

Вытягиваем монорепку



**Никита
Балихин**
Газпромбанк



**Михаил
Потапов**
Тинькофф

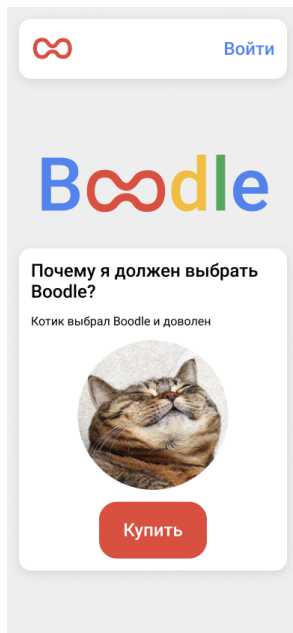
A wide-angle photograph of a flat, snow-covered landscape under a clear blue sky. The sun is bright and positioned on the right side of the frame, creating a lens flare. Two thin, dark poles or masts are visible in the distance, one on the left and one on the right. The horizon is low, showing some distant trees and structures.

THIS IS A TRUE STORY

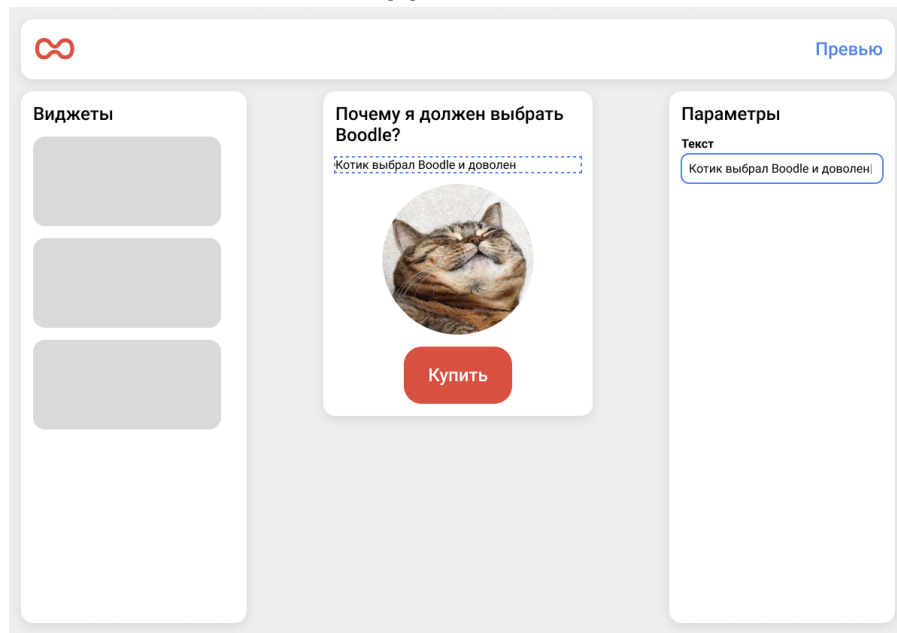
Boodle

Состав проекта

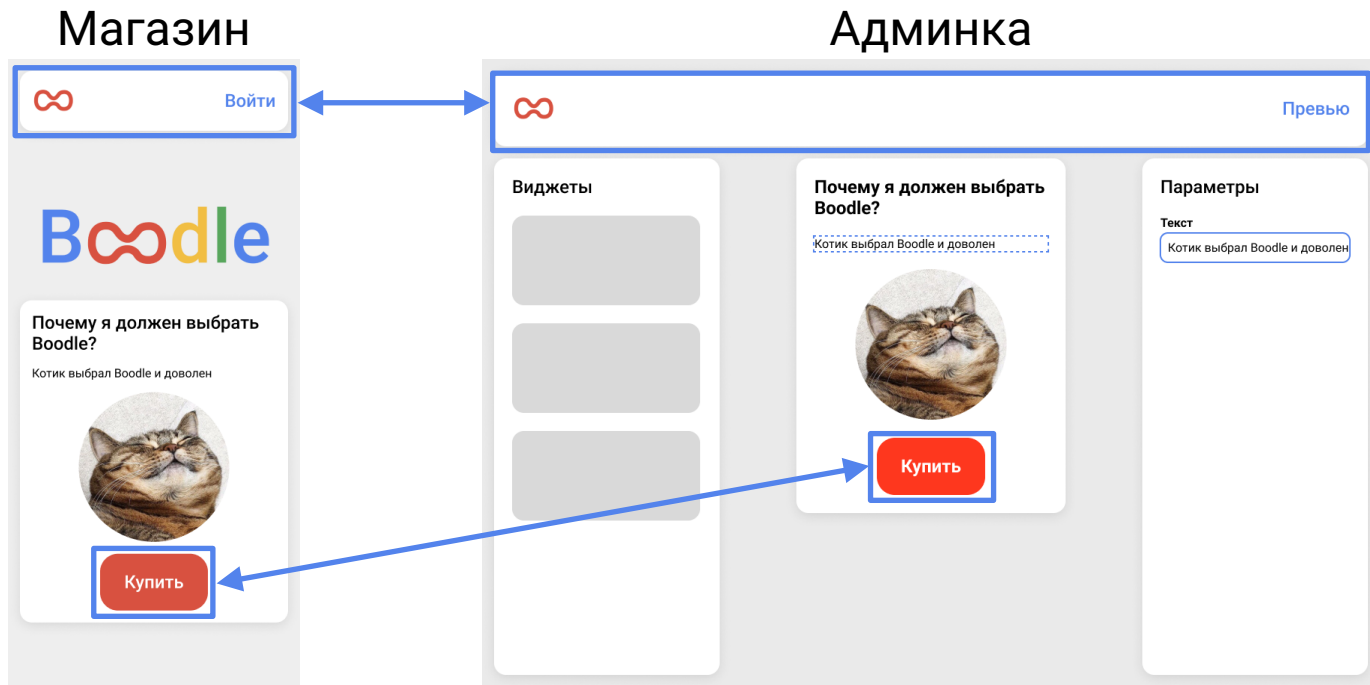
Магазин



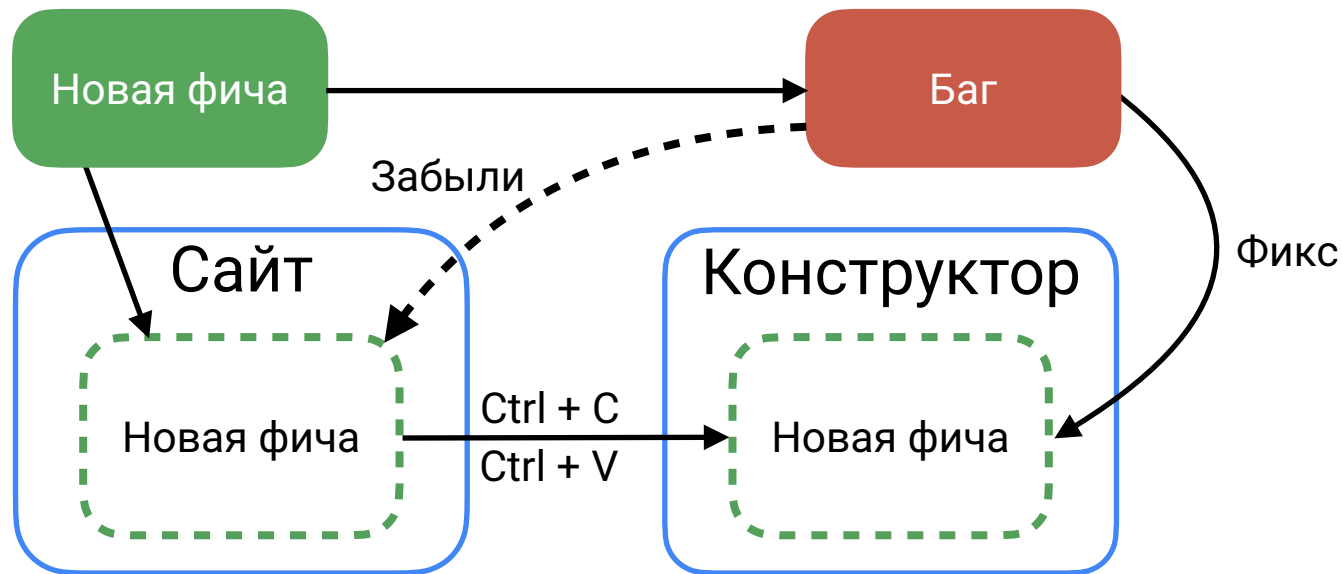
Админка



Рассинхрон в вёрстке

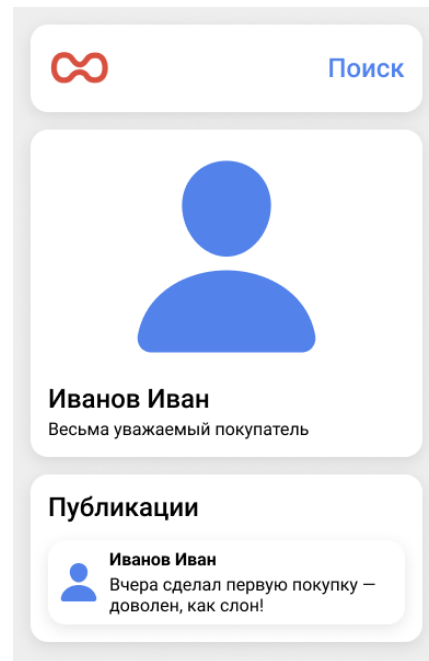
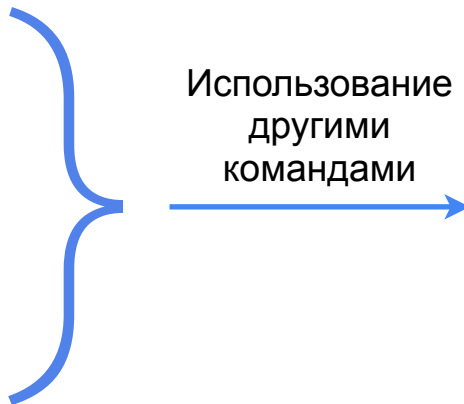


