

Использование `tuist` для генерации `xcodeproj`

Tuist

- Open source консольная утилита для работы со структурой проекта
- Основная задача: Генерация проекта из swift файла конфигурации при этом сохраняется иерархия папок и файлов
- Также упрощает работу с ресурсами, модуляризацией, тестами

Достоинства

Основные достоинства tuist

- Файл конфигурации легко редактируется с помощью Xcode через terminal (tuist edit) с автодополнениями, подсказками и прочими плюсами IDE
- Автоматизация создания xcodeproj файлов для избежания merge конфликтов
- Автоматизация работы с ресурсами
- Упрощение модуляризации за счет генерации фреймворков и возможностью прописать зависимости
- Можно вынести часто используемые константы в enum(например версию приложения, название, бандл идентифайр)
- Можно выводить разные сообщения в консоль, что может быть полезно при отладке
- Можно интегрировать с CI/CD, например через Fastfile
- Возможность кеширования зависимостей для ускорения сборки
- Скрипты на Swift для замены Ruby и Shell команд
- Построение графа зависимостей проекта
- Легко выпилить при необходимости
- Удобная и масштабируемая структура проекта

Недостатки

Основные недостатки tuist

- Необходимо вручную создавать начальный файл конфигурации, сложность начальной конфигурации
- Не всегда очевидно какое значение использовать для ключа в proj файле например "UISupportedInterfaceOrientations":
[«UIInterfaceOrientationPortrait»]
- Необходимость при добавлении файлов или например смене рабочей ветки делать tuist generate
-

Пример файла конфигурации

```
import ProjectDescription

let target = Target(
    name: "App",
    platform: .iOS,
    product: .app,
    bundleId: "io.tuist.Demo",
    deploymentTarget: .iOS(targetVersion: "13.0", devices: .iphone),
    infoPlist: "App/Info.plist",
    sources: ["App/**"],
    resources: [
        "App/Resources/**",
        "App/**/*.storyboard"
    ],
    dependencies: [],
    settings: Settings(
        configurations: [
            .debug(
                name: "Debug",
                xcconfig: "App/Configurations/Debug.xcconfig"
            ),
            .release(
                name: "Release",
                settings: SettingsDictionary()
                    .bitcodeEnabled(true)
                    .swiftCompilationMode(.wholemodule)
            ),
            .release(
                name: "Beta",
                xcconfig: "App/Configurations/Beta.xcconfig"
            )
        ]
    )
)

let project = Project(
    name: "TuistDemo",
    targets: [target]
)
```