

## Why is Architecture Testing important?

Application architecture defines the model of application & tells us how it will function. It makes it easier to understand the whole system and makes the decision-making process more efficient.

## ArchUnit: A library to Test Java Application Architecture

ArchUnit can check dependencies between *packages* and *classes*, *layers* and *slices*, check for *cyclic dependencies* and more. It allows us to implement rules for application architecture in the form of executable tests such as the following:

- Package dependency checks
- Class dependency checks
- Class and package containment checks
- Inheritance checks
- Annotation checks
- Layer checks
- Cycle check

Furthermore, ArchUnit lets you implement *custom tests* for constructors, methods, fields to verify specific architectural constraints as well as coding rules, such as naming conventions.

## ArchUnit with Java Test Framework:

ArchUnit can work with any test framework that executes Java code. To use ArchUnit in such a context, include the core ArchUnit dependency from Maven Central:

```
<dependency>
  <groupId>com.tngtech.archunit</groupId>
  <artifactId>archunit</artifactId>
  <version>0.20.1</version>
  <scope>test</scope>
</dependency>
```

Following example will give an idea to use ArchUnit:



```
noClasses().that().resideInAPackage("..source..")
    .should().dependOnClassesThat().resideInAPackage("..foo..")
```

This is a simple test to check dependency between “*source Package*” & “*foo Package*”.

## NetArch: Library to Test .NET Application Architecture

NetArchTest is a basic implementation inspired by parts of the ArchUnit library. It allows us to write unit tests that enforce architectural and design rules in the .NET application.

With NetArchTest one can test the application for *immutability*, *encapsulation*, *type dependencies*, *naming conventions*, *project layering*, and more.

Simple example of NetArchTest to check Naming Convention of class:

```
test = extent.CreateTest("service_name_should_start_with_service_word");
result = Types.InCurrentDomain()
    .That()
    .ResideInNamespace("NetArchTestApp.Service")
    .Should()
    .HaveNameStartingWith("Service").GetResult();
```

Above test ensures that all service classes in the application should start with service word only.

## Conclusion

With *ArchUnit* and *NetArchTest* we can build group of architecture rules and use it again. Even if a new member joins the team, they will learn the rules quickly as these rules are like simple English sentence.

## References

1. <https://github.com/BenMorris/NetArchTest>
2. [https://www.archunit.org/userguide/html/000\\_Index.html](https://www.archunit.org/userguide/html/000_Index.html)