**POM in Appium**

Page Object Model design pattern in Appium is similar to Selenium but we have to use @AndroidFindBy instead @FindBy & we have to create object for AppiumFieldDecorator class while initializing the elements

**PageFactory**

PageFactory is a tool used in Appium to create page object models for mobile apps. It simplifies the process of creating & accessing UI elements in the app by automatically initializing them & allowing them to be easily accessed by the automation code. This makes the automation code more organized & easier to maintain.

**@AppiumFindBy**

* It is annotation of PageFactory.
* It helps us to locate mobile element in a Mobile App.
* We can locate elements like textField, Button, Link, icon.

Ex-

@AndroidFindBy(id="com.androidsample.generalstore:id/spinnerCountry")

**private** WebElement selectCountry;

**AppiumFieldDecorator(C)**

* It is a Class of PageFactory.
* It automatically adds necessary codes interact with Mobile App’s user interface elements like buttons, textfields.
* This makes mobile app automation faster, easier & less error-prone.

Ex-

AndroidDriver driver;

**public** GeneralStore\_LandingPage(AndroidDriver driver)

{

**this**.driver=driver;

PageFactory.*initElements*(**new** AppiumFieldDecorator(driver), **this**);

}

**Rules of POM in Appium**

1. We must create separate class for each & every page in the app.
2. We should locate mobile element using @AppiumFindBy annotation.
3. We should initialize the mobile element by creating constructor.
4. We must create object for AppiumFieldDecorator as an argument for initElement method.
5. We should create getters() & business logic as per our requirement.

**Framework**

Framework is a set of rules or collection of components like gesture Utility, DriverUtility, object repository, FileUtility classes to automate a Mobile Application.

