



Ниндзя кодогенерации Ускоряем разработку





Вадим Малютин

телеграм: t.me/mfrjke

почта: mltn.dev@yandex.ru

сайт: mltn.dev

Я **Senior Software Engineer** в **Apple Inc.**

Со-основатель **первой** реил-тайм виртуальной примерки одежды в вебе **clo-z.app**

Со-основатель IT-сообщества **VectorWay**, в котором организовывал и проводил мероприятия, соревнования, лекции, CTF

Со-автор приложения “**Eye Helper** тренировка для глаз”

7+ лет опыта в разработке

Сертифицированный разработчик **AWS**



Вадим Малютин

телеграм: t.me/mfrjke

почта: mltn.dev@yandex.ru

сайт: mltn.dev

Как кодер сейчас

- Пишу микрофронтенды с Module Federation и React
- Пишу микросервисы на NodeJS, NestJS, PostgreSQL и GraphQL
- Девопсю Cloud Infrastructure

Как кодер раньше

- Девопсил Cloud Infrastructure
- Разрабатывал бэкенд на Node.js использованием GraphQL, gRPC, REST, Kafka, PostgreSQL, MongoDB, DynamoDB, Timestream
- Переписывал несколько проектов (язык, технология, архитектура)
- Писал кроссплатформенные приложения на Flutter и React Native
- Сделал AR примерку одежды Three.js
- CLI for automation на Python3
- Писал Web Scraper's с PuppeteerJS / Cheerio
- Месил легаси с BEM bh, AngularJS, Python 2, Perl

Что применял в проде?

- Кодогенерацию 4-х SDK с типами из Swagger
- Прототипировал новый сервис из GQL схемы
- Генерировал документацию и песочницу из кода Apollo GQL Playground, Swagger
- Генерировал типы из GQL чтобы мигрировать с JS на TS
- Разработал несколько плагинов и трансформаторов, используя AST, в том числе писал GQL → SQL
- Генерировал хуки для работы с АПИ и управления локальным стейтом в React
- Генерировал данные, миграции и методы для работы с БД
- Писал шаблоны для быстрой генерации новых микросервисов и микрофронтендов

Что не применял в проде?

- Генерацию из схемы базы данных в готовый сервер
- Дизайн из фигмы в компоненты (React, React Native, Flutter)
- Декларативный единый GraphQL Gateway из разных протоколов
- Автоматизацию кодгена через CI / CD
- Некоторые плагины

Что будет в докладе?

- Поговорим о генерации кода
- Отвечу на вопрос “А нужно ли это мне?”
- Обзор инструментов
- Расскажу про плюсы и минусы, подводные камни, ограничения
- Демо

Что будет в докладе?

- Поговорим о генерации кода
- Отвечу на вопрос “А нужно ли это мне?”
- Обзор инструментов
- Расскажу про плюсы и минусы, подводные камни, ограничения
- Демо

Что будет в докладе?

- Поговорим о генерации кода
- Отвечу на вопрос “А нужно ли это мне?”
- **Обзор инструментов**
- Расскажу про плюсы и минусы, подводные камни, ограничения
- Демо

Что будет в докладе?

- Поговорим о генерации кода
- Отвечу на вопрос “А нужно ли это мне?”
- Обзор инструментов
- Расскажу про плюсы и минусы, подводные камни, ограничения
- Демо

Что будет в докладе?

- Поговорим о генерации кода
- Отвечу на вопрос “А нужно ли это мне?”
- Обзор инструментов
- Расскажу про плюсы и минусы, подводные камни, ограничения
- Демо

Что такое кодогенерация?

Основа программирования

```
000111101001110010101100011001010000001010011011000101000011100000001
0 Press G to generate code. If not working, click this window and try again. 11000001000
11010110111011001000110100111111111010000110101100110011110010110111
011000011011010101111110011100111101101111011001101011100100110100101
010011011101011010110111001010011110010000100111111011111010111011000
011000001100010000111011100111001101101001100001100111110000011000001
001100110111010110011111100111001010001010110111111110110100101101011
111001101110001100000011111111110101101011001001010110000111001100010
1010001010001000011100110110111111011101010010101000010101011110110
0011110000000100111110000110100010011001100000001110000000010011110010
011011010111011001111101111100101000000001011000000101111001010010101
110011101110011010110100111110000010001111111011100011001111010100001
011101111100100100100011111111101001111000100100001000010110110001000
000101000110010100111111000011111010001111011111000001100000111010111
```



```
codegen-ninja
hello-world.asm U X
prisma > [10] hello-world.asm
/ ;·compiling·the·program·...
8 ;·For·compiling·this·hello·world·program·you·can·use·nasm·and·ld·
9 ;
10 ;···$·nasm·-f·elf32·-o·hello.o·hello.asm
11 ;···$·ld·-m·elf_i386·-o·hello·hello.o
12 ;···$·./hello
13
14 section······.text··········;·declare·the·.text·section
15 global······_start··········;·has·to·be·declared·for·the·linker·(ld)
16 _start:··········;·entry·point·for·_start
17 ····mov·edx,·len··········;·"invoke"·the·len·of·the·message
18 ····mov·ecx,·msg··········;·"invoke"·the·message·itself
19
20 ····mov·ebx,·1··········;·set·the·file·descriptor·(fd)·to·stdout
21
22 ····mov·eax,·4··········;·system·call·for·"write"···
23 ····int·0x80··········;·call·the·kernel
24
25 ····mov·eax,·1··········;·system·call·for·"exit"
26 ····int·0x80··········;·call·the·kernel
27
28 section······.data··········;·here·you·declare·the·data
29 ····msg·····db·"Hello·world!"······;·the·actual·message·to·use
30 ····len·····equ·$-msg··········;·get·the·size·of·the·message
```