Дублирование кода



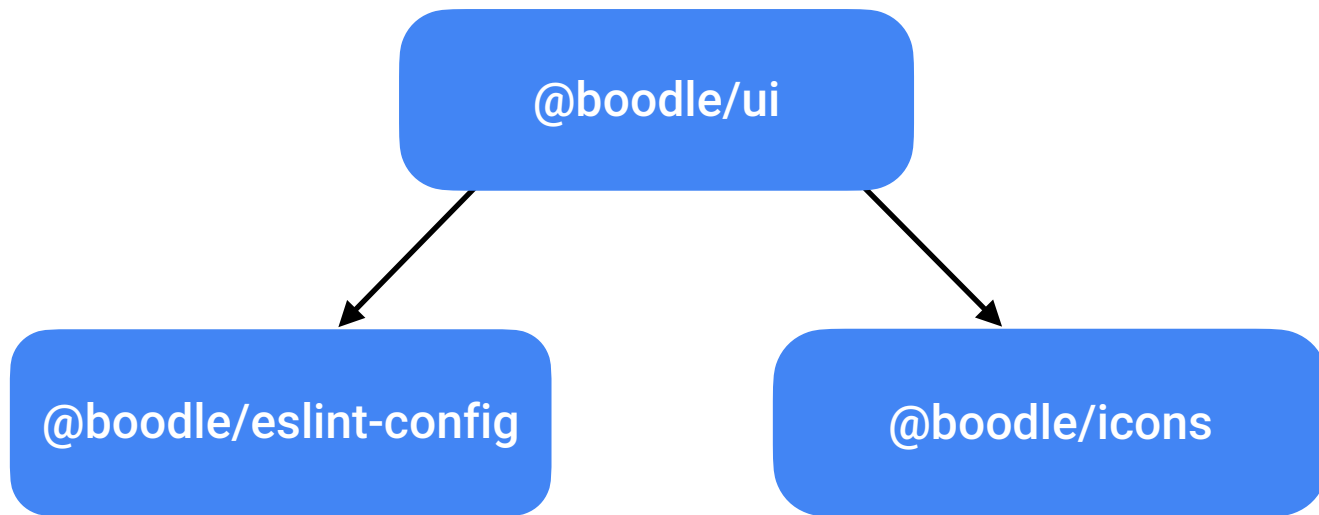
Первая задача

- Общие компоненты дизайн-системы
- Общий набор иконок
- Общий кодстайл

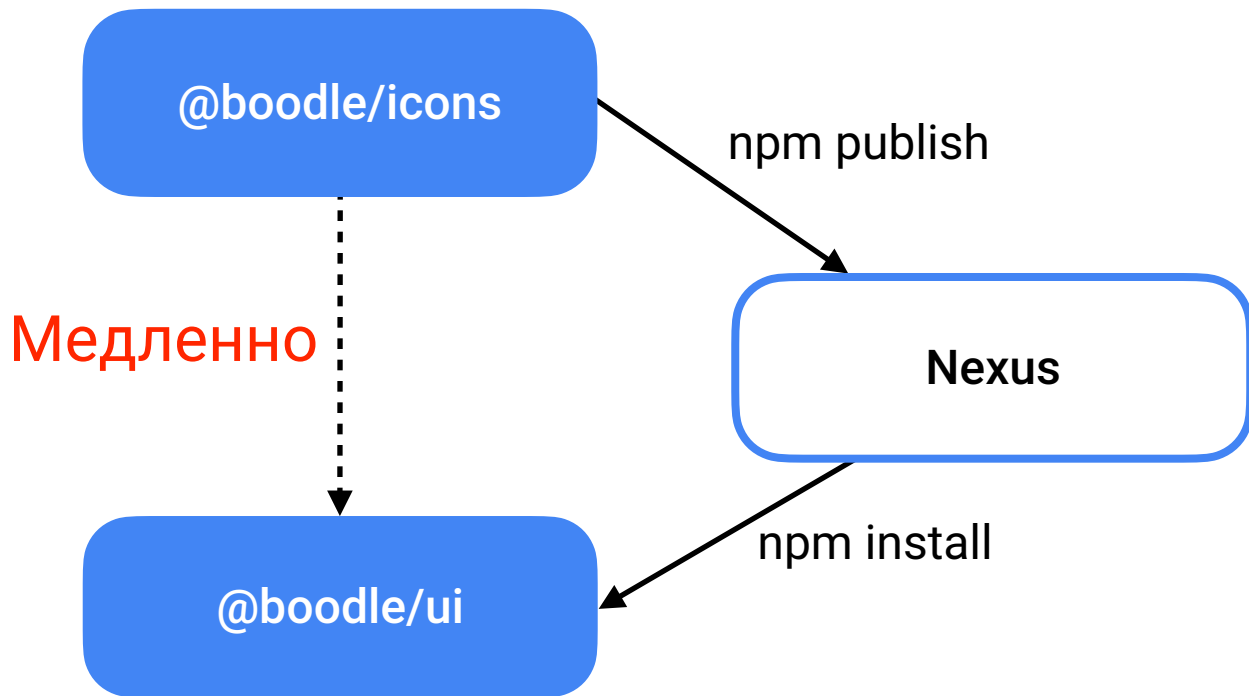




Платформа



Связываем пакеты



Монорепозиторий

@boodle/ui

npm install

@boodle/eslint-config

@boodle/icons

Организуем монорепозиторий

Указание путей до пакетов

```
1  {  
2    "name": "@boodle/ui",  
3    "version": "1.0.0",  
4    "main": "lib/index.js",  
5    "dependencies": {  
6      "@boodle/icons": "file:../icons"  
7    }  
8  }
```

Путь до
директории,
содержащей
package.json

npm install

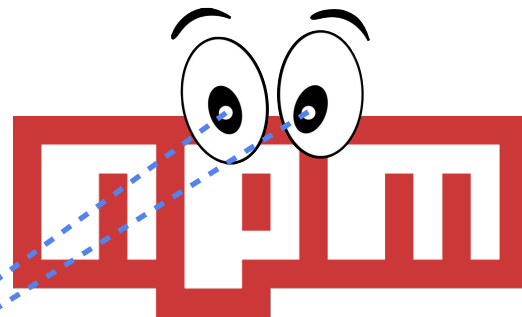
```
npm WARN reify invalid or damaged lockfile detected
npm WARN reify please re-try this operation once it completes
npm WARN reify so that the damage can be corrected, or perform
npm WARN reify a fresh install with no lockfile if the problem persists.
```

npm start

```
Error: Cannot find module '@boodle/icons'
```

```
1 {  
2   "name": "@boodle/ui",  
3   "version": "1.0.0",  
4   "main": "lib/index.js",  
5   "dependencies": {  
6     "@boodle/icons": "file:../icons"  
7   }  
8 }
```

Работает только в нашей монорепо



Это чё?
node_modules/icons?
Нет у меня такого

```
npm WARN reify invalid or damaged lockfile detected  
npm WARN reify please re-try this operation once it completes  
npm WARN reify so that the damage can be corrected, or perform  
npm WARN reify a fresh install with no lockfile if the problem persists.
```


Workspaces

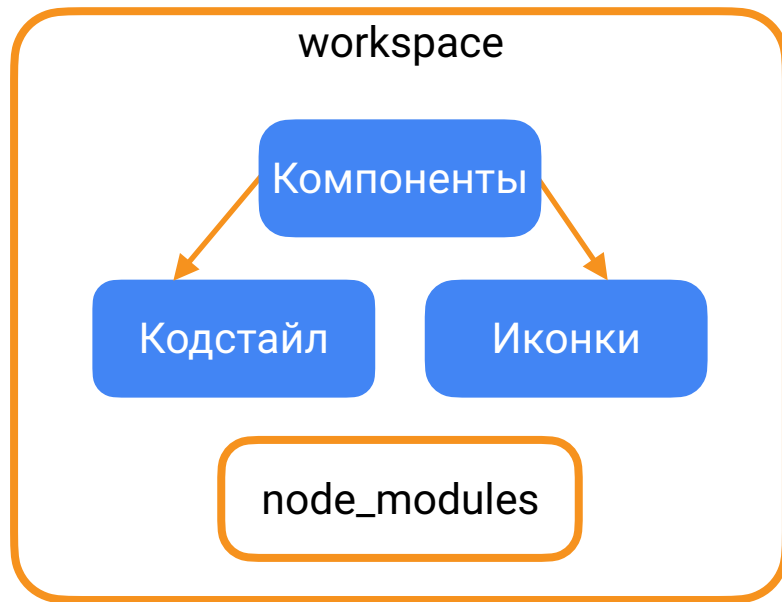
```
1 {
2   "name": "@boodle/ui",
3   "version": "1.0.0",
4   "dependencies": {
5     "@boodle/icons": "^1.0.0"
6   }
7 }
```



Работает у пользователей

```
1 {
2   "name": "npm-workspaces",
3   "workspaces": [
4     "ui-kit",
5     "icons",
6     "eslint-config"
7   ]
8 }
```

Workspaces

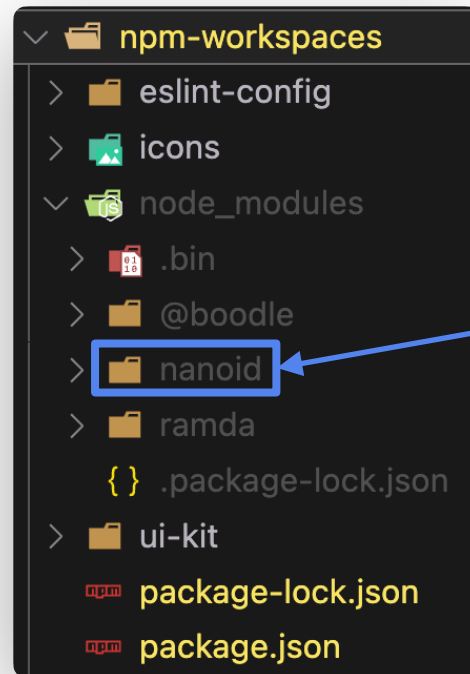
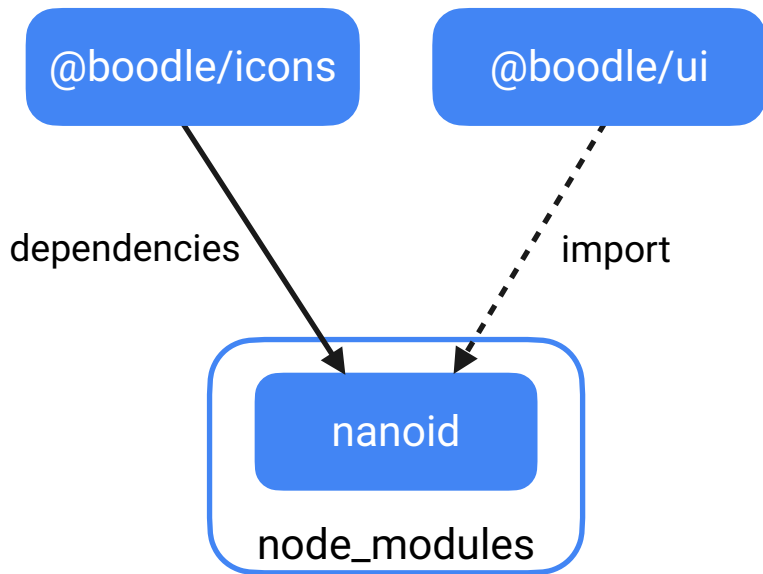


