

Инфраструктура сборки F#

Николай Пономарев

FAKE

FAKE¹

¹Официальный сайт

FAKE

FAKE¹ (F# Make) — система автоматизации сборки и релиза приложений. Целью проекта является замена «устаревших» скриптов на `bash` или `powershell`, скриптами на F#.

Способы использования FAKE:

- Уменьшение количества зависимостей в CI/CD
- Создание воспроизводимых и тестируемых скриптов
- Замена скриптов написанных на `bash/powershell`
- Автоматизация ручных действий

¹Официальный сайт

Использование FAKE

FAKE можно запускать несколькими способами:

- FAKE runner
 - По сути заменяет MSBuild и устанавливается как отдельное приложение
- F# interactive (FSI)
 - Выполняется как обычный F# скрипт
- Использование готового проекта
 - MiniScaffold
 - SAFE-Dojo

Пример использования FAKE

```
#r "paket:
nuget Fake.Core.Target //"
open Fake.Core
// *** Define Targets ***
Target.create "Clean" (fun _ -> Trace.log " --- Cleaning stuff
    ↪ --- ")
Target.create "Build" (fun _ -> Trace.log " --- Building the app
    ↪ --- ")
Target.create "Deploy" (fun _ -> Trace.log " --- Deploying app
    ↪ --- ")
open Fake.Core.TargetOperators
// *** Define Dependencies ***
"Clean" ==> "Build" ==> "Deploy"
// *** Start Build ***
Target.runOrDefault "Deploy"
```

Paket

Paket² — замена NuGet из мира F#

Зачем оно надо:

- Правильная работа с транзитивными зависимостями
 - NuGet не различает транзитивные и прямые зависимости
 - NuGet выбирает версии не самым адекватным образом

²Официальный сайт

MiniScaffold

MiniScaffold³ — готовый шаблон проекта для F#.

Зачем это надо:

- Уже готовая структура проекта
- Тщательно подобранные библиотеки
- Готовые настройки репозитория
- Инфраструктура, заточенная под F#

Что включено:

- Автоматизация сборки через FAKE
- Управление пакетами через Paket
- Модульное тестирование и оценка покрытия с помощью Expecto/Altcover
- Форматирование кода с помощью Fantomas

³Репозиторий