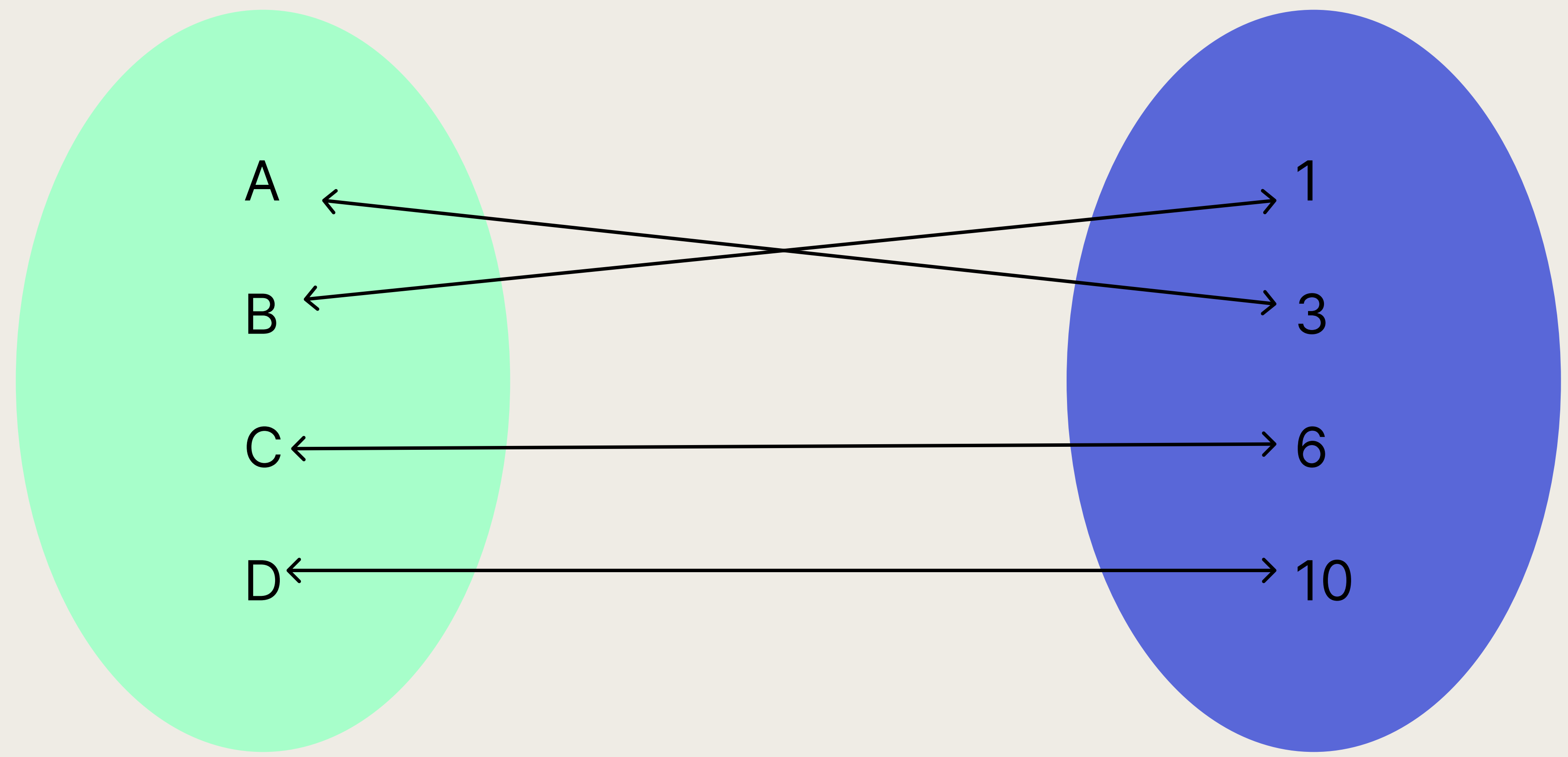
```
v5.2.2 ▾ Run Export ▾ Share → .JS .D.TS Errors Logs Plugins
3
4 // By default an enum is number based
5 // and each option is assigned an inc
6 // useful when the value is not impor
7
8 enum CompassDirection {
9   North,
10  East,
11  South,
12  West,
13 }
14
15 // By annotating an enum option, you
16 // increments continue from that valu
17
18 enum StatusCodes {
19   OK = 200,
20   BadRequest = 400,
}

// and each option is assigned an increment by one. THIS IS
// useful when the value is not important.
var CompassDirection;
(function (CompassDirection) {
  CompassDirection[CompassDirection["North"] = 0] = "North";
  CompassDirection[CompassDirection["East"] = 1] = "East";
  CompassDirection[CompassDirection["South"] = 2] = "South";
  CompassDirection[CompassDirection["West"] = 3] = "West";
})(CompassDirection || (CompassDirection = {}));
// By annotating an enum option, you set the value;
// increments continue from that value:
var StatusCodes;
(function (StatusCodes) {
  StatusCodes[StatusCodes["OK"] = 200] = "OK";
  StatusCodes[StatusCodes["BadRequest"] = 400] = "BadRequest";
})
```

```
☰ BABEL
1 ReactDOM.render(<h1>Hello world</h1>,
  document.getElementById('hello'));
2
3 ReactDOM.render(
  /*#__PURE__*/React.createElement("h1", null, "Hello
  world"), document.getElementById('hello'));
```


Множество правил букв

Множество правил цифр



Нужно ли мне генерировать код и
зачем?

У всех по-разному

- Для компании
 - Ускоряет разработку и деливери → дешевле и быстрее
 - Унификация структуры и стиля кода → дешевле и быстрее
 - Упрощает взаимодействие команд → быстрее и безопаснее
 - Снижает вероятность ошибок → безопаснее
- Для разработчика
 - Освободить себе время
 - Не заниматься монотонной работой
 - Быстро прототипировать свои проекты
- Для дизайнера
 - Проще делать pixel perfect дизайн
 - Есть возможность воплощать свои дизайны в реальность без кода
 - Проще общаться с фронтендерами

У всех по-разному

- Для компании
 - Ускоряет разработку и деливери → дешевле и быстрее
 - Унификация структуры и стиля кода → дешевле и быстрее
 - Упрощает взаимодействие команд → быстрее и безопаснее
 - Снижает вероятность ошибок → безопаснее
- Для разработчика
 - Освободить себе время
 - Не заниматься монотонной работой
 - Быстро прототипировать свои проекты
- Для дизайнера
 - Проще делать pixel perfect дизайн
 - Есть возможность воплощать свои дизайны в реальность без кода
 - Проще общаться с фронтендерами

У всех по-разному

- Для компании
 - Ускоряет разработку и деливери → дешевле и быстрее
 - Унификация структуры и стиля кода → дешевле и быстрее
 - Упрощает взаимодействие команд → быстрее и безопаснее
 - Снижает вероятность ошибок → безопаснее
- Для разработчика
 - Освободить себе время
 - Не заниматься монотонной работой
 - Быстро прототипировать свои проекты
- Для дизайнера
 - Проще делать pixel perfect дизайн
 - Есть возможность воплощать свои дизайны в реальность без кода
 - Проще общаться с фронтендерами

Что будем кодогенерировать?



shaOnline

Search bujinkan's by name, sensei or subject...