npm install

npm start

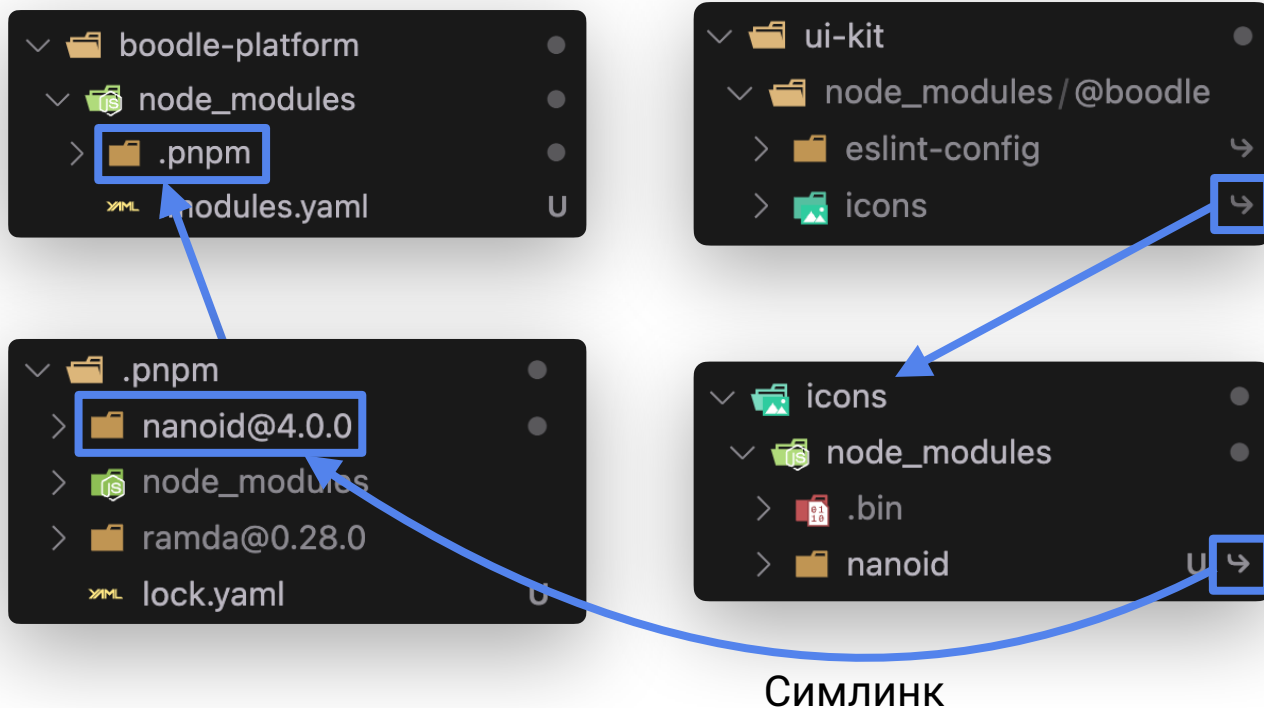
```
Error: Cannot find module 'nanoid'
```

Фантомные зависимости



Доступна всем
пакетам в
монорепе

PNPM



eslint-plugin-import

import/no-extraneous-dependencies

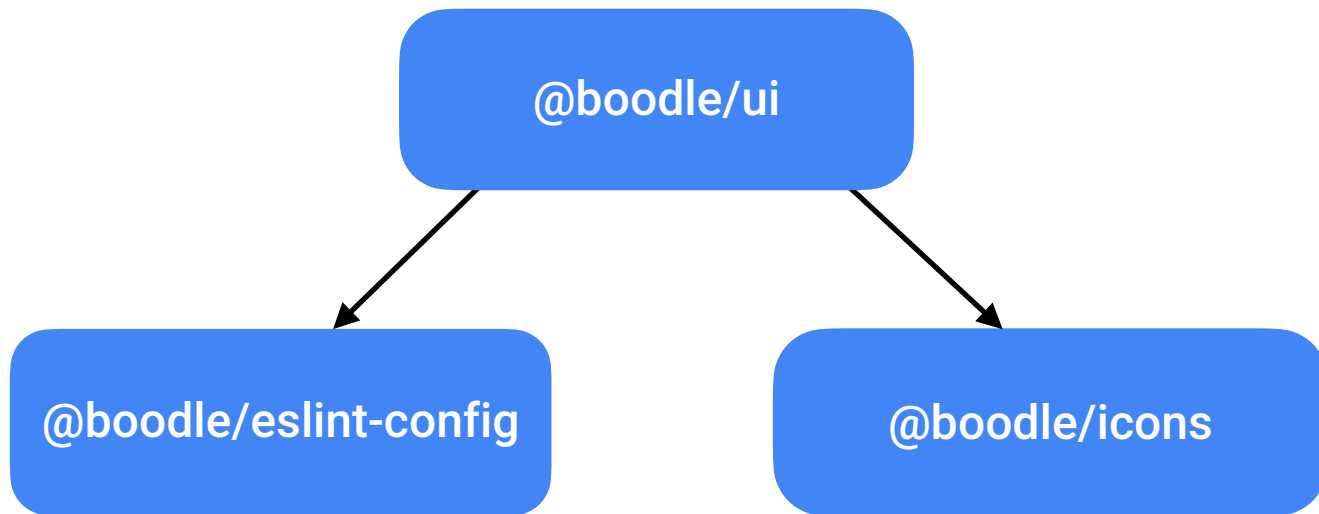
```
import { nanoid } from "nanoid";
```

```
module "/Users/landslide/Code/monorepo-tools/examples/npm-workspaces/node_modules/nanoid/index"
```

'nanoid' should be listed in the project's dependencies. Run 'npm i -S nanoid' to add it
eslint(import/no-extraneous-dependencies)

[View Problem](#) [Quick Fix... \(⌘.\)](#)

Оркестрация скриптов



Сборка каждого пакета — 10 секунд

npm run -ws

Сборка @boodle/ui
началась самой
первой

```
landslide@MacBook-Pro-Nikita npm-workspaces % time npm run -ws build  
  
> @boodle/ui@1.0.0 build  
> node build.js  
@boodle/ui build succeeded!  
  
> @boodle/icons@1.0.0 build  
> node build.js  
@boodle/icons build succeeded!  
  
> @boodle/eslint-config@1.0.0 build  
> node build.js  
@boodle/eslint-config build succeeded!  
npm run -ws build 0,37s user 0,12s system 1% cpu 30,512 total
```

~30 секунд

- ❌ Не учтён граф зависимостей
- ❌ Скрипты запущены последовательно

pnpm run -r

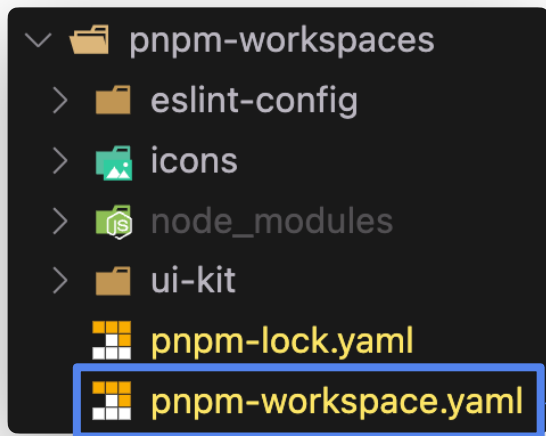
Сборка @boodle/ui
началась
последней

```
landslide@MacBook-Pro-Nikita pnpm-workspaces % time pnpm run -r build
Scope: all 3 workspace projects
eslint-config build$ node build.js
  @boodle/eslint-config build succeeded!
  Done in 10s
icons build$ node build.js
  @boodle/icons build succeeded!
  Done in 10s
ui-kit build$ node build.js
  @boodle/ui build succeeded!
  Done in 10s
pnpm run -r build 0,42s user 0,10s system 2% cpu 20,486 total
```

~20 секунд

- ✓ Пакеты собраны в правильном порядке
- ✓ Скрипты запущены по возможности параллельно

Переход на PNPM



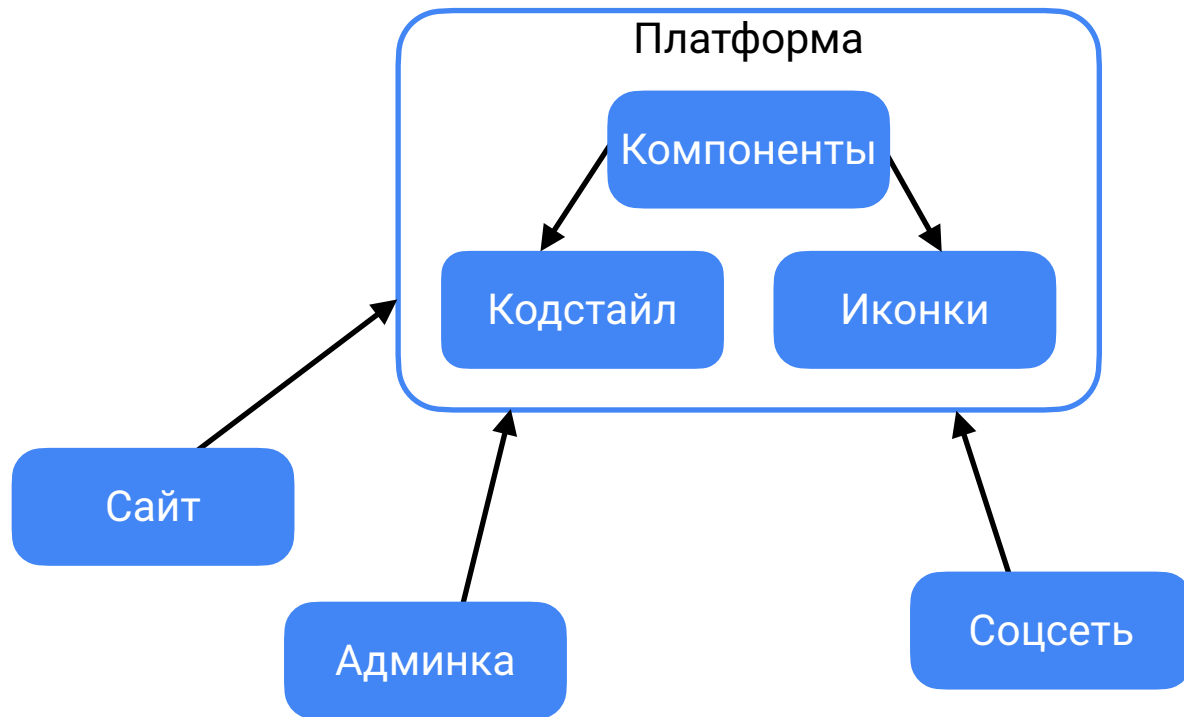
```
1 packages:
2   - eslint-config
3   - ui-kit
4   - icons
```

`npm install` ➡ `pnpm install`

`npm install --save package` ➡ `pnpm add package`

Что дал нам PNPM?

