what they would want to see.

Workshop

You mainly do this by prototyping. With prototyping I mean making a proof of concept for example. The reason for doing this is to see if you will get the desired results.

Lab

You use test methods to see if your solution has delivered the desired results, you can do this by using: security, system, unit and static test methods.

Showroom

Show what makes your solution different from others by communicating with experts or by comparing similar solutions to yours.

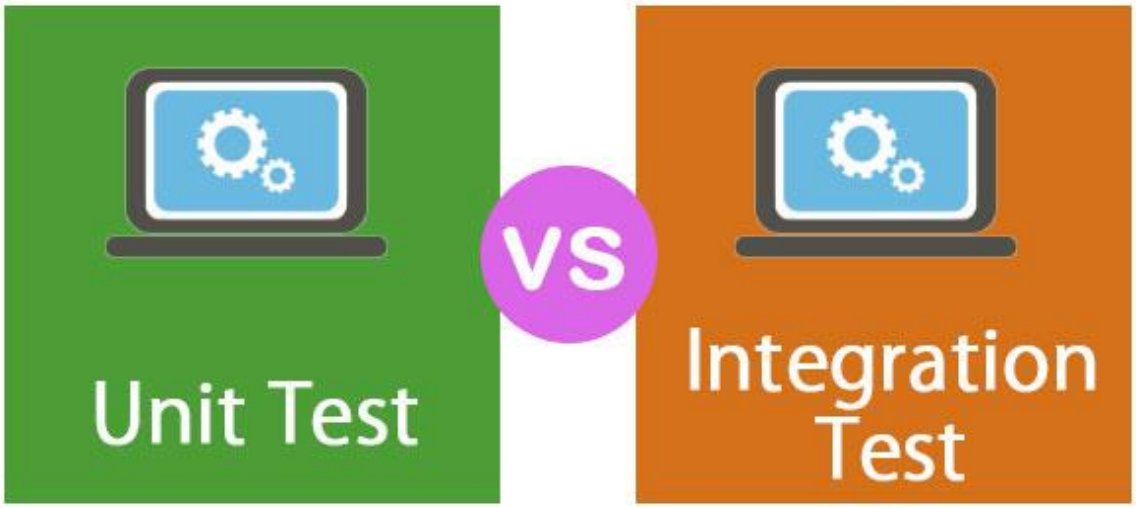
The information I have gathered to create this part can be found here.

Why (the reason for the research)

What does the research solve?

As a group we want a testing solution to be able to test the communication between services. That is why we ended up with integration tests. So now we want to see if integration tests are the solution, we are looking for. If this is the case, we can use these tests to test the communication between services and make sure the quality of the communication is good. If we don't implement tests of such sort, we can't assure the quality of our software.

Unit tests VS integration tests



What is the difference between a Unit test and an Integration test? Because these tests can be very similar. I'm going to answer the question: how unit tests and integration tests differ? First let's see what the definition is for both subjects:

Unit test

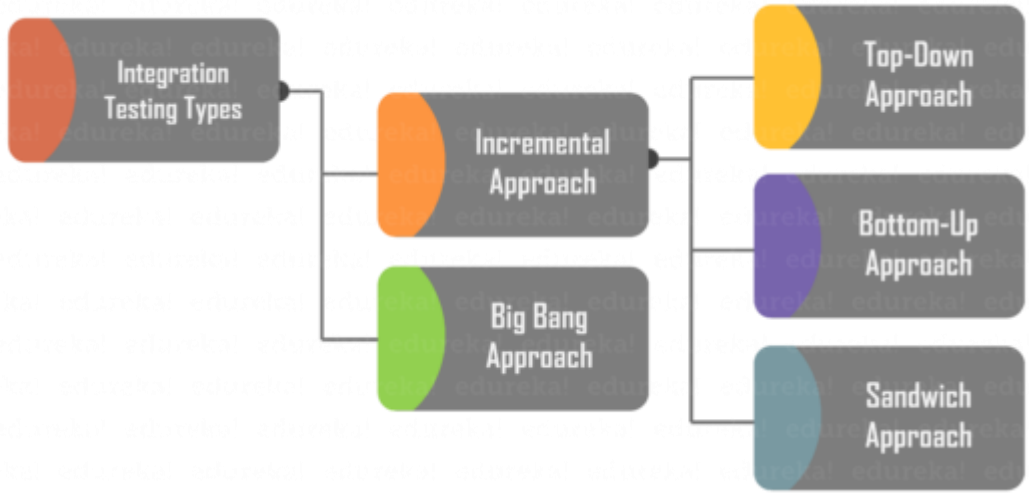
In computer programming, unit testing is a software testing method by which individual units of source code—sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures—are tested to determine whether they are fit for use. (Source)

