Software Design H.W 1 – Dry Part

Sahar Cohen, 206824088
Yuval Nahon, 206866832

We'll discuss the usage of dependency injection (abbreviated "DI" henceforth) in the Android project https://github.com/RocketChat/Rocket.Chat. Android.

Like any Android app – the relevant modules we're discussing here are present under app/src/main/[app domain name].

Configuration of the DI:

The project is organized in such way that most packages (seems like the ones that contain more complex modules) also have a "DI" package that contains modules and providers and binds the classes under that package. There's also a "dagger" package in the root package: we think the classes inside this package inject dependencies for the framework itself (Application class & SharedPreferences etc).

This structure follows the notion of "SSoT" (single source of truth) design, and in turn allows for easier testing and debugging because of the centralized nature of these dependencies which can be changed once and utilized from anywhere else in the project's modules. Of course, the usage of DI improves class decoupling (that's why it makes unit testing easier).

RocketChatClient & CreateChannelView – Overview:

- RocketChatClient is not present in the project's source code and is provided externally through a separate SDK. Because of this, the RocketChatClientFactory singleton is being passed as a dependency for the creation of RocketChatClient instances.

- CreateChannelView is a pure interface. Its implementations are provided by the CreateChannelModule class (located under the DI package as summarized above). This class is annotated by the *@Module* annotation and its methods are annotated by the *@Provides* annotation: Dagger uses these annotations to define classes and methods (respectively) that provide dependencies. In particular, the createChannelView method is a provider that provides CreateChannelView instances.

Injecting another implementation for RocketChatClient & CreateChannelView:

- The RocketChatClientFactory singleton is being passed, so we'd change the implementation of this factory's getter method to produce the alternative implementation of RocketChatClient

- The only method we'd have to change is the provider method that was discussed above: createChannelView in the CreateChannelModule.