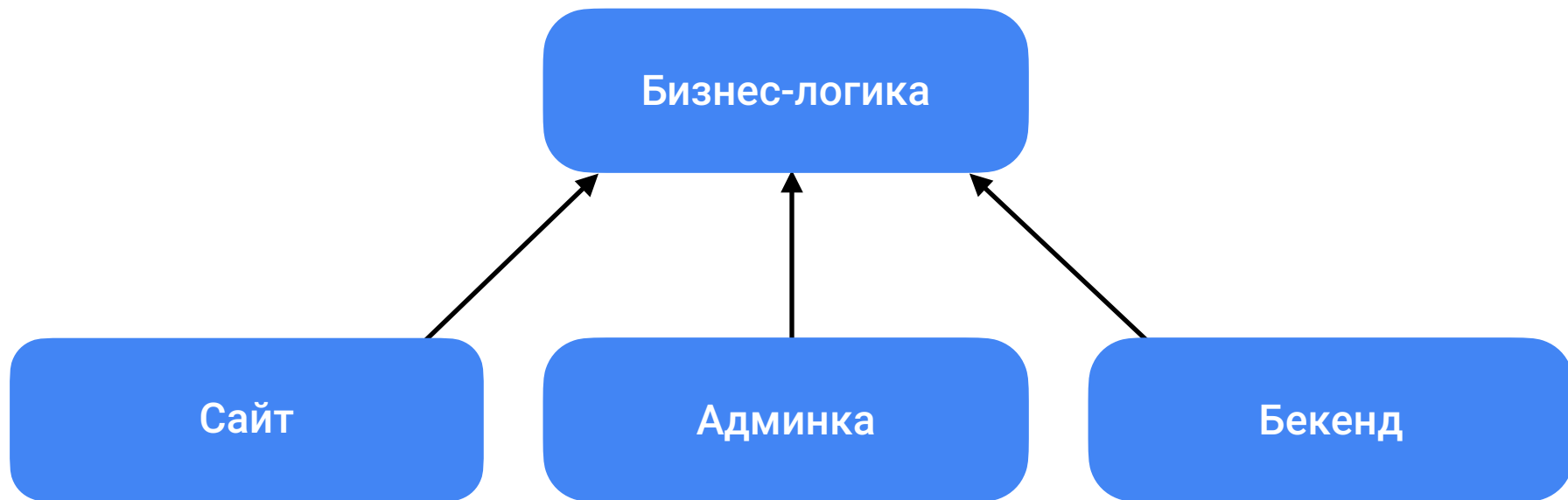
- Изоляция пакетов ~~eslint plugin import~~
- Оркестрация скриптов
- Быстрая установка зависимостей (быстрее, чем NPM)



Проблемы в продукте

- Дублирование кода в бизнес логике
- Разный стиль написания кода в приложениях
- Разные версии зависимостей
- Сложно внедрять фичи, затрагивающие несколько приложений

Монорепозиторий продукта



monorepo.tools

Everything you need to know about monorepos,
and the tools to build them.

- Made with love by Nrwl (the company behind Nx)



**Smart, Fast
and Extensible
Build System**

Стили монорепозитория от NX

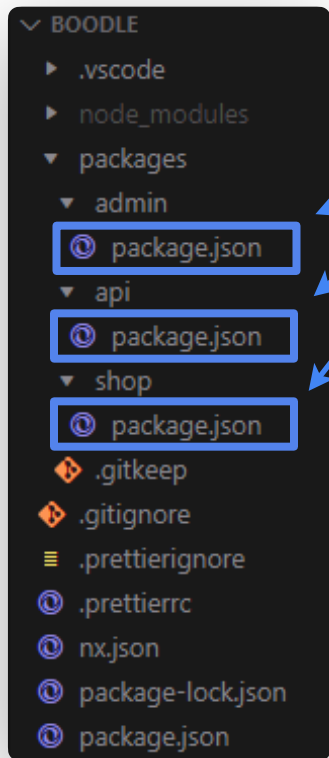
Package Based Repository



Integrated Repository



Package Based Repo



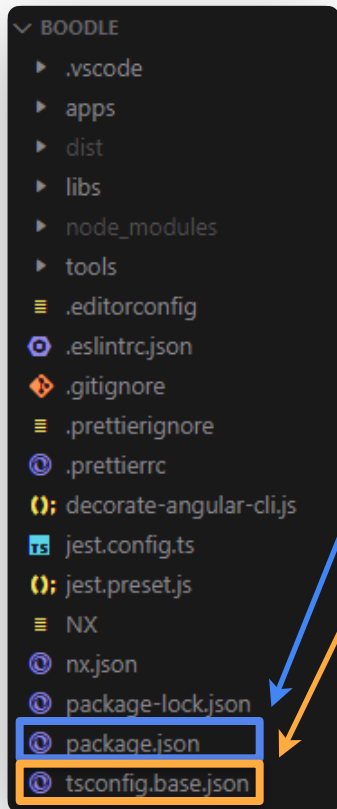
- Акцент на разработку библиотек и публикацию
- У каждого пакета свой package.json и версии одних и тех же зависимостей могут отличаться
- Связь между пакетами

```
1 "workspaces": [  
2   "packages/*"  
3 ]
```

package.json

- Nx - дополнительный пакет, для оркестрации, кеширования

Integrated Repo



- Акцент на разработку приложений
- Приложения и библиотеки используют общий package.json и другие конфиги
- Связь между пакетами

```
1 "paths": {
2   "@boodle/models": ["libs/models/src/index.ts"],
3   "@boodle/products": ["libs/products/src/index.ts"]
4 }
```

tsconfig.base.json

- Nx задает структуру монорепозитория, предусматривает и настраивает некоторые инструменты

Создание NX workspace

`npx create-nx-workspace boodle`



Создание монорепо

`npm install -D @nrwl/nest`

`npm install -D @nrwl/angular`



Установка плагинов

`npx nx generate @nrwl/nest:app api`

`npx nx generate @nrwl/angular:library models`



Генерация
бойлерплейта

Интерфейс командной строки

`npx nx run api`



Запустить в режиме разработки

`nx run-many --target=lint --projects=shop,admin,api`



Запустить линтер,
на нескольких
приложениях

`nx run-many --target=test --all --parallel=4`



Протестировать,
все параллельно

Интеграция с IDE



или



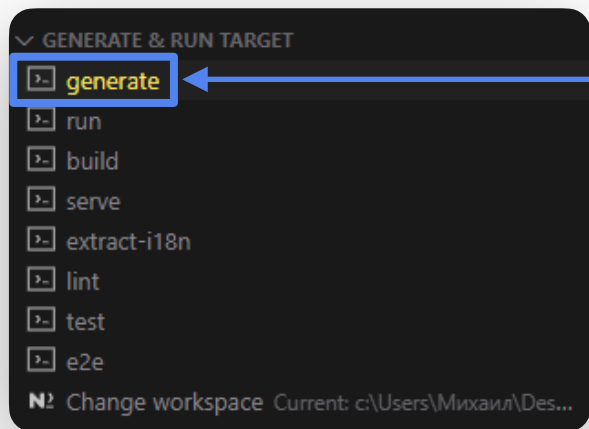
+

N>_ CONSOLE

=

DX

Все можно сделать одним щелчком мыши

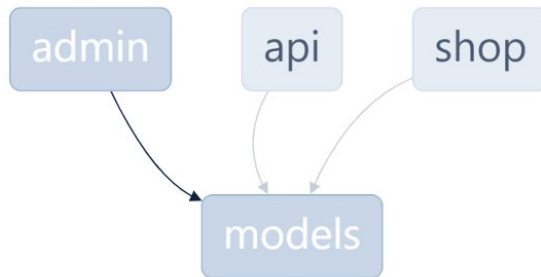
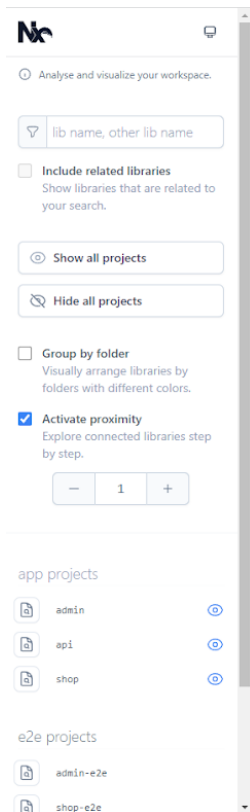


A screenshot of the 'nx generate @nestjs/schematics:library' command interface. The interface shows the following fields and options:

- name ***: The name of the library. models --prefix=models
- prefix ***: The prefix of the library. models
- language**: Nest library language.
- path**: The path to create the library.
- rootDir**: The libraries root directory.

Визуализация монорепозитария

`npx nx graph`



Задача на бекенде

Переделать авторизацию на бекенде

Что то пошло не так :(

```
chunk (runtime: main) main.js (main) 4.6 KiB [entry] [rendered]
webpack compiled successfully (798d9dfa25f77c93)
Debugger listening on ws://localhost:9229/6df1c716-2b6f-41fd-b6a7-d35da6f3c49a
Debugger listening on ws://localhost:9229/6df1c716-2b6f-41fd-b6a7-d35da6f3c49a
For help, see: https://nodejs.org/en/docs/inspector
Type-checking in progress...
Error [ERR_REQUIRE_ESM]: Must use import to load ES Module: C:\Users\Михаил\Desktop\holly-nx\boodle\node_modules\@angular\core\fesm2015\core.mjs
    at Module.load (internal/modules/cjs/loader.js:935:11)
    at Function.Module._load (internal/modules/cjs/loader.js:778:12)
    at Function.Module.load (C:\Users\Михаил\Desktop\holly-nx\boodle\packages\js\src\executors\node\node-with-require-overrides.ts:16:27)
    at Module.require (internal/modules/cjs/loader.js:961:19)
    at require (internal/modules/cjs/helpers.js:92:18)
    at Object.<anonymous> (C:\Users\Михаил\Desktop\holly-nx\boodle\dist\apps\api\webpack:boodle\external commonjs "@angular\core":1:1)
    at __webpack_require__ (C:\Users\Михаил\Desktop\holly-nx\boodle\dist\apps\api\webpack:boodle\webpack\bootstrap:19:1)
    at Object../apps/api/src/app/auth/auth.service.ts (C:\Users\Михаил\Desktop\holly-nx\boodle\dist\apps\api\webpack:boodle\apps\api\src\app\auth\auth.service.ts:1:1)
    at __webpack_require__ (C:\Users\Михаил\Desktop\holly-nx\boodle\dist\apps\api\webpack:boodle\webpack\bootstrap:19:1)
    at Object../apps/api/src/app/auth/auth.module.ts (C:\Users\Михаил\Desktop\holly-nx\boodle\dist\apps\api\webpack:boodle\apps\api\src\app\auth\auth.module.ts:3:1)
No errors found.
```

