

Инфраструктура сборки F#: FАКЕ, Paket

Арабелла Громова
st098246@student.spbu.ru

19.05.2023

FAKE

- ▶ Open-source кроссплатформенная система автоматизации сборки
- ▶ Интегрирован в F#
 - ▶ Обширная библиотека классов
 - ▶ VS Code+Ionide, Visual Studio, Rider

Почему FAKE?

- ▶ Работа поверх файлов скриптов
- ▶ Упрощение запуска внешних процессов с помощью различных API
- ▶ Встроенная поддержка Git, NuGet, Unit тестов, Octopus Deploy, Xamarin и т.д.
- ▶ Обеспечение непрерывной интеграции











Как запустить FAKE?

- ▶ FAKE Runner
- ▶ F# interactive (FSI)
- ▶ Специальный проект сборки (.NET 7)

Запуск FAKE

- ▶ `dotnet new -i "fake-template::*"`
- ▶ `dotnet new fake`
- ▶ `dotnet fake build`

Запуск FAKE

 .config	18.05.2023 23:52	Папка с файлами	
 bin	18.05.2023 23:52	Папка с файлами	
 build	18.05.2023 10:52	Папка с файлами	
 ConsoleApp1	18.05.2023 12:09	Папка с файлами	
 obj	18.05.2023 23:52	Папка с файлами	
 build.fsx	18.05.2023 23:53	Исходный файл F...	1 КБ
 fake.cmd	18.05.2023 23:53	Сценарий Windo...	1 КБ
 fake.sh	18.05.2023 23:53	sh_auto_file	1 КБ
 Program.cs	18.05.2023 23:51	C# Source File	1 КБ
 RiderProjects.csproj	18.05.2023 23:51	C# Project File	1 КБ

build.fsx

```
#r "paket:  
nuget Fake.DotNet.Cli  
nuget Fake.IO.FileSystem  
nuget Fake.Core.Target //"br/>#load ".fake/build.fsx/intellisense.fsx"  
open Fake.Core  
open Fake.DotNet  
open Fake.IO  
open Fake.IO.FileSystemOperators  
open Fake.IO.Globbing.Operators  
open Fake.Core.TargetOperators  
  
Target.initEnvironment ()  
Target.create "Clean" (fun _ -> !! "src/**/bin" ++ "src/**/obj" |> Shell.cleanDirs)  
Target.create "Build" (fun _ -> !! "src/**/*.*proj" |> Seq.iter (DotNet.build id))  
Target.create "All" ignore  
"Clean" ==> "Build" ==> "All"  
  
Target.runOrDefault "All"
```

Paket

- ▶ Менеджер зависимостей для .NET с поддержкой пакетов NuGet и репозиториях git
- ▶ Возможность ссылаться на файлы с HTTP-ресурсов

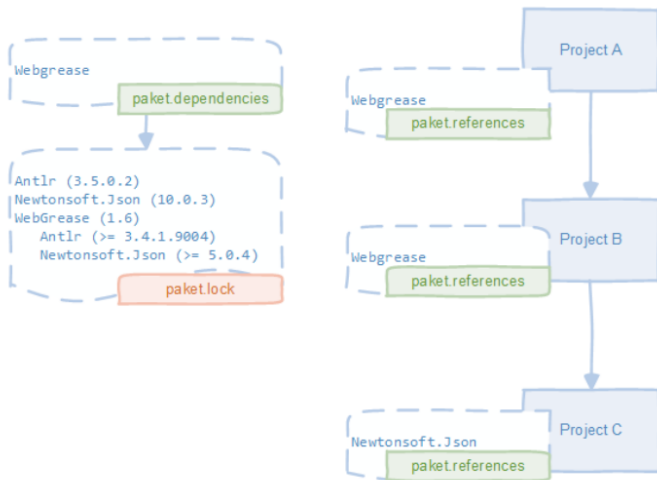
Почему Paket?

- ▶ Обеспечивает точный и предсказуемый контроль над зависимостями
 - ▶ Разделяет транзитивные зависимости
 - ▶ Нет конфликта версий
 - ▶ Зависимости решения

Как использовать Paket?

- ▶ Установка Paket
 - ▶ `dotnet new tool-manifest`
 - ▶ `dotnet tool install paket`
- ▶ Инициализация Paket
 - ▶ `dotnet paket convert-from-nuget`
- ▶ Добавления пакета в проект
 - ▶ `cd src/project-to-add-package-to`
 - ▶ `dotnet paket add <package>`

Paket



Полезные ссылки

- ▶ Документация FAKE
- ▶ Документация Paket
- ▶ <https://www.youtube.com/@bengobeil933>