

Android UI Automation with Espresso



Jeriel Ng (@jerielng)

Mobile Software Engineer

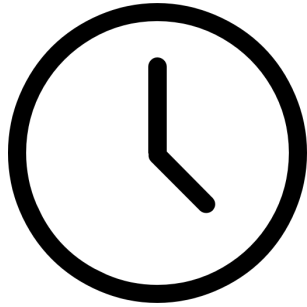
Types of Testing

- Unit Testing
- Manual UI Testing
- **Automated UI Testing**

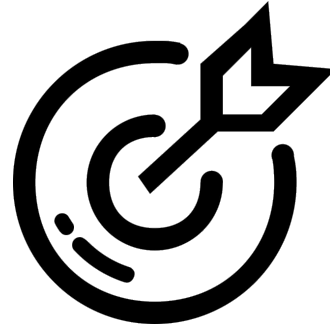


CommitStrip.com

Why should we automate UI testing?



Time-efficient



Accurate & Consistent

Espresso: An Overview

- Android's native UI testing framework
- Simulates user interactions with application views
- Usage: Verify correct application behavior
 - New feature development
 - Feature removal
 - Code refactoring



Espresso: Usage



- Procedure:
 - **Espresso** - set an entry point
 - **ViewMatchers** - find a view
 - **ViewActions** - perform an action on a view
 - **ViewAssertions** - verify your behavior
- Sample functions
 - onView()
 - perform()
 - click()
 - scrollTo()
 - check()
 - onData()

Considerations

- Write out your test cases first, then implement them
- Keep your tests as simple as possible
- Accurate timing is essential:
 - Too fast - may return false negatives
 - Too slow - increases total test suite time

Demo

Resources

- [Espresso Official Documentation](#)
- [Efficient Android Testing Using WireMock And Espresso](#)
- [GitHub repo for Espresso demo code](#)

Questions