From 5/20/2023

To 7/15/2023

Joined



Bujinkan

Sensei

Subject

Padowans

14:00, 6/23/2013

join



Bujinkan

Sensei

Subject

Padowans

14:00, 6/23/2013

join



Bujinkan1

Sensei

Subject

Padowans

14:00, 6/23/2013

cancel



Bujinkan

Sensei

Subject

Padowans

14:00, 6/23/2013

join



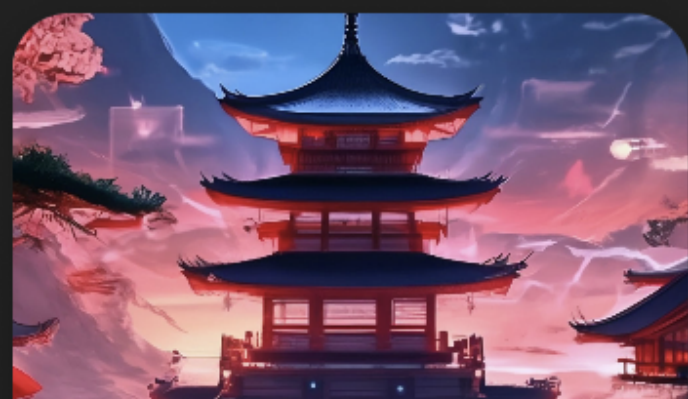
Bujinkan

Sensei

Subject

Padowans

join



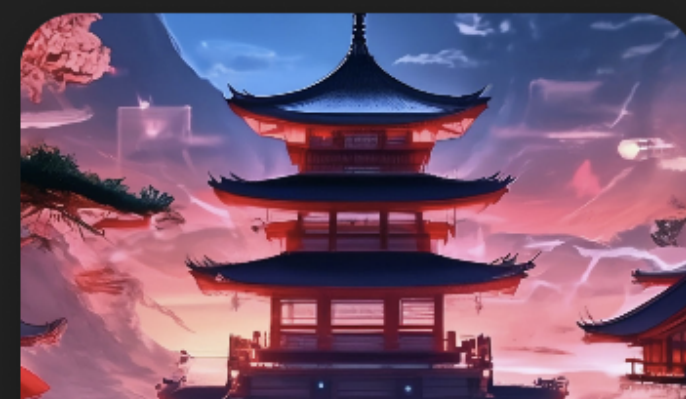
Bujinkan1

Sensei

Subject

Padowans

cancel



Bujinkan1

Sensei

Subject

Padowans

cancel



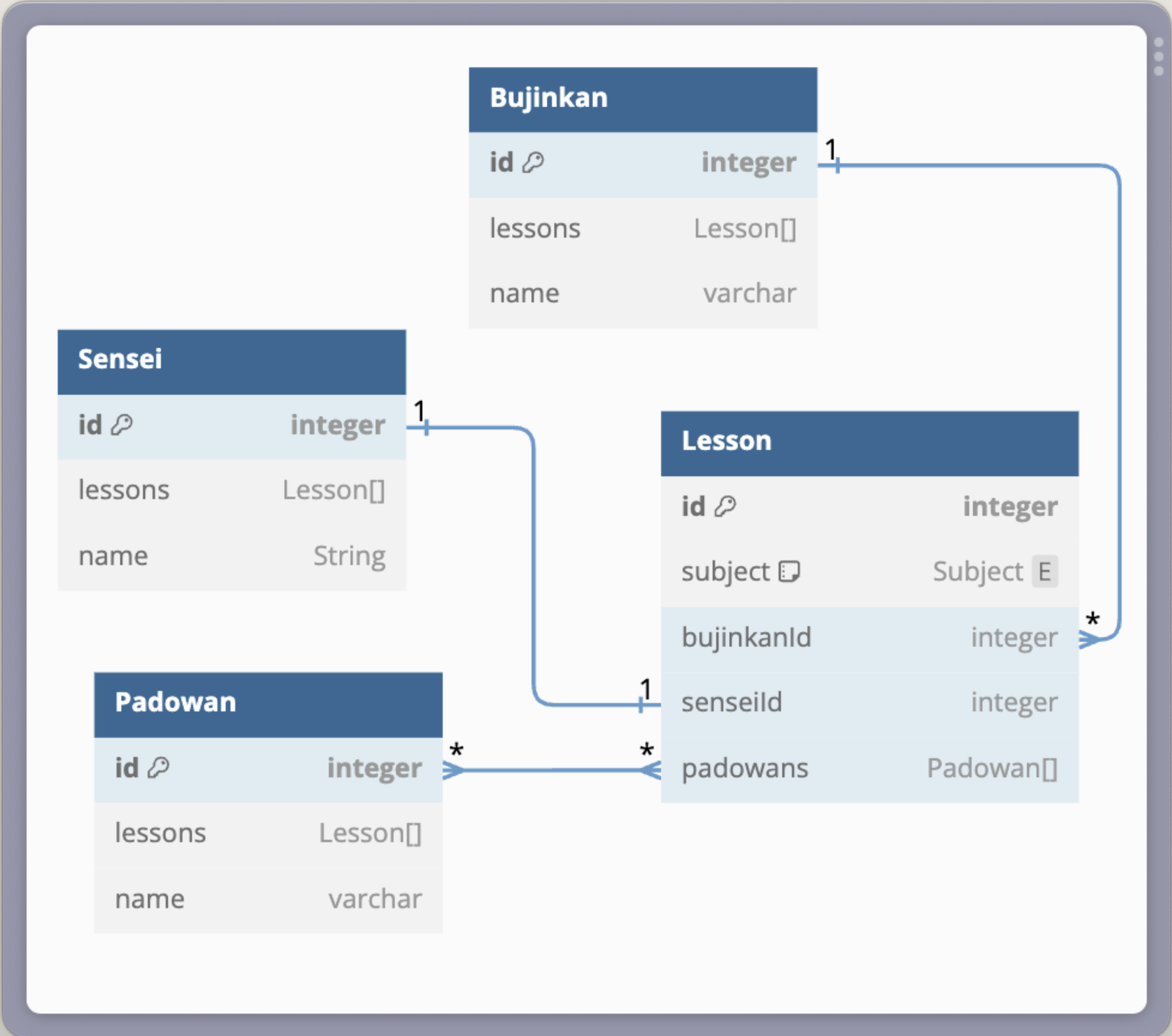
Bujinkan1

Sensei

Subject

Padowans

cancel



Поехали

Генерируем структуру проекта



- NX
- Turbo
- Gatsby
- Next.js
- CRACO



- Hygen
- Plop
- Yeoman
- Slush



- Samples
- Open-source
- Boilerplates

Общие технологии проекта



Генерируем Бэкенд

```
vmaliutin@Vadims-MacBook-Pro-2:~/Projects/Personal
> npx create-nx-workspace@latest

> NX Let's create a new workspace [https://nx.dev/getting-started/intro]

✓ Where would you like to create your workspace? · codegen-ninja
✓ Which stack do you want to use? · node
✓ What framework should be used? · nest
✓ Integrated monorepo, or standalone project? · integrated
✓ Application name · backend
✓ Would you like to generate a Dockerfile? [https://docs.docker.com/] · Yes
✓ Enable distributed caching to make your CI faster · Yes

> NX Creating your v17.0.2 workspace.

To make sure the command works reliably in all environments, and that the preset
is applied correctly,
Nx will run "npm install" several times. Please wait.

✓ Installing dependencies with npm
✓ Successfully created the workspace: codegen-ninja.
✓ NxCloud has been set up successfully
```


Генерируем Фронтенд

```
nx g @nx/react:application frontend
```

> **NX** Its time to update Nx 🎉

Your repository uses a higher version of Nx (17.0.2) than your global CLI version (16.10.0)
For more information, see <https://nx.dev/more-concepts/global-nx>

> **NX** Generating @nx/react:application

? Which stylesheet format would you like to use? ...

