

RocketChatClientFactory:

- @Inject added to its constructor.
- @Singleton before declaration of the constructor.
- @Module added before class MainModule, different than “: kotlinModule” in Guice.
- @Provides, binds an implementation of a factory.
- @PerActivity comes with “@Provides”, activates the supplied function whenever the factory creates an object.
- RocketChatClient have an object builder.

CreateChannelModule:

- is a class module annotated with @Module.
- @Provides and @PerFragment are added to the function: createChannelView inside the class, which create a channel view. This is the binding between the interface CreateChannelClientView to CreateChannelFragment.

As we see in RocketChatClientFactory an instance of RocketChatClient is created with a builder that gets all its dependencies with concrete objects.

A better way to create an instance of RocketChatClient is to bind all the dependencies like: httpClient, restUrl, userAgent, ... with Dagger without using a Builder pattern.

If we would like to change the implementation of CreateChannelView, then we need to change the configuration at CreateChannelModule to the new implementation.

If we would like to change the implementation of RocketChatClient, then we need to change the configuration of the get function at RocketChatClientFactory, specifically change the objects that RocketChatClient builds supplied with.