Ошибка в импортах

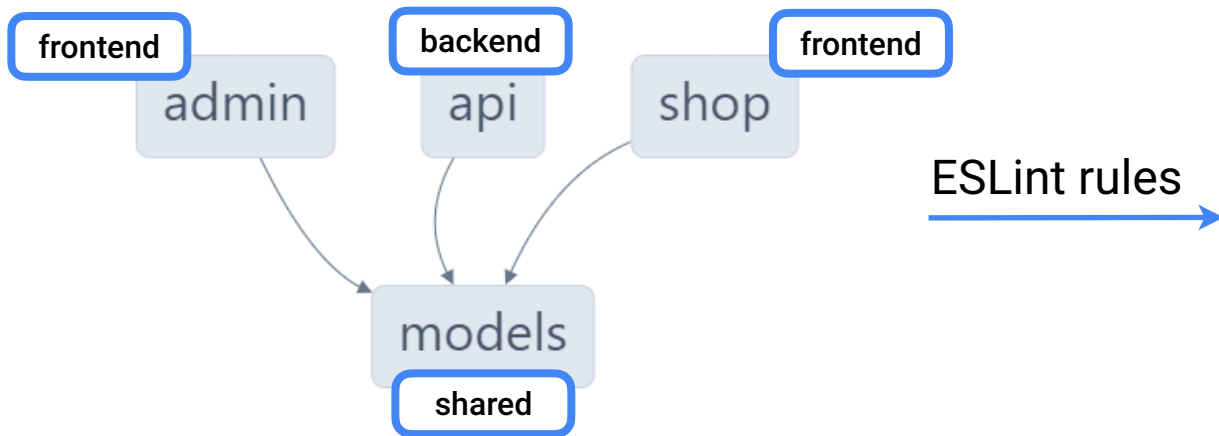
```
1 import { Injectable } from '@angular/core'
2 import { AuthRepository } from '../auth.repository';
3
4 @Injectable()
5 export class AuthService {
6   constructor(private readonly authRepository: AuthRepository) {}
```



Angular на
бекенде?

```
1 import { Injectable } from '@nestjs/common';
2 import { AuthRepository } from '../auth.repository';
3
4 @Injectable()
5 export class AuthService {
6   constructor(private readonly authRepository: AuthRepository) {}
```

Тегирование приложений и библиотек



Black list of imports

Не может содержать импорт из @angular/*

```
1 "tags": ["backend"]
```

api/project.json

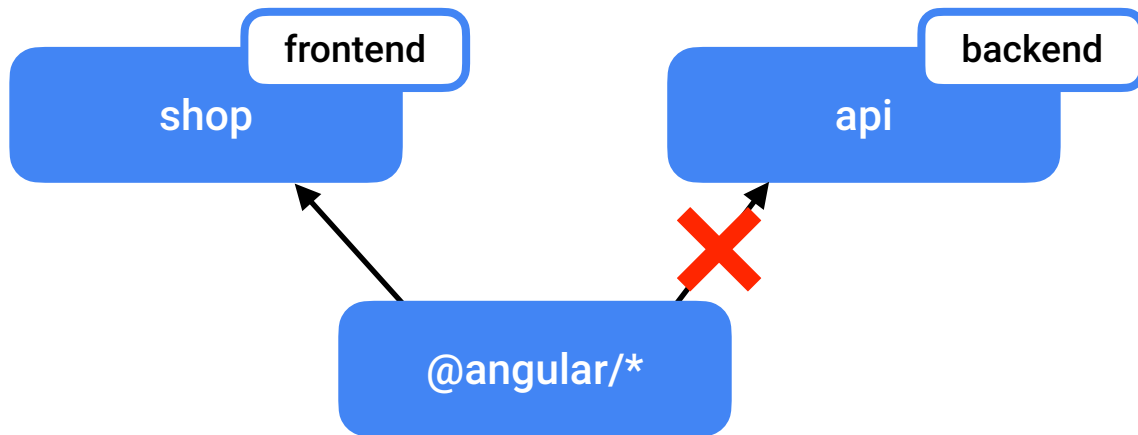
```
1 "tags": ["frontend"]
```

shop/project.json

```
1 "rules": {  
2   "@nrwl/nx/enforce-module-boundaries": [  
3     "error",  
4     {  
5       "enforceBuildableLibDependency": true,  
6       "allow": [],  
7       "depConstraints": [  
8         {  
9           "sourceTag": "backend",  
10          "bannedExternalImports": ["@angular/*"]  
11        },  
12        {  
13          "sourceTag": "frontend",  
14          "bannedExternalImports": ["@nestjs/*"]  
15        }  
16      ]  
17    }  
18  ]  
19 }
```

.eslintrc.json

Ограничения по использованию библиотек



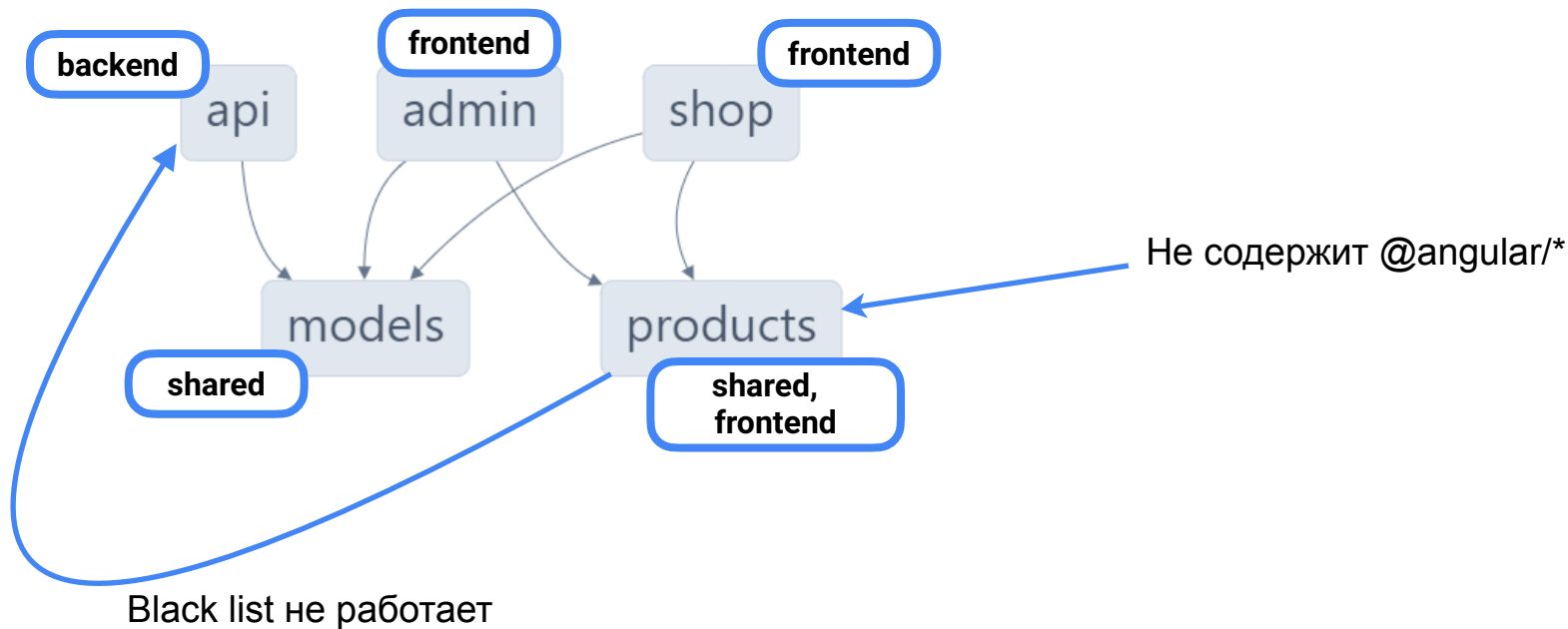
```
import { Injectable } from '@angular/core';
```

A project tagged with "backend" is not allowed to import the "@angular/core" package eslint([@nrwl/nx/enforce-module-boundaries](#))

[View Problem](#) [Quick Fix...](#) (Ctrl+.)

```
constructor(private readonly authRepository: AuthRepository) {}
```

Зависимость только от определенных тегов



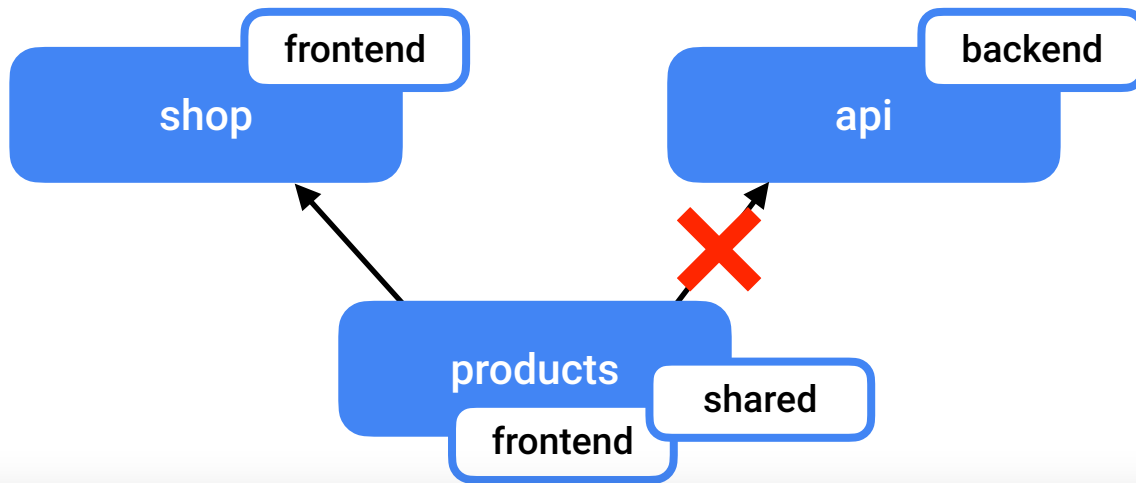
White list of imports

```
1  "rules": {
2    "@nrwl/nx/enforce-module-boundaries": [
3      "error",
4      {
5        "enforceBuildableLibDependency": true,
6        "allow": [],
7        "depConstraints": [
8          {
9            "sourceTag": "backend",
10           "bannedExternalImports":["@angular/*"],
11           "onlyDependOnLibsWithTags": ["backend", "shared"]
12         },
13         {
14           "sourceTag": "frontend",
15           "bannedExternalImports":["@nestjs/*"],
16           "onlyDependOnLibsWithTags": ["frontend", "shared"]
17         }
18       ]
19     }
20   ]
21 }
```