- CSS
- SASS(.scss) [<http://sass-lang.com>]
- LESS [<http://lesscss.org>]
- styled-components [<https://styled-components.com>]
- emotion [<https://emotion.sh>]
- styled-jsx [<https://www.npmjs.com/package/styled-jsx>]
- None

Генерируем Фронтенд

```
nx g @nx/react:application frontend

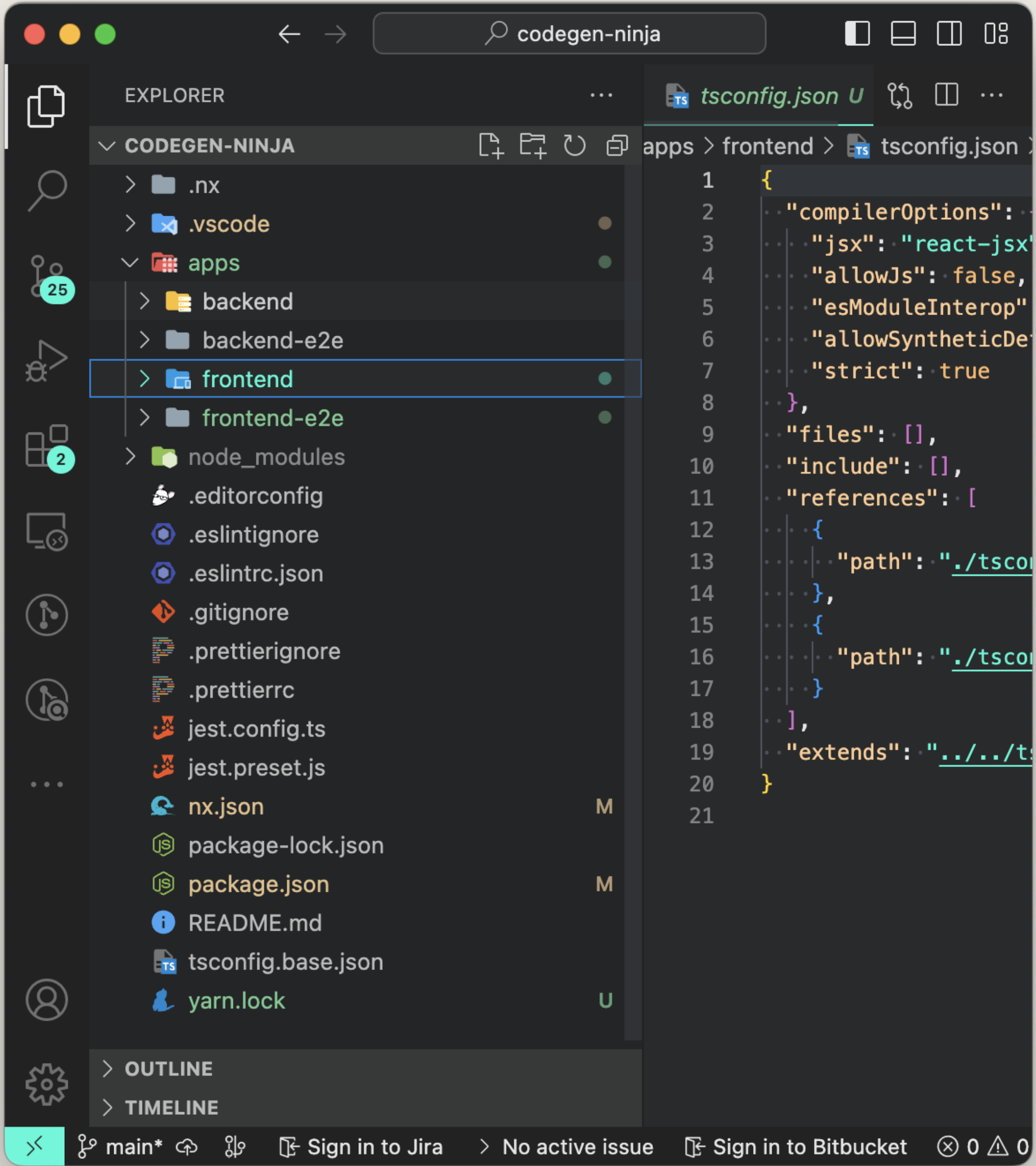
> NX Its time to update Nx 🎉

Your repository uses a higher version of Nx (17.0.2) than your global CLI version (16.10.0)
For more information, see https://nx.dev/more-concepts/global-nx

> NX Generating @nx/react:application

✓ Which stylesheet format would you like to use? · styled-components
✓ Would you like to add React Router to this application? (y/N) · true
✓ Which E2E test runner would you like to use? · playwright
✓ Which bundler do you want to use to build the application? · webpack
✓ What should be the project name and where should it be generated? · frontend @ apps/frontend

In Nx 18, generating projects will no longer derive the name and root.
Please provide the exact project name and root in the future.
Example: nx g @nx/react:application frontend --directory frontend
Fetching @nx/playwright...
UPDATE nx.json
UPDATE package.json
CREATE apps/frontend/src/app/app.spec.tsx
CREATE apps/frontend/src/assets/.gitkeep
CREATE apps/frontend/src/environments/environment.prod.ts
CREATE apps/frontend/src/environments/environment.ts
CREATE apps/frontend/src/favicon.ico
CREATE apps/frontend/src/index.html
CREATE apps/frontend/src/main.tsx
CREATE apps/frontend/tsconfig.app.json
CREATE apps/frontend/webpack.config.js
```

