OZON tech

Что нового в .NET10: dotnet tools

Кочубеев Николай
Старший разработчик, Ozon

О чём будет доклад

- Что такое .NET tools и CLI
- Проблемы «до»
- Что нового в .NET 10:
  - platform-specific tools
  - one-shot / dnx execution
  - CLI introspection и UX
- Выводы

.NET CLI

Больше, чем build/run

Tooling как часть платформы

Tooling как часть платформы

01

Сборки

Tooling как часть платформы

01

Сборки

02

Тестирования

Tooling как часть платформы

01

Сборки

02

Тестирования

03

Публикации

Tooling как часть платформы

01

Сборки

02

Тестирования

03

Публикации

04

Управления SDK и инструментами

Tooling как часть платформы

01

Сборки

02

Тестирования

03

Публикации

04

Управления SDK
и инструментами

05

Инструменты версионируются вместе с проектом

Tooling как часть платформы

01

Сборки

02

Тестирования

03

Публикации

04

Управления SDK
и инструментами

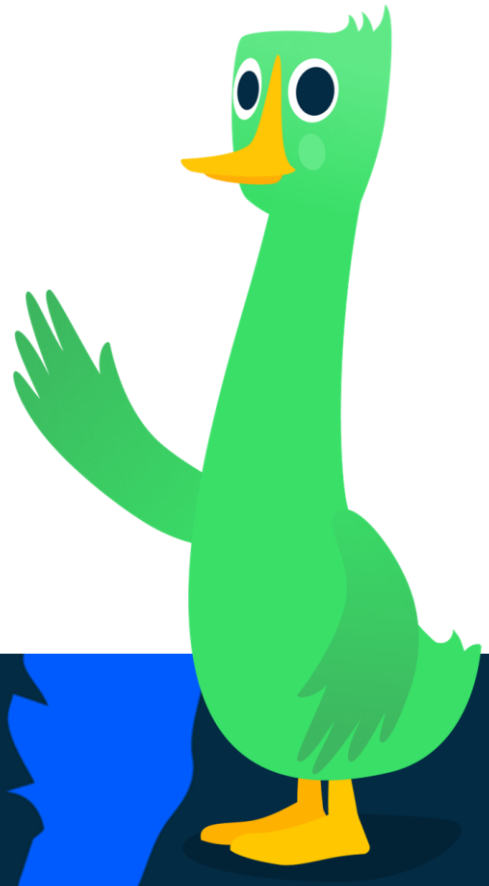
05

Инструменты
версионируются
вместе с проектом

06

**Одинаковый опыт
на Windows / macOS /
Linux**

dotnet CLI — это не просто запуск приложений, а **единая точка входа** во всю .NET платформу



Как было раньше (до .NET 10)

Как было раньше (до .NET 10)

Сложно с native / AOT

Как было раньше (до .NET 10)

- Сложно с native / AOT
- Отдельные пакеты под платформы

Как было раньше (до .NET 10)

- Сложно с native / AOT
- Отдельные пакеты под платформы
- **Ручной выбор RID**

Как было раньше (до .NET 10)

- Сложно с native / AOT
- Отдельные пакеты под платформы
- Ручной выбор RID
- Костыли в скриптах автоматизации

Как было раньше (до .NET 10)

- Сложно с native / AOT
- Отдельные пакеты под платформы
- Ручной выбор RID
- Костыли в скриптах автоматизации
- Отсутствие автокомплита

Новое в .NET 10

Platform-specific tools

Usecase: native / AOT tools 

Контекст

Usecase: native / AOT tools

Контекст

01

tools используют
native бинарники

Usecase: native / AOT tools

Контекст

01

tools используют
native бинарники

02

AOT / Go / Rust / C++

Usecase: native / AOT tools 

Контекст

01

tools используют
native бинарники

02

AOT / Go / Rust / C++

03

Обёртки
над платформенными
утилитами

Проблема

Проблема

tools ≠ обычные NuGet

Проблема

- tools ≠ обычные NuGet
- Нет поддержки runtimes/

Проблема

tools ≠ обычные NuGet

Нет поддержки runtimes/

Либо:

- всё в одном пакете + логика выбора
- либо разные tools под разные платформы

Проблема

tools ≠ обычные NuGet

Нет поддержки runtimes/

Либо:

- всё в одном пакете + логика выбора
- либо разные tools под разные платформы

Ручной RID

Проблема

- tools ≠ обычные NuGet
- Нет поддержки runtimes/
- Либо:
  - всё в одном пакете + логика выбора
  - либо разные tools под разные платформы
- Ручной RID
- Костыли в CI-скриптах

Решение: platform-specific tools


```
tool.nupkg
└─ runtimes/
   ├─ win-x64/
   ├─ linux-x64/
   └─ osx-arm64/
```

- Один tool-пакет → несколько RuntimeIdentifier
- Linux / macOS / Windows в одном NuGet
- CLI сам выбирает нужный бинарник
- Аналогично runtimes / в обычных NuGet-пакетах

```
<PropertyGroup>
  <RuntimeIdentifiers>
        linux-x64;
        linux-arm64;
        macos-arm64;
        win-x64;
        win-arm64;
        any
  </RuntimeIdentifiers>
</PropertyGroup>
```