Можно импортировать
библиотеки только этих
типов

.eslinttrc.json

Ограничения по использованию тегов




```
import { ProductsModule } from '@booodle/products';
```

```
(alias) class ProductsModule  
import ProductsModule
```

A project tagged with "backend" can only depend on libs tagged with "backed", "shared" `eslint(@nrwl/nx/enforce-module-boundaries)`

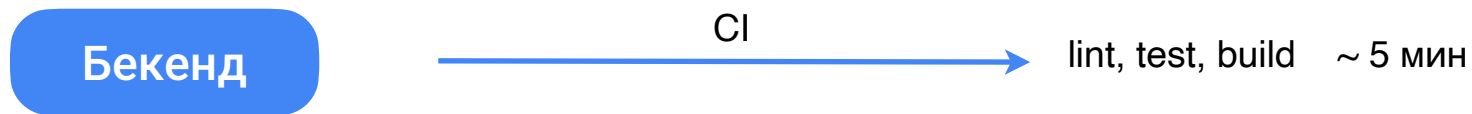
[View Problem](#) [Quick Fix... \(Ctrl+.\)](#)

CI/CD стал дольше

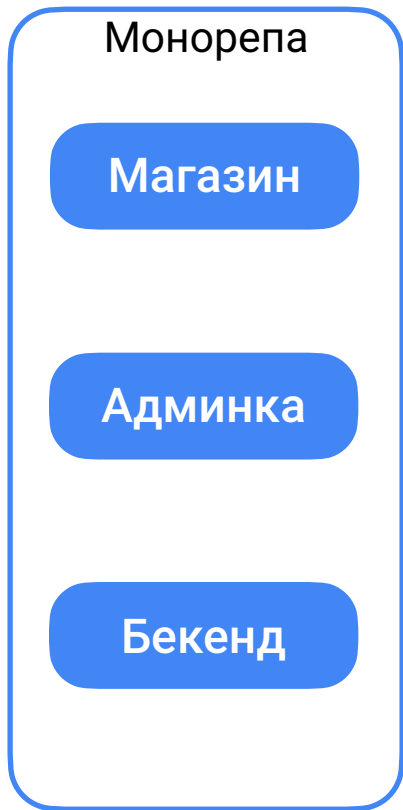
Status	Pipeline	Triggerer	Stages
<div>✓ passed</div> <div>🕒 00:09:21</div> <div>🕒 10 minutes ago</div>	<div>New auth</div> <div>#682800123 🔗 5 🔑 614e516c 🌐</div> <div>latest merge request</div>		<div>✓ ✓ ✓ ✓</div>

Раньше pr с фичей на беке выполнялся за 5 мин

CI/CD стал дольше



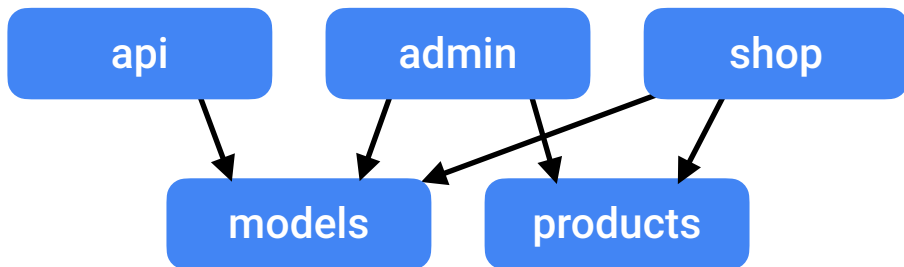
CI/CD стал дольше



Тестируем только изменения

~~npx nx run many target=build all~~


Соберётся всё



npx nx affected --target=build --base=master --head=HEAD

Разница с master веткой

CI/CD стал быстрее

Status	Pipeline	Triggerer	Stages
<div>✓ passed</div> <div>🕒 00:05:48</div> <div>📅 6 minutes ago</div>	<div>New auth</div> <div>#682800176 🔑 5 🔗 434e524d 🌐</div> <div>latest merge request</div>		<div>✓ ✓ ✓ ✓</div>

Раньше pr с фичей на беке выполнялся за ~10 мин

Кэширование

```
$ npx nx run-many --target=build --all
```

```
> NX Running target build for 3 projects
```

```
✓ nx run api:build (7s)
```

```
✓ nx run shop:build:production (19s)
```

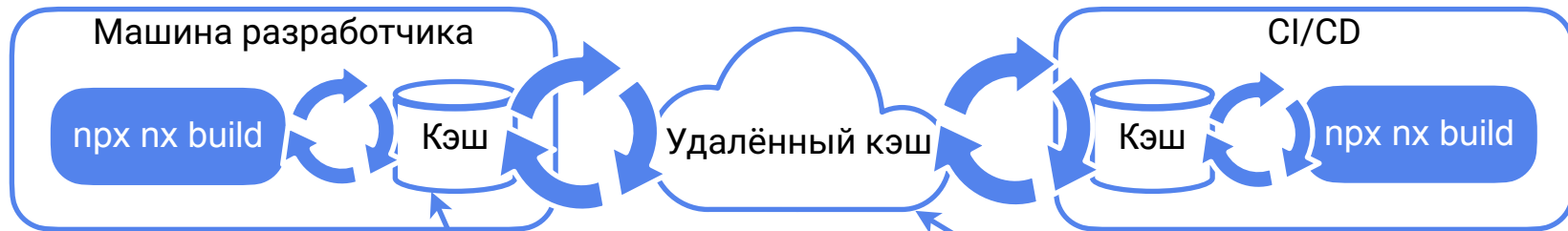
```
✓ nx run admin:build:production (20s)
```

```
-----  
> NX Successfully ran target build for 3 projects (20s)
```

Собрали все приложения



Кэширование



```
$ npx nx run-many --target=build --all
✓ nx run admin:build:production [existing outputs match the cache, left as is]
✓ nx run api:build [existing outputs match the cache, left as is]
✓ nx run shop:build:production [existing outputs match the cache, left as is]






-----
> NX Successfully ran target build for 3 projects (96ms)
```

```
$ npx nx run-many --target=build --all
✓ nx run admin:build:production [remote cache]
✓ nx run api:build [remote cache]
✓ nx run shop:build:production [remote cache]

-----
> NX Successfully ran target build for 3 projects (3s)

Nx read the output from the cache instead of running the command for 3 out of 3 tasks.
Nx Cloud made it possible to reuse 3 tasks: https://nx.app/runs/reOHIIS2g1
```

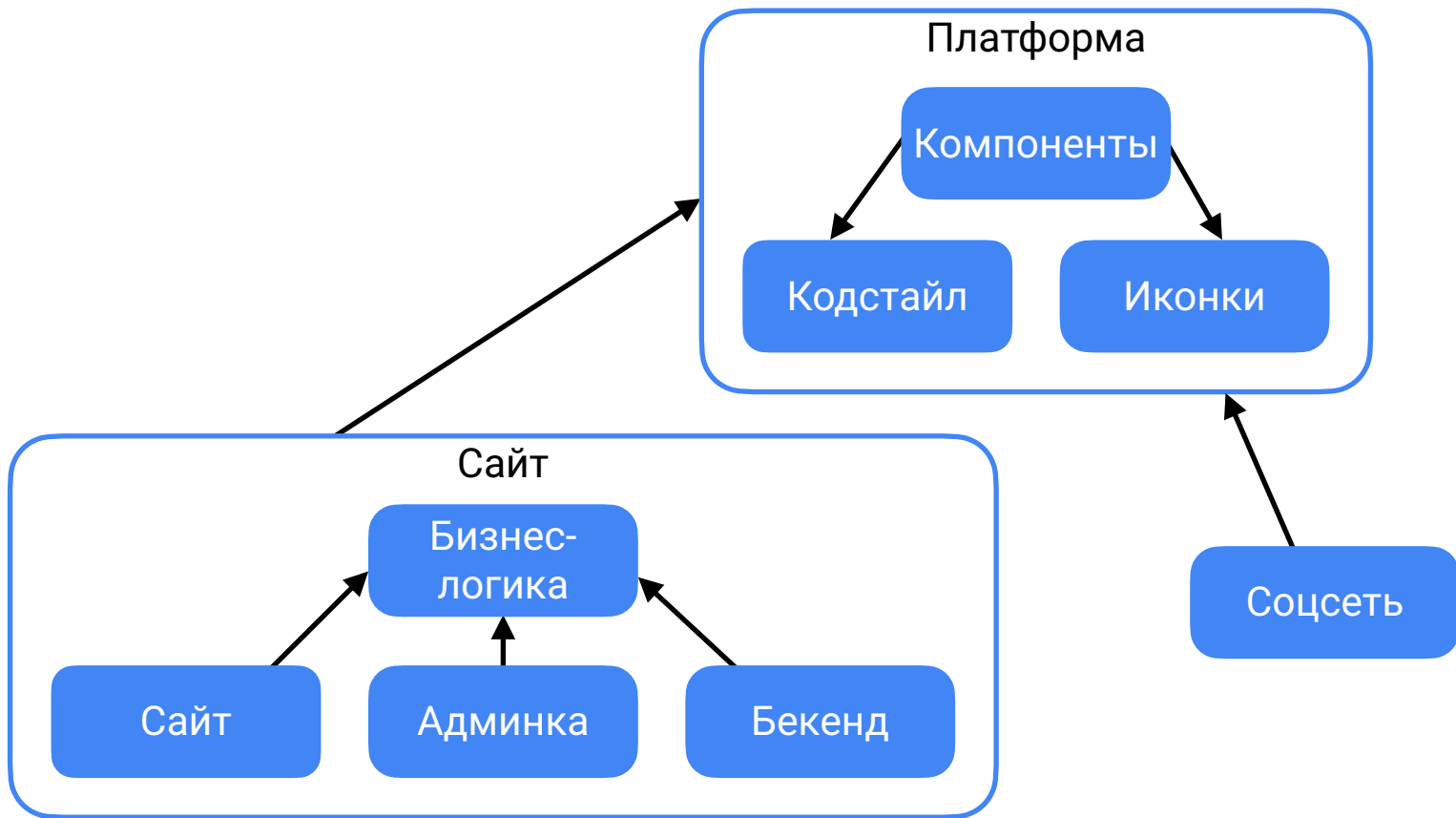

CI/CD стал еще быстрее

Status	Pipeline	Triggerer	Stages
<div>passed</div> <div>00:01:49</div> <div>5 minutes ago</div>	<div>Update readme</div> <div>#682800180 5 641d12a89</div> <div>latest merge request</div>		<div></div>

Раньше pr с фичей на беке выполнялся за ~6 мин

Что дал нам NX?