Integration test:

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. (Source)

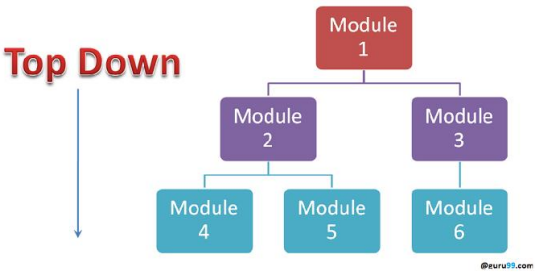
As you can see, both definitions have similarities. You could see an integration test as a unit test but focused on the communication between the services in your project. Unit tests are focused on single "units". They test the smallest testable pieces of code within a project and will mock every necessary external "piece". Integration tests test how these pieces work together by testing mainly the communication between these pieces. (Information)

What types of integration test methods are there?



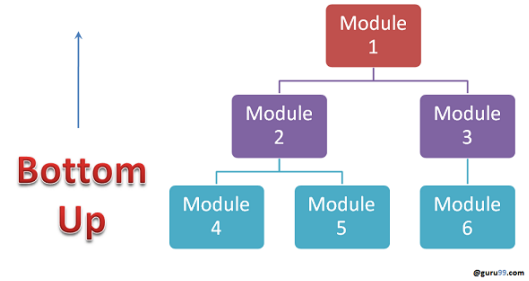
There are multiple types of integration test methods in this part we are going to look at a couple of them. Let's first see what the definitions of these types are:

Top-Down



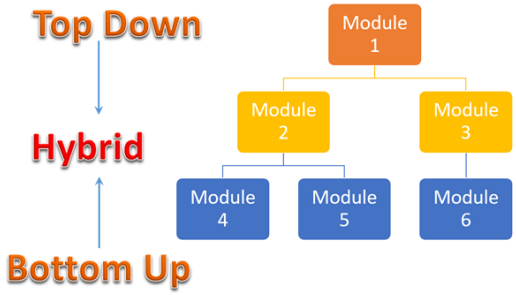
Top-Down Integration Testing is a method in which integration testing takes place from top to bottom following the control flow of software system. The higher-level modules are tested first and then lower-level modules are tested and integrated in order to check the software functionality. (Source)

Bottom-Up



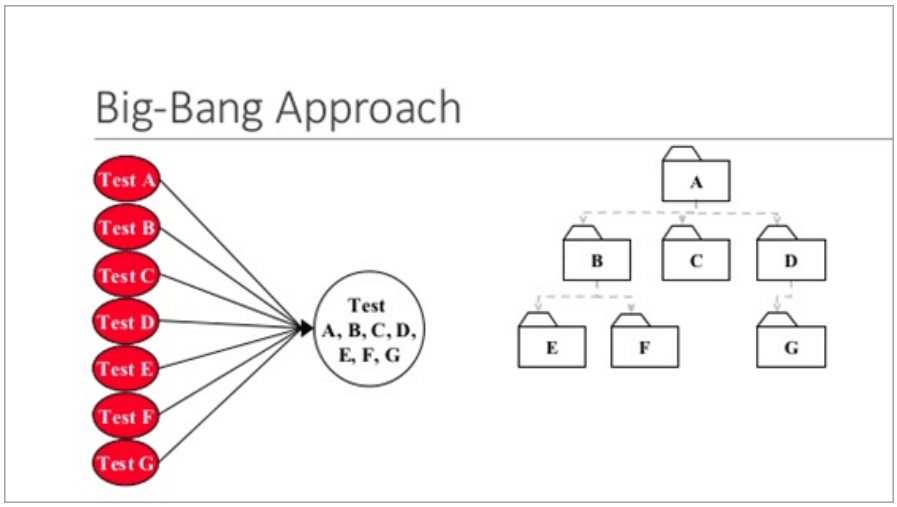
Bottom-Up Integration Testing is a strategy in which the lower-level modules are tested first. These tested modules are then further used to facilitate the testing of higher-level modules. The process continues until all modules at top level are tested. (Source)

