

通过自动化测试和工具提升 iOS 开发

这篇博文是由 *ChatGPT-4o* 协助整理的。

单元测试的重要性

在 LeanCloud，我们在项目早期就实施了单元测试，这被证明是非常有价值的。每个拉取请求（PR）都会在 Jenkins 上触发单元测试，我们的覆盖率目标约为 80%。编写测试的主要场景有两个：验证新接口和重现和修复错误。积累的测试越多，我们的代码库就越强大。自动化测试使我们能够自信地发布和重构代码，而无需手动验证。

测试流程和实际应用

以下是一些单元测试如何帮助我们的实际例子：

测试流程 1： 有用户报告在保存带有描述键的对象时会报错。我写了一个测试来重现这个问题，找到了问题并修复，然后保留了这个测试以供将来验证。

测试流程 2： 在开发新接口时，我在实现代码后写了相应的测试，以确保代码正常工作。

测试流程 3： 修改 `AVObject.m` 代码后，我运行 `AVObjectTest.m` 测试，查看改动是否导致任何测试失败。

测试流程 4： 提交 PR 会在 Jenkins 上触发自动化测试。

编写单元测试的好处

- **减少手动验证：** 单元测试通过消除手动检查来节省时间。
- **错误检测：** 提前检测由于代码更改引起的问题，防止错误影响项目的其他部分。
- **协作项目：** 在多开发者项目中，单元测试确保一致性和可靠性，即使项目交由他人处理。
- **高质量的开源项目：** 流行的开源项目通常具有广泛的单元测试，这有助于它们的可靠性和受欢迎程度。

如何编写有效的单元测试

- **模块化代码：** 分离数据层和 UI 层以便于测试。
- **最大化覆盖率：** 使用最少的测试代码实现最大的覆盖率。
- **异步处理：** 确保测试能够处理异步操作。
- **框架选择：** 选择适合需求的测试框架。
- **覆盖率报告：** 使用覆盖率报告了解代码的哪些部分被测试了。

评估测试框架

我们评估了几个框架：- **Expecta**: `expect(error).not.beNil()` - **Specta**: `describe("") it("")` - **Kiwi**: `describe("") it("")` - TDD 和 BDD 框架有一些限制，例如与 Xcode 集成不好，没有测试按钮，侧边栏没有列出所有单元测试。

处理异步测试

异步测试对于不立即完成的操作至关重要。确保你的框架有效支持异步测试。例如，在 XCTest 中使用 `expectations` 来等待异步操作完成，然后再进行断言。

覆盖率报告

Xcode 7 引入了内置的覆盖率报告功能。启用步骤如下：1. 在方案设置中开启 `Gather Coverage Data`。2. 针对 App Target 进行测试，而不是 Test Target。

这个功能允许开发者确切地看到哪些代码行被测试了，有助于识别未测试的代码部分。更多详情请访问 [Big Nerd Ranch](#) 的博客。

使用 Jenkins 进行远程自动化测试

设置 Jenkins 进行自动化测试涉及几个步骤：1. **安装 Jenkins**: 在本地机器或数据中心服务器上设置 Jenkins。2. **GitHub 集成**: 使用 GitHub PR 构建插件在提交拉取请求时触发测试。- 配置 Webhooks 将事件发送到 Jenkins。- 确保 Jenkins 能够访问拉取请求的最新代码。3. **测试脚本**: 在 Jenkins 中设置测试脚本以自动化测试过程。- 确保 Jenkins 能够通知 GitHub 测试结果。- 配置 Slack 或电子邮件通知测试失败。

使用 Jenkins 进行远程自动化测试提供了自动化测试的全部好处，超过本地测试，通过在一个干净、受控的环境中运行测试。

远程打包和部署

虽然不是所有项目都需要远程打包，但它可以简化 SDK 和其他可重用组件的部署过程。步骤包括：- 配置 Jenkins 读取代码。- 读取发布版本。- 在命令行中解锁钥匙串以访问签名证书。

额外工具和提示

- **Reveal**: 在越狱 iPhone 上分析任意 App 的 UI 界面。
- **Flex**: 在越狱 iPhone 上分析网络请求、UI、本地文件、NSUserDefaults 和日志。
- **Pod 管理**: 本地 Pod 使用、高级 Podfile 配置和发布 Pods。
- **框架创建**: 动态库和静态库的区别，以及如何打包同时适用于模拟器和真机的框架。
- **Xcode 提示**: 有用的快捷键，如 `Shift + Command + J` 在导航器中显示文件，`Shift + Command + O` 快速打开文件。

结论

自动化测试和合适的工具显著提升了开发过程。通过早期引入单元测试、利用异步处理和使用覆盖率报告，我们可以构建更可靠和可维护的应用程序。像 Jenkins 这样的 CI/CD 工具和 Xcode 开发工具结合一个强大的测试策略，确保高质量的软件交付。

致谢

特别感谢 LeanCloud 团队和所有为我们的测试流程做出贡献的人。