- Единые версии зависимостей
- Генерация бойлерплейта
- Оркестрация скриптов
- Инструменты для CI в монорепозитории





Публикация

- 1) Понять, какие пакеты нужно опубликовать
- 2) В каждом пакете инкрементировать версию и обновить ченджлог
- 3) Опубликовать каждый пакет

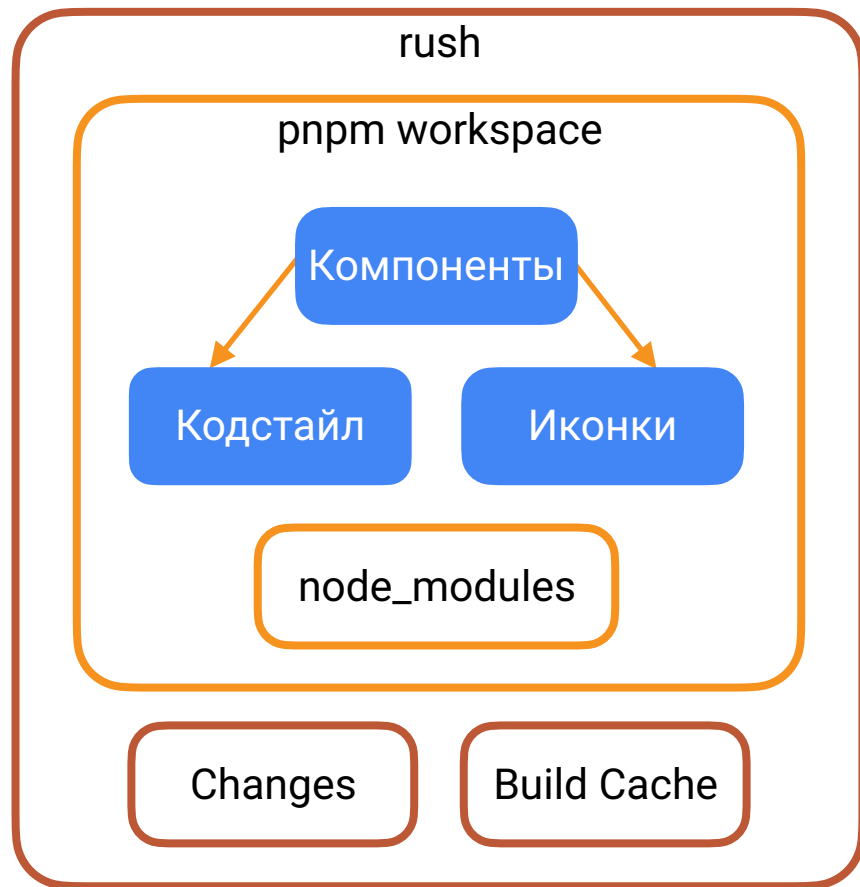


Этот процесс можно автоматизировать



rush

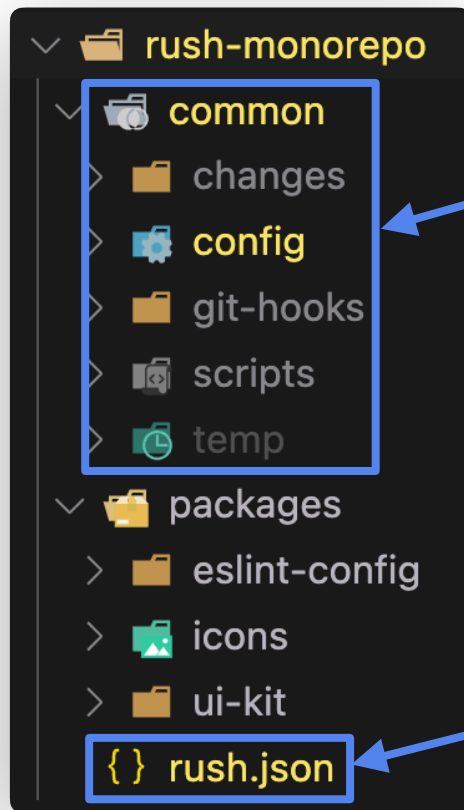
Концепция



Инициализация

1) `npm i -g @microsoft/rush`

2) `rush init`



Сгенерировал Rush

Основная
конфигурация
монорепы

Конфигурация

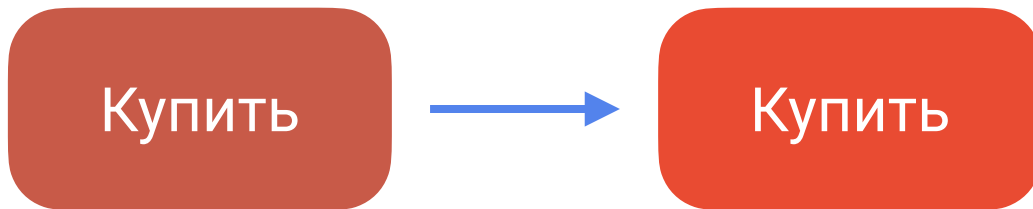
rush.json

```
1  "projects": [  
2    {  
3      "packageName": "@boodle/ui",  
4      "projectFolder": "packages/ui-kit",  
5      "shouldPublish": true  
6    },  
7    {  
8      "packageName": "@boodle/icons",  
9      "projectFolder": "packages/icons",  
10     "shouldPublish": true  
11   },  
12   {  
13     "packageName": "@boodle/eslint-config",  
14     "projectFolder": "packages/eslint-config",  
15     "shouldPublish": true  
16   }  
17 ]
```

Процесс разработки

Задача

Изменить цвет кнопки на более продающий



Установка зависимостей

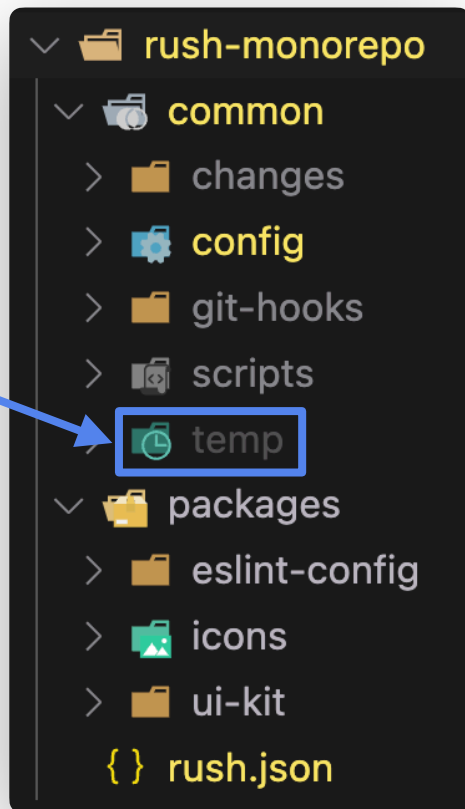
`pnpm install` 🙋 `rush install`

```
Installation is already up-to-date.
```

```
Rush install finished successfully. (0.06 seconds)
```

Временные файлы

Локальные файлы, не
индексируемые гитом
(в том числе node_modules)



Сборка зависимостей

```
rush build --to-except @boodle/ui
```

```
rush build -T .
```

```
==[ SKIPPED: 2 projects ]=====
```

These projects were already up to date:

@boodle/eslint-config

@boodle/icons

```
rush build (0.09 seconds)
```

```
1  .button {  
2      background-color: #C85A48  
3  }
```



```
1  .button {  
2      background-color: #EA4B32  
3  }
```

```
git commit -am "Changed button color"
```

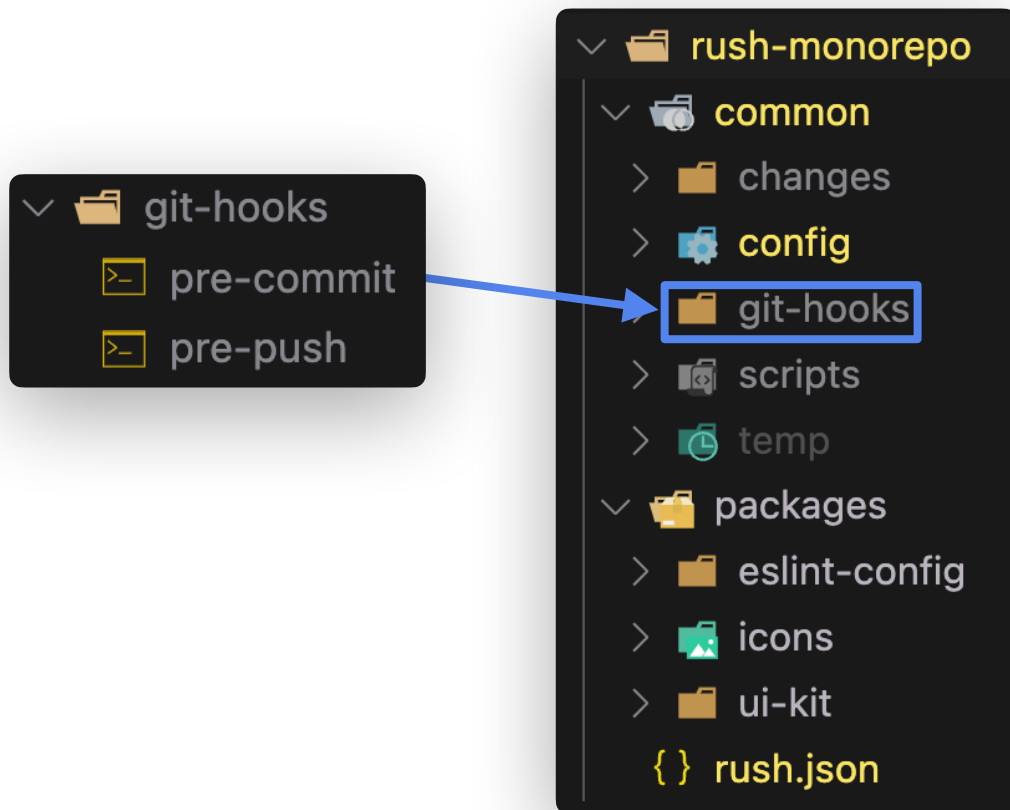
pre-commit →
lint-staged

```
==[ SUCCESS: 1 project ]=====
These projects completed successfully:
  @boodle/ui                      0.44 seconds

==[ SKIPPED: 2 projects ]=====
These projects were already up to date:
  @boodle/eslint-config
  @boodle/icons

rush lint-staged (0.49 seconds)
```


Git-хуки



Тестирование изменений

```
rush test --from git:origin/master
```



Запустить тесты у изменённых
пакетов и у зависимых от них
пакетов

```
git push
```

```
ERROR: The following projects have been changed and require change descriptions,  
but change descriptions were not detected for  
them:
```

```
- @boodle/ui
```

```
To resolve this error, run "rush change." This will generate change description  
files that must be committed to source control.
```



pre-push

rush change --verify

Документирование изменений

rush change

Starting "rush change"

The target branch is origin/develop
Checking for updates to origin/develop...