Hybrid/Sandwich



Hybrid testing also called as sandwich testing is an approach to integration testing which is a combination of Top-Down and Bottom-Up approaches. Top modules are tested with lower modules at the same time lower modules are integrated with top modules and tested. and the layer below the target layer. (Source)

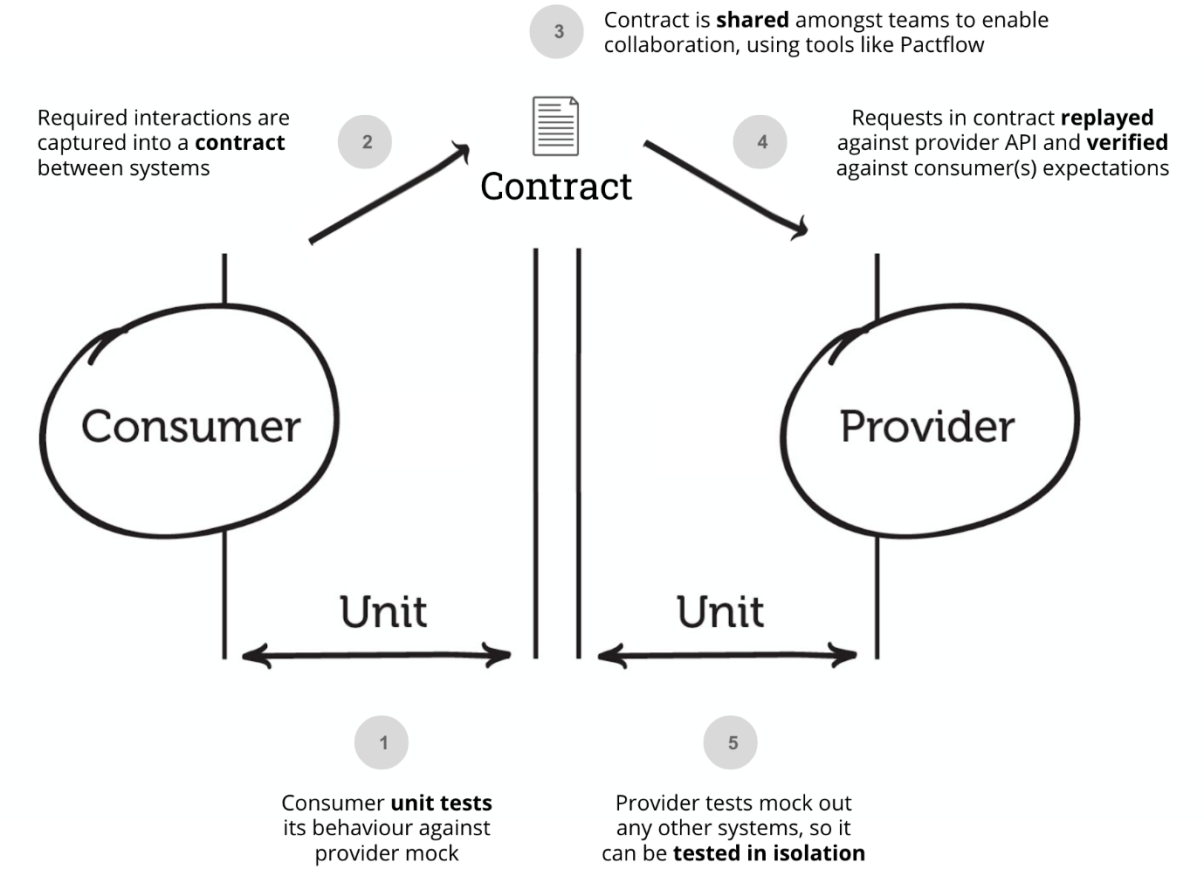
Big Bang



Big Bang Integration Testing is an approach in which all software components (modules) are combined at once and make a complicated system. This unity of different modules is then tested as an entity. According to this checking method, the integration process will not be executed until all components are completed. (Source)