codegen-ninja

EXPLORER

- CODEGEN-NINJA
 - .nx
 - .vscode
 - apps
 - backend
 - src
 - app
 - app.controller.spec.ts
 - app.controller.ts
 - app.module.ts
 - app.service.spec.ts
 - app.service.ts
 - assets
 - main.ts
 - .eslintrc.json
 - jest.config.ts
 - project.json
 - tsconfig.app.json
 - tsconfig.json
 - tsconfig.spec.json
 - webpack.config.js
 - backend-e2e
 - frontend
 - frontend-e2e
 - OUTLINE
 - TIMELINE

tsconfig.json U

```
1 {
2   "compilerOptions": {
3     "jsx": "react-jsx",
4     "allowJs": false,
5     "esModuleInterop": true,
6     "allowSyntheticDefaultImports": true,
7     "strict": true
8   },
9   "files": [],
10  "include": [],
11  "references": [
12    {
13      "path": "./tsconfig.app.json"
14    },
15    {
16      "path": "./tsconfig.spec.json"
17    }
18  ],
19  "extends": "../../tsconfig.json"
20 }
21
```

main* Sign in to Jira No active issue Sign in to Bitbucket 0 0

codegen-ninja

EXPLORER

CODEGEN-NINJA

- apps
 - backend
 - backend-e2e
 - frontend
 - src
 - app
 - app.spec.tsx
 - app.tsx
 - nx-welcome.tsx
 - assets
 - environments
 - favicon.ico
 - index.html
 - main.tsx
 - .babelrc
 - .eslintrc.json
 - jest.config.ts
 - project.json
 - tsconfig.app.json
 - tsconfig.json
 - tsconfig.spec.json
 - webpack.config.js
 - frontend-e2e

tsconfig.json

```
1 {
2   "compilerOptions": {
3     "jsx": "react-jsx",
4     "allowJs": false,
5     "esModuleInterop": true,
6     "allowSyntheticDefaultImports": true,
7     "strict": true
8   },
9   "files": [],
10  "include": [],
11  "references": [
12    {
13      "path": "./tsconfig.app.json"
14    },
15    {
16      "path": "./tsconfig.spec.json"
17    }
18  ],
19  "extends": "../tsconfig.json"
20 }
21
```

main* Sign in to Jira No active issue Sign in to Bitbucket 0 0

Hello there,
Welcome frontend 🙌

✅ You're up and running

What's next?



Learning materials

📖 **Documentation** >
Everything is in there

📄 **Blog**



Install Nx Console for
VSCode
The official VSCode plugin for
Nx.

 http://localhost:3000/api   |    |  

```
{"message": "Users count: 0"}
```

Кастомная генерация

Зачем?

- Нестандартная структура проекта
- Нет подходящего плагина
- Нужно добавлять код в существующие файлы
- Избавляться от монотонной работы

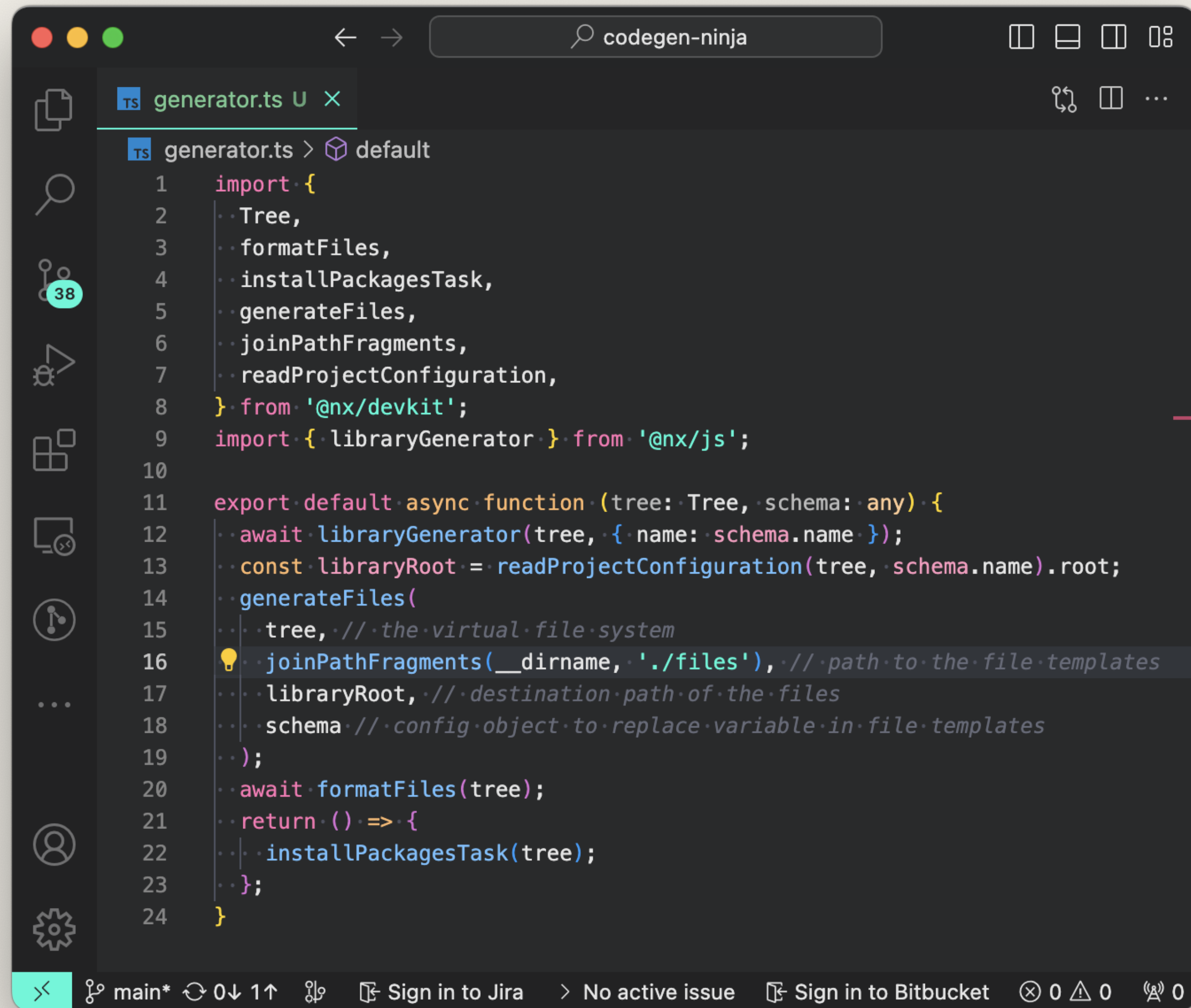
Генерируем плагин

```
vmaliutin@Vadims-MacBook-Pro-2:~/Projects/Personal/codegen-ninja 1
> yarn nx g @nx/plugin:plugin my-plugin
yarn run v1.22.19
$ /Users/vmaliutin/Projects/Personal/codegen-ninja/node_modules/.bin/nx g @nx/plugin:plugin my-plugin

> NX Generating @nx/plugin:plugin

> NX We've updated the vscode settings for this repository to ensure that plugin lint checks show up inside your IDE. This created .vscode/settings.json. To read more about this file, check vscode's documentation. It is frequently not committed, so other developers may need to add similar settings if they'd like to see the lint checks in the IDE rather than only during linting.
UPDATE package.json
CREATE my-plugin/tsconfig.json
CREATE my-plugin/README.md
CREATE my-plugin/src/index.ts
CREATE my-plugin/tsconfig.lib.json
CREATE my-plugin/package.json
CREATE my-plugin/project.json
CREATE my-plugin/.eslintrc.json
CREATE my-plugin/jest.config.ts
CREATE my-plugin/tsconfig.spec.json
UPDATE tsconfig.base.json
UPDATE .eslintrc.json
CREATE .vscode/settings.json
```