@boodle/ui

? Describe changes, or ENTER if no changes: Цвет кнопки заменён на более продающий

? Select the type of change: (Use arrow keys)

major – for changes that break compatibility, e.g. removing an API

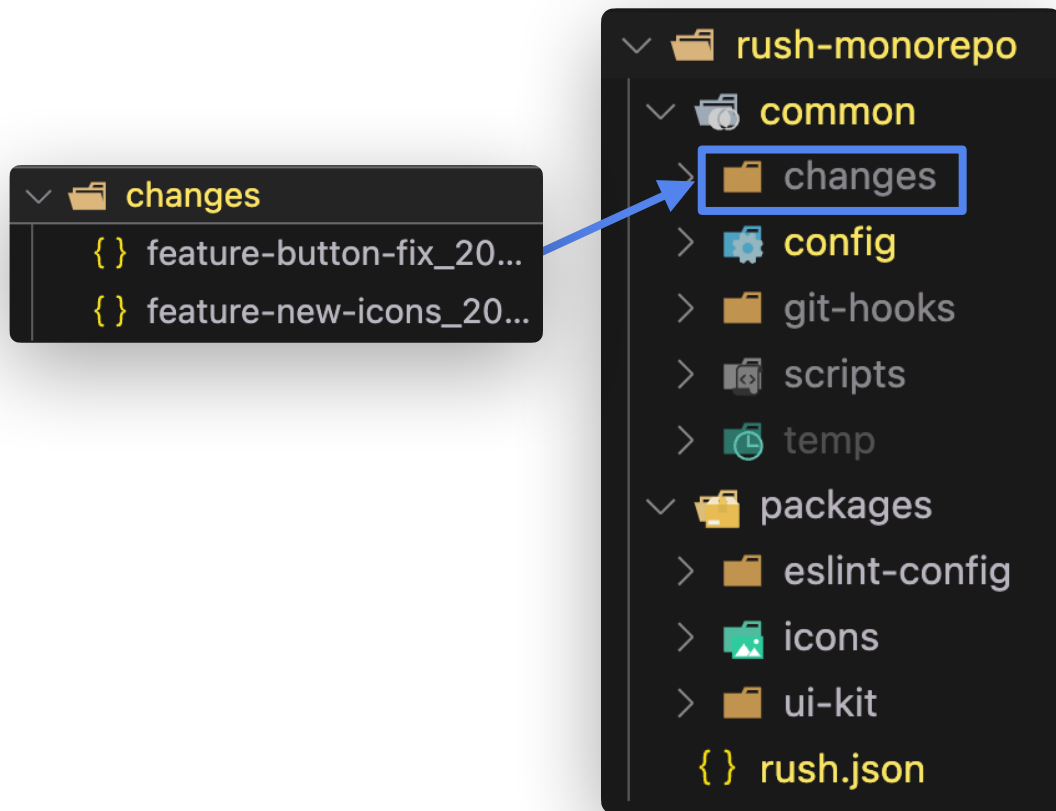
minor – for backwards compatible changes, e.g. adding a new API

> patch – for changes that do not affect compatibility, e.g. fixing a bug

none – for changes that do not need an immediate release, e.g. eslint config change

```
1 {  
2   "changes": [  
3     {  
4       "packageName": "@boodle/ui",  
5       "comment": "Цвет кнопки заменён на более продающий",  
6       "type": "patch"  
7     }  
8   ],  
9   "packageName": "@boodle/ui"  
10 }
```

Хранение изменений



Инкрементирование версий

```
rush version --bump
```

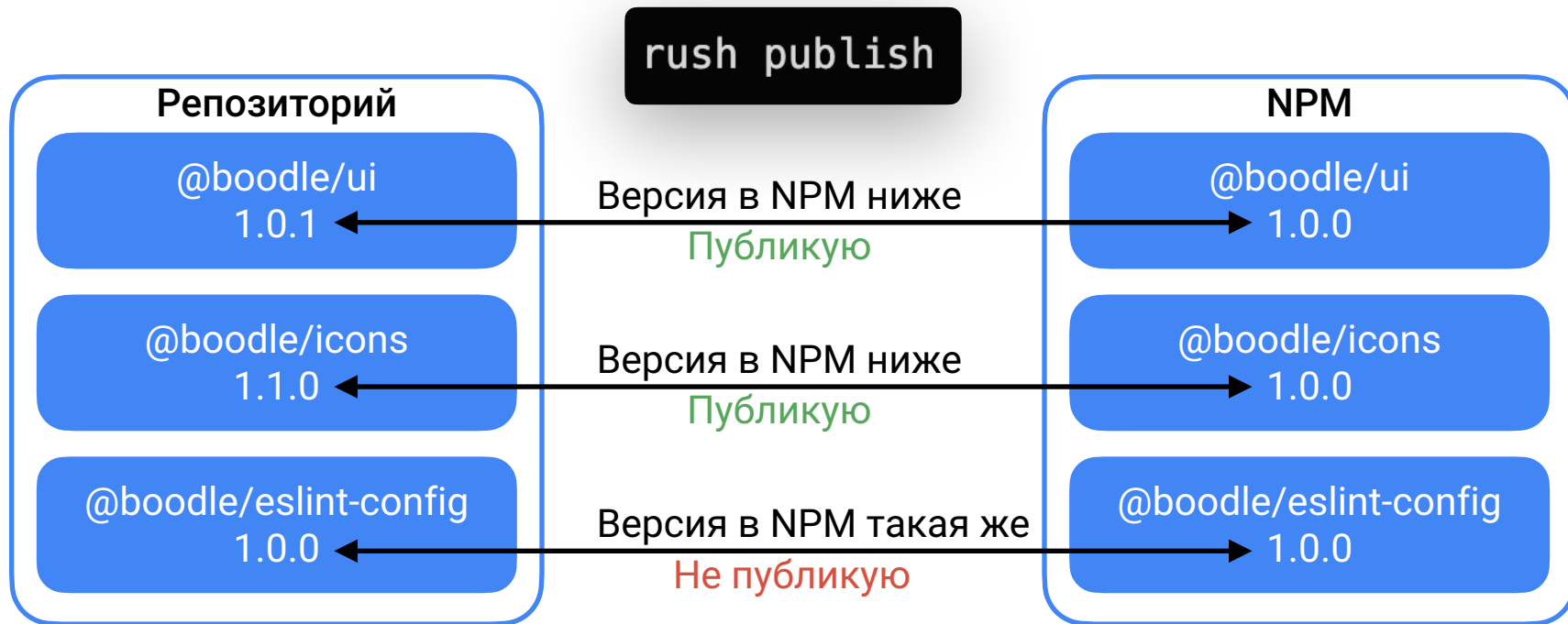
```
1 {  
2   "changes": [  
3     {  
4       "packageName": "@boodle/ui",  
5       "comment": "Цвет кнопки заменён на более продающий",  
6       "type": "patch",  
7     }  
8   ],  
9   "packageName": "@boodle/ui"  
10 }
```

@boodle/ui

1.0.0 → 1.0.1

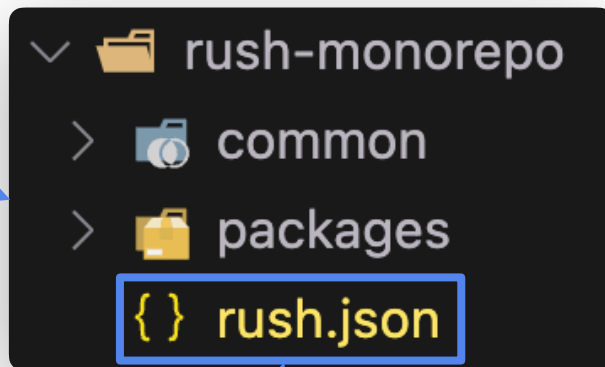
```
1 # Change Log - @boodle/ui  
2  
3 ## 1.0.1  
4  
5 Wed, 2 Nov 2022 11:35:56 GMT  
6  
7 ### Patches  
8  
9 - Цвет кнопки заменён на более продающий
```

Публикация



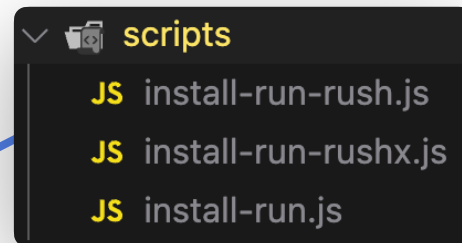
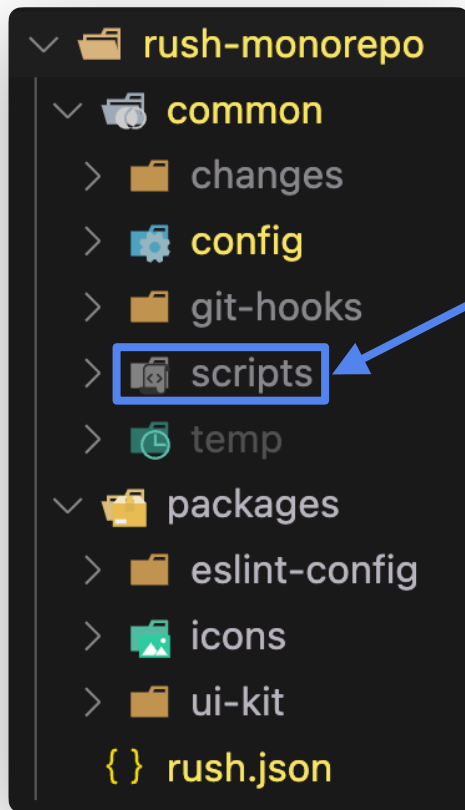
Воспроизводимость

Нет package.json



```
1 "nodeSupportedVersionRange": ">=14.15.0 <15.0.0 || >=16.13.0 <17.0.0",  
2 "rushVersion": "5.62.4",  
3 "pnpmVersion": "6.7.1",
```


CI/CD



Скрипты,
устанавливающие и
запускающие Rush

CI/CD

```
1  # Устанавливаем зависимости
2  install:
3    script:
4      - node common/scripts/install-run-rush.js install
5
6  # Собираем
7  build:
8    script:
9      - node common/scripts/install-run-rush.js build
10
11 # Публикуем
12 publish:
13   script:
14     - node common/scripts/install-run-rush.js publish
```

Что дал нам Rush?

- Изоляция пакетов
- Оркестрация скриптов
- Быстрая установка зависимостей
- Версионирование
- Git-хуки
- Инкрементальная сборка

~~eslint plugin import~~

~~@changesets/cli~~

~~husky~~

Ретроспектива

Что нам дали монорепы?

- Атомарность изменений
- Возможность быстро определить, как изменения в одном проекте повлияют на другие
- Мотивация к написанию переиспользуемого кода



Спасибо за внимание!



**Никита
Балихин**

Boodle



@Landslide



**Михаил
Потапов**

Boodle



@Mppotapov