

透過自動化測試和工具提升 iOS 開發

這篇博文是由 ChatGPT-4o 協助整理的。

單元測試的重要性

在 LeanCloud，我們在項目早期就實施了單元測試，這被證明是非常有價值的。每個拉取請求 (PR) 都會在 Jenkins 上觸發單元測試，我們的覆蓋率目標約為 80%。編寫測試的主要場景有兩個：驗證新接口和重現和修復錯誤。積累的測試越多，我們的代碼庫就越強大。自動化測試使我們能夠自信地發布和重構代碼，而無需手動驗證。

測試流程和實際應用

以下是一些單元測試如何幫助我們的實際例子：

測試流程 1: 有用戶報告在保存帶有描述鍵的對象時會報錯。我寫了一個測試來重現這個問題，找到了問題並修復，然後保留了這個測試以供將來驗證。

測試流程 2: 在開發新接口時，我在實現代碼後寫了相應的測試，以確保代碼正常工作。

測試流程 3: 修改 `AVObject.m` 代碼後，我運行 `AVObjectTest.m` 測試，查看改動是否導致任何測試失敗。

測試流程 4: 提交 PR 會在 Jenkins 上觸發自動化測試。

編寫單元測試的好處

- **減少手動驗證:** 單元測試通過消除手動檢查來節省時間。
- **錯誤檢測:** 提前檢測由於代碼更改引起的問題，防止錯誤影響項目的其他部分。
- **協作項目:** 在多開發者項目中，單元測試確保一致性和可靠性，即使項目交由他人處理。
- **高質量的開源項目:** 流行的開源項目通常具有廣泛的單元測試，這有助於它們的可靠性和受歡迎程度。

如何編寫有效的單元測試

- **模塊化代碼:** 分離數據層和 UI 層以便於測試。
- **最大化覆蓋率:** 使用最少的測試代碼實現最大的覆蓋率。

- **異步處理:** 確保測試能夠處理異步操作。
- **框架選擇:** 選擇適合需求的測試框架。
- **覆蓋率報告:** 使用覆蓋率報告了解代碼的哪些部分被測試了。

評估測試框架

我們評估了幾個框架：- **Expecta:** `expect(error).not.beNil()` - **Specta:** `describe("") it("")` - **Kiwi:** `describe("") it("")` - TDD 和 BDD 框架有一些限制，例如與 Xcode 集成不好，沒有測試按鈕，側邊欄沒有列出所有單元測試。

處理異步測試

異步測試對於不立即完成的操作至關重要。確保你的框架有效支持異步測試。例如，在 XCTest 中使用 expectations 來等待異步操作完成，然後再進行斷言。

覆蓋率報告

Xcode 7 引入了內置的覆蓋率報告功能。啟用步驟如下：1. 在方案設置中開啟 Gather Coverage Data。2. 針對 App Target 進行測試，而不是 Test Target。

這個功能允許開發者確切地看到哪些代碼行被測試了，有助於識別未測試的代碼部分。更多詳情請訪問 Big Nerd Ranch 的博客。

使用 Jenkins 進行遠程自動化測試

設置 Jenkins 進行自動化測試涉及幾個步驟：1. **安裝 Jenkins:** 在本地機器或數據中心服務器上設置 Jenkins。2. **GitHub 集成:** 使用 GitHub PR 構建插件在提交拉取請求時觸發測試。- 配置 Webhooks 將事件發送到 Jenkins。- 確保 Jenkins 能夠訪問拉取請求的最新代碼。3. **測試腳本:** 在 Jenkins 中設置測試腳本以自動化測試過程。- 確保 Jenkins 能夠通知 GitHub 測試結果。- 配置 Slack 或電子郵件通知測試失敗。

使用 Jenkins 進行遠程自動化測試提供了自動化測試的全部好處，超過本地測試，通過在一個乾淨、受控的環境中運行測試。

遠程打包和部署

雖然不是所有項目都需要遠程打包，但它可以簡化 SDK 和其他可重用組件的部署過程。步驟包括：
- 配置 Jenkins 讀取代碼。- 讀取發布版本。- 在命令行中解鎖鑰匙串以訪問簽名證書。

額外工具和提示

- **Reveal:** 在越獄 iPhone 上分析任意 App 的 UI 界面。
- **Flex:** 在越獄 iPhone 上分析網絡請求、UI、本地文件、NSUserDefaults 和日誌。
- **Pod 管理:** 本地 Pod 使用、高級 Podfile 配置和發布 Pods。
- **框架創建:** 動態庫和靜態庫的區別，以及如何打包同時適用於模擬器和真機的框架。
- **Xcode 提示:** 有用的快捷鍵，如 Shift + Command + J 在導航器中顯示文件，Shift + Command + O 快速打開文件。

結論

自動化測試和合適的工具顯著提升了開發過程。通過早期引入單元測試、利用異步處理和使用覆蓋率報告，我們可以構建更可靠和可維護的應用程序。像 Jenkins 這樣的 CI/CD 工具和 Xcode 開發工具結合一個強大的測試策略，確保高質量的軟件交付。

致謝

特別感謝 LeanCloud 團隊和所有為我們的測試流程做出貢獻的人。