Новое в .NET 10

One-shot execution

Usecase: CI / одноразовые задачи 

Контекст

Usecase: CI / одноразовые задачи 

Контекст

01

Нужно выполнить
tool один раз

Usecase: CI / одноразовые задачи 

Контекст

01

Нужно выполнить tool один раз

02

Чистые окружения

Проблема

Без local или global установку
dotnet tool не запустить

One-shot execution: dotnet tool exec

Запуск без установки

Просто выполнить и забыть.
Откуда берётся версия? Не из global и не из local:

- если версия не указана → берётся последняя доступная версия tool’a из NuGet
- если указана версия → используется она

Примеры:

- dotnet tool exec dotnet-ef migrations script
- dotnet tool exec dotnet-ef --version 9.0.1 migrations script

✔ Идеально для CI/CD и одноразовых задач

DNX-style execution

- Новый dnx script
- Упрощённый запуск dotnet tool exec

Пример команды:

- dnx dotnet-ef database update

Новое в .NET 10

CLI introspection

Usecase: сложный CLI и плохой UX

- В больших проектах dotnet и dotnet tools имеют много команд, подкоманд и опций
- Это сложно запомнить и неудобно использовать, особенно когда CLI активно используется в повседневной работе

Чем больше инструментов и возможностей, тем хуже становится UX обычного текстового CLI

Проблема

Проблема

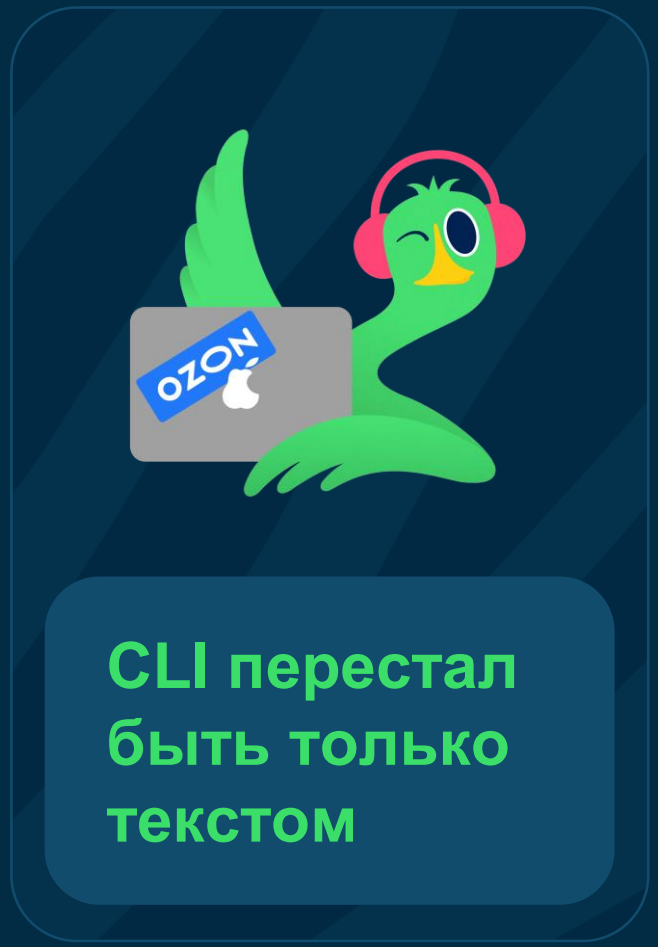
Исторически CLI в .NET был чисто текстовым интерфейсом

Проблема

Исторически CLI в .NET был чисто текстовым интерфейсом

Единственный источник правды это --help

Что нового в .NET 10: CLI introspection



- dotnet --cli-schema → JSON-описание команд
- Одинаково для: dotnet CLI и dotnet tools
- Основа для tooling вокруг CLI

Пример:

- dotnet --cli-schema

Зачем нужен cli-schema

Зачем нужен cli-schema

Автодокументация

Зачем нужен cli-schema

Автодокументация

Обёртки

Зачем нужен cli-schema

- Автодокументация
- Обёртки
- **IDE и tooling**

Зачем нужен cli-schema

- Автодокументация
- Обёртки
- IDE и tooling
- **Основа для UX**

Откуда берётся --cli-schema

```
→  CsharpTests git:(master) ✗ dotnet clean --cli-schema
{
  "name": "clean",
  "version": "10.0.101",
  "description": ".NET Clean Command",
  "hidden": false,
  "arguments": {
    "PROJECT | SOLUTION | FILE": {
      "description": "The project or solution or C# (file-based program) file to operate on. If a file
is not specified, the command will search the current directory for a project or solution.",
      "order": 0,
      "hidden": false,
      "valueType": "System.String[]",
      "hasDefaultValue": false,
      "arity": {
        "minimum": 0
      }
    }
  },
  ...
```

Native tab-completion

UX для CLI tools

- dotnet completions script [SHELL], где [SHELL] это bash, fish, nushell, pwsh, zsh
- Генерируется из cli-schema
- Подключается в shell startup (rc/profile)

Поддерживаемые оболочки:

- bash
- fish
- pwsh
- nushell
- zsh

Гайд от Microsoft, как включить Native tab-completion

.NET CLI развивает UX и кроссплатформенность, добавляя важные фичи

Подписывайтесь на мой канал

OZON Tech

Николай Кочубеев
nkochubeev@ozon.ru