Реализуем генератор



```
1 import {
2   Tree,
3   formatFiles,
4   installPackagesTask,
5   generateFiles,
6   joinPathFragments,
7   readProjectConfiguration,
8 } from '@nx/devkit';
9 import { libraryGenerator } from '@nx/js';
10
11 export default async function (tree: Tree, schema: any) {
12   await libraryGenerator(tree, { name: schema.name });
13   const libraryRoot = readProjectConfiguration(tree, schema.name).root;
14   generateFiles(
15     tree, // the virtual file system
16     joinPathFragments(__dirname, './files'), // path to the file templates
17     libraryRoot, // destination path of the files
18     schema // config object to replace variable in file templates
19   );
20   await formatFiles(tree);
21   return () => {
22     installPackagesTask(tree);
23   };
24 }
```

main* 0↓ 1↑ Sign in to Jira > No active issue Sign in to Bitbucket 0 0 0

Генератор посложнее

```
20     ···});  
21   }  
22  
23   export default function (tree: Tree, schema: any) {  
24     ···const sourceRoot = readProjectConfiguration(tree, schema.name).sourceRoot!  
25     ···visitAllFiles(tree, sourceRoot, (filePath) => {  
26       ···const fileEntry = tree.read(filePath);  
27       ···const contents = fileEntry?.toString();  
28  
29       ···// Check each `TypeReference` node to see if we need to replace it  
30       ···const newContents = tsquery.replace(contents!, 'TypeReference', (node) => {  
31         ···const trNode = node as TypeReferenceNode;  
32         ···if (trNode.typeName.getText() === 'Array') {  
33           ···const typeArgument = trNode.typeArguments?.[0];  
34           ···return `${typeArgument?.getText()}[]`;  
35         }  
36         ···// return undefined does not replace anything  
37         }  
38       }  
39       ···// only write the file if something has changed  
40       ···if (newContents !== contents) {  
41         ···tree.write(filePath, newContents);  
42       }  
43     }  
44   }  
}
```

Минусы

- Сложно настраивать
- Много писать



Нуген и ему подобные

The screenshot shows a code editor window with the following details:

- Search bar: codegen-ninja
- Explorer sidebar: CODEGEN-NINJA, _templates/component, help, new, app.ejs.t (selected), component.ejs.t, prompt.js
- File tabs: app.service.ts, component.ejs.t U, app.ejs.t U X, company-name.d.ts, .gitignore, .env
- Code content (app.ejs.t):

```
1 ---
2 inject: true
3 to: apps/frontend/src/app/app.tsx
4 after: import NxWelcome from './nx-welcome';
5 ---
6
7 import <%= h.changeCase.pascal(name) %> from '../components/<%= h.inflection.dasherize(name) %>.tsx';
```
- Status bar: main*, 0↓ 1↑, Sign in to Jira, No active issue, Sign in to Bitbucket, 0⊗ 0⚠ 0🔊, AWS: profile:default, Wallaby, Spaces: 4, UTF-8, LF, Perl, Prettier

Подсказки

```
vmaliutin@Vadims-MacBook-Pro-2:~/Projects/Personal/codegen-ninja Ƶ⌘1
> yarn hygen component new
yarn run v1.22.19
$ /Users/vmaliutin/Projects/Personal/codegen-ninja/node_modules/.bin/hygen component new
✓ What is new component name? · Button

Loaded templates: _templates
  inject: apps/frontend/src/app/app.tsx
  added: apps/frontend/src/components/Button.tsx
✨ Done in 6.29s.

~/Pr/P/codegen-ninja on main ↑1 !2 ?3 > | took ⌚ 6s
```


Генерируем данные

prisma.io



Next-generation **Node.js** and **TypeScript ORM**

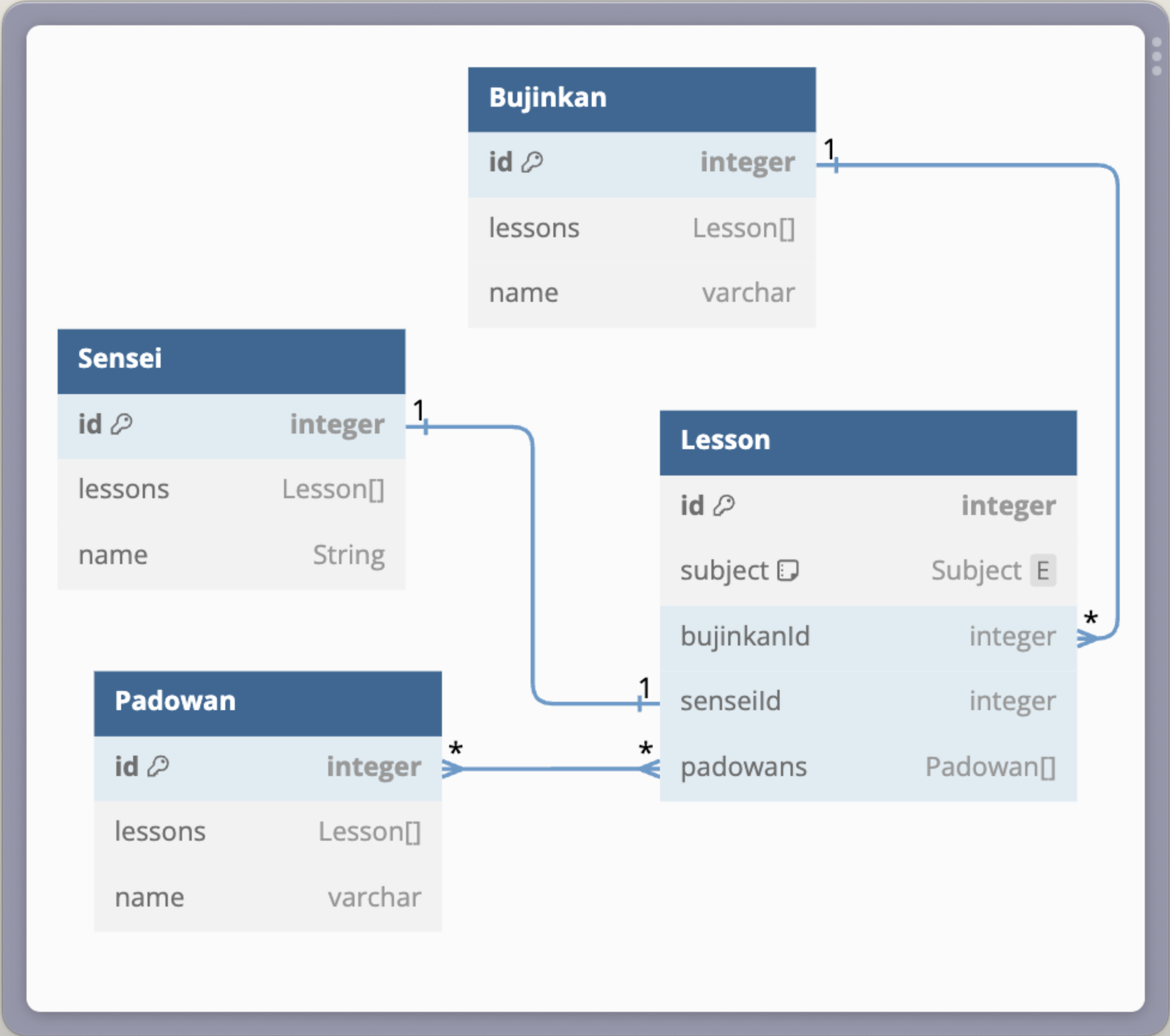
Prisma unlocks a new level of **developer experience** when working with databases thanks to its intuitive data model, automated migrations, type-safety & auto-completion.