As you can see there are quite a lot of different types of integration test methods. Some are more similar than others, but they are all different in such a way that they can be classified as. In the Top-Down strategy you test the high-level modules first by implementing the necessary lower modules for the functionality and go lower after each test. In the Bottom-Up strategy it's the complete opposite. You must start testing the low-level modules and go higher after each test. In the Hybrid/Sandwich strategy you must test each top-level module with every lower module that depends on it. So that it tests the high-level module with everything it has a relation with. In the Big Bang strategy, you must test all modules at the same time so it will be tested as one thing instead of multiple small modules.

What is contract testing?



If you have investigated integration tests before you might have heard about contract testing. But what is contract testing? And what does it have to do with integration tests? These are the question we are going to answer in this part.

What is it?

Simply said, contract tests are tests that make sure that services are able communicate with each other. It tests the interactions between the services. For example, http requests. You can test this by mocking a request and check if you get the desired return. This method of testing is mainly a counter against end-to-end integration testing. (Source)

End-to-end integration testing V.S. contract tests

End-to-end integration testing

- Are very slow, because the test will have to traverse through the entire ecosystem of the application. It also takes a lot of time to create the data required to perform these tests.
- Are hard to maintain, because these tests rely on the entire system to function as it should.
- They can be hard to maintain because you need to make sure that the application ecosystem is setup correctly.
- Are hard to fix because of how global these tests can be.
- Can cause the CI to be overloaded because of how slow these tests can be. Especially if you are working with a big team.
- Because these tests are so slow it's also possible to lose a lot of time and resources for your team if such a test fails.

Contract tests

- These tests are very fast because they don’t have to talk to multiple systems.
- These tests are easy to maintain because you don’t need to understand the entire application ecosystem to maintain these tests. You only need to understand the service you are testing.
- These tests are easy to debug because they are on a way smaller scale which will make finding the problem a lot easier.
- Because the tests are testing individual components. They won’t increase the time of a pipeline by much.
- These tests can be performed locally so that will make debugging a lot easier.

Proof of concept

I have implemented some integration tests in our group project to show the process of how they work. We are using microservices in the form of Spring applications for our group project. For clarification, these tests will be performed on the service called menu-service. This is a service that maintains the menu in our application.

What does it look like?

For this example, I will show a simple integration test that I have performed in the application:

```
@Test
void shouldDeleteCategoryById() throws Exception {
    mvc.perform(MockMvcRequestBuilders
            .delete("/menu/categories/delete/1")
            .contentType(MediaType.APPLICATION_JSON)
            .accept(MediaType.APPLICATION_JSON))
            .andExpect(status().isOk());
}
```

So, what does this code snipped do exactly? It checks if the service can delete a category by id. It sends a mocked delete request to the service. It sends this request to this URL:"/menu/categories/delete/1" where 1 is the id of the category that will be deleted. So, what does this function look like? Like this:

```
@DeleteMapping("/delete/{id}")
public void deleteCategoryById(@PathVariable("id") Long categoryId){
    categoryService.deleteCategoryById(categoryId);
}
```

This controller function will send the id to another function in the categoryService class. Which looks like this:

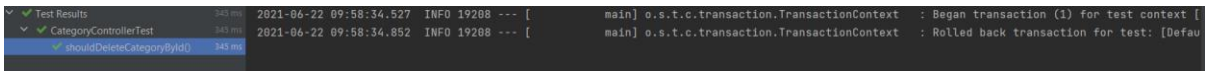
```
public void deleteCategoryById(Long categoryId) {
    categoryRepository.deleteById(categoryId);
}
```

