תיכון תוכנה 1 – חלק יבש

ראשית, נתאר את המבנה הכללי של DI באמצעות Dagger. מחלקה אשר נרצה לעשות לה DI (כלומר להזריק לה מימושים שונים) תסומן באנוטציה @Inject. המימושים נמצאים במודול המסומן באנוטציה @Module, כאשר מימוש ספציפי מסומן באנוטציה @Provides. המודולים בהם נשתמש יוצהרו תחת האנוטציה @Component (כפי שניתן לראות למשל בקובץ AppComponent.kt שנמצא בתיקיה dagger).

בפרויקט RocketChat מתבצע DI למחלקות RocketChatClient ו-CreateChannelView. כדי לקבל מופעים של המחלקת RocketChatClient, יש שימוש בRocketChatClientFactory אשר לו עושים DI. נסתכל על הגדרת המחלקה RocketChatClientFactory:

```
@Singleton
class RocketChatClientFactory @Inject constructor(
    private val okHttpClient: OkHttpClient,
    private val repository: TokenRepository,
    private val logger: PlatformLogger
) {
```

נשים לב שהיא מסומנת באנוטציה @Inject. מכך נסיק שלפחות אחד המאפיינים שניתן לראות בתמונה מוזרק. מימוש אותו אנו מזריקים מסומן באנוטציה @Provides, והמימוש שלו מוגדר בAppModule שנמצא בתיקיה dagger. באופן דומה גם TokenRepository מסומן ב@Provides וכך גם PlatformLogger. כלומר, כדי להחליף מימושים, כל שעלינו לעשות זה להחליף את הprovider. בנוסף, השימוש באנוטציה @Singleton מסמן שאחד המאפיינים המוזרקים הוא singleton.

CreateChannelView הוא ממשק, אשר המימוש שלו נמצא במחלקה CreateChannelModule בפונקציה createChannelView אשר מסומנת גם היא באנוטציה @Provides. לכן, כדי להחליף את המימוש של CreateChannelView, עלינו להחליף את המימוש של הפונקציה createChannelView. דרך נוספת להחליף מימוש היא להגדיר module חדש עם מימוש אחר, ולהשתמש בmodule החדש במקום הנוכחי.