[Quickstart](#)

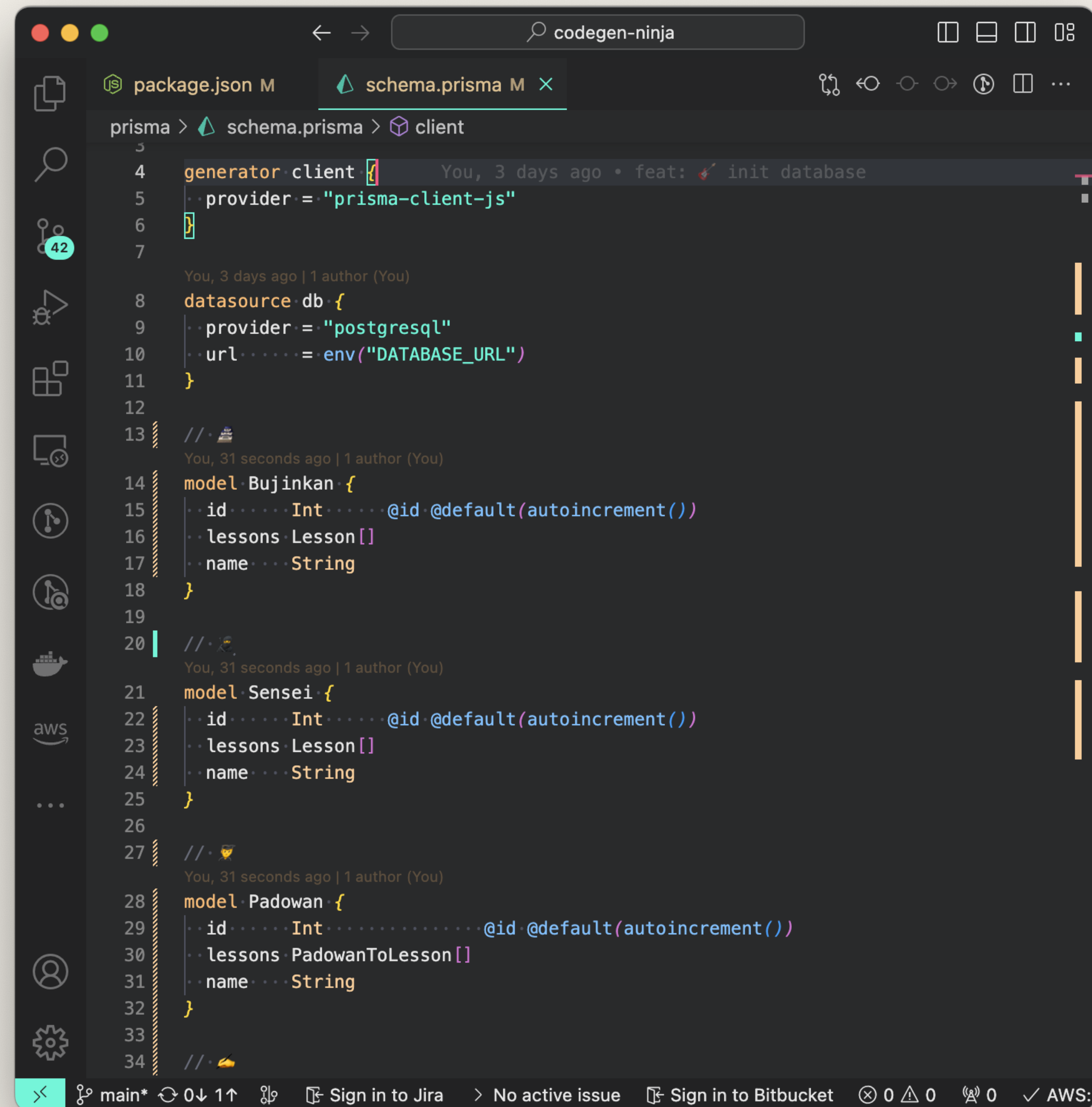
[Playground ↗](#)

Почему?