JPA will handle the rest from here. But we shouldn't forget our data mock-ups because there won't be anything to delete otherwise. So, I made a SQL file that will be used for testing purposes. I will show the part which is important for this case:

```
INSERT INTO CATEGORY (CATEGORY_ID, NAME, DESCRIPTION, IMAGE_URL) VALUES
(1L,'Vlees','Vlees producten','https://fontys.nl/upload/50716580-70d3-4c39-
86dd-4237e0166f38_image6483968316988970112.png'),
(2L,'Fruit','Fruit producten','https://fontys.nl/upload/50716580-70d3-4c39-
86dd-4237e0166f38_image6483968316988970112.png'),
(3L,'Groenten','Groenten producten','https://fontys.nl/upload/50716580-
70d3-4c39-86dd-4237e0166f38_image6483968316988970112.png');
```

This SQL code will insert 3 categories into de category table. And as you may have noticed one of these categories has id 1. This will be the category that will be deleted after our test succeeds.

After we run this test, we could see if the test failed or succeeded in this case:



The test passed and the Mocked category data with CATEGORY_ID 1 was deleted. A test failing would also mean a desirable result because you will know that something is wrong with whatever you are testing instead of not noticing it in the first place. Now it gives you the option to see whatever is going wrong.

Implemented strategies

In this research I have mainly searched for existing information of the subject and have used this information to create an implementation for our group project. In this research I have used the following strategies:

Library

Methods

- Research on available articles
- Research in my interpretation

Implemented where?

- Unit tests VS integration tests
- What types of integration test methods are there?
- What is contract testing?

Workshop

Methods

- Prototyping

Implemented where?

- Proof of concept

Lab

Methods

- Test for desired results

Implemented where?

- Proof of concept

Conclusion

By doing this research we came to understand what the value is of integration testing and why we should use it in our project as well. To be able to test out services connections makes sure that the code quality stays a lot better. First off many of us couldn't see the difference between unit tests and integration tests. And even though they are quite similar we can distinguish them now.

Sources

DOT-framework (EN) - Lesmateriaal - Wikiwijs. (z.d.-b). wikiwijs. Geraadpleegd op 20 mei 2021, van https://maken.wikiwijs.nl/129804/DOT_framework__EN_#!page-4594586

Wikipedia contributors. (2021c, juni 13). *Unit testing*. Wikipedia. https://en.wikipedia.org/wiki/Unit_testing

Wikipedia contributors. (2021b, mei 26). *Integration testing*. Wikipedia. https://en.wikipedia.org/wiki/Integration_testing

What's the difference between unit tests and integration tests? (2011, 18 maart). Stack Overflow. https://stackoverflow.com/questions/5357601/whats-the-difference-between-unit-tests-and-integration-tests

Rungta, K. (2021, 9 mei). *Integration Testing: What is, Types, Top Down & Bottom Up Example*. Https://Www.Guru99.Com. https://www.guru99.com/integration-testing.html#:%7E:text=Top%2Ddown%20Integration%20Testing,to%20check%20the%20software%20functionality.

Rungta, K. (2021b, mei 9). *Integration Testing: What is, Types, Top Down & Bottom Up Example*. Https://Www.Guru99.Com. https://www.guru99.com/integration-testing.html#:%7E:text=Bottom%2Dup%20Integration%20Testing%20is,at%20top%20level%20are%20tested.

Faculty, P. S. (2018, 26 december). *HYBRID TESTING - SOFTWARE TESTING*. H2kinfosys Blog. https://www.h2kinfosys.com/blog/hybrid-testing/

Big-Bang Testing Specifics – QATestLab. (z.d.). Https://Qatestlab.Com. Geraadpleegd op 20 mei 2021, van https://qatestlab.com/resources/knowledge-center/big-bang-testing/

Fellows, M. (2021, 29 maart). *What is contract testing?* Pactflow Contract Testing Platform. https://pactflow.io/blog/what-is-contract-testing/