- Быстрая настройка на SQL-подобном языке
- Не нужно писать самому entity / декораторы
- Генерация client ORM, миграций, энтити, схем и релейшенов на основании схемы
- Студия из коробки
- Простой сидинг



Описали схему Ч. 1

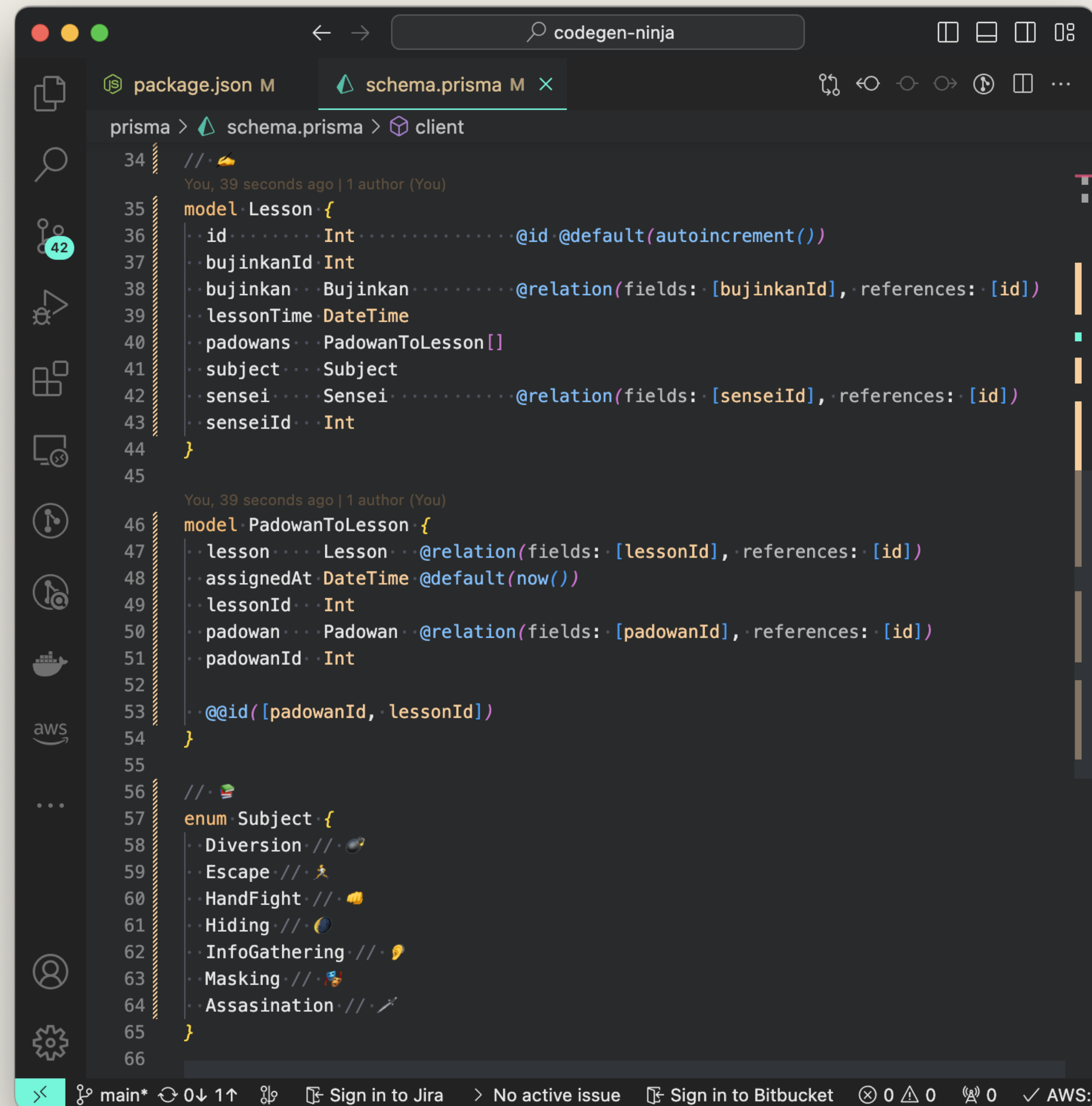


```
prisma > schema.prisma > client
4 generator client {
5   provider = "prisma-client-js"
6 }
7
8 datasource db {
9   provider = "postgresql"
10  url       = env("DATABASE_URL")
11 }
12
13 // ...
14 model Bujinkan {
15   id      Int      @id @default(autoincrement())
16   lessons Lesson[]
17   name    String
18 }
19
20 // ...
21 model Sensei {
22   id      Int      @id @default(autoincrement())
23   lessons Lesson[]
24   name    String
25 }
26
27 // ...
28 model Padowan {
29   id      Int      @id @default(autoincrement())
30   lessons PadowanToLesson[]
31   name    String
32 }
33
34 // ...
```

The screenshot shows a code editor window with the following details:

- Browser tabs: package.json M, schema.prisma M
- File path: prisma > schema.prisma > client
- Line numbers: 3 to 34
- Code content: Prisma schema for a client, PostgreSQL data source, and three models (Bujinkan, Sensei, Padowan).
- Model fields: id (Int, @id, @default(autoincrement())), lessons (array), name (String).
- Model relationships: Padowan has a relationship to Lesson (PadowanToLesson).
- Editor interface: Includes a sidebar with icons for search, git, and other tools. The bottom status bar shows "main*" and various utility icons.

Описали схему Ч. 2



```
prisma > schema.prisma > client
34 // 🍌
35 You, 39 seconds ago | 1 author (You)
36 model Lesson {
37   · id ····· Int ····· @id @default(autoincrement())
38   · bujinkanId Int
39   · bujinkan · Bujinkan ····· @relation(fields: [bujinkanId], references: [id])
40   · lessonTime DateTime
41   · padowans · PadawanToLesson[]
42   · subject · Subject
43   · sensei · Sensei ····· @relation(fields: [senseiId], references: [id])
44   · senseiId · Int
45 }
46 You, 39 seconds ago | 1 author (You)
47 model PadawanToLesson {
48   · lesson · Lesson ····· @relation(fields: [lessonId], references: [id])
49   · assignedAt DateTime @default(now())
50   · lessonId · Int
51   · padawan · Padawan ····· @relation(fields: [padawanId], references: [id])
52   · padawanId · Int
53   · @@id([padawanId, lessonId])
54 }
55
56 // 🍌
57 enum Subject {
58   · Diversion // 🌀
59   · Escape // 🏃
60   · HandFight // 🥊
61   · Hiding // 🌑
62   · InfoGathering // 🕵️
63   · Masking // 🎭
64   · Assassination // 🗡️
65 }
66
```

Сгенерировали ORM client

```
vmaliutin@Vadims-MacBook-Pro-2:~/Projects/Personal/codegen-ninja 1
> yarn db:generate
yarn run v1.22.19
$ prisma generate
Environment variables loaded from .env
Prisma schema loaded from prisma/schema.prisma

✓ Generated Prisma Client (v5.4.2) to ./node_modules/@prisma/client in 67ms

Start using Prisma Client in Node.js (See: https://pris.ly/d/client)
```
import { PrismaClient } from '@prisma/client'
const prisma = new PrismaClient()
```
or start using Prisma Client at the edge (See: https://pris.ly/d/accelerate)
```
import { PrismaClient } from '@prisma/client/edge'
const prisma = new PrismaClient()
```
See other ways of importing Prisma Client: http://pris.ly/d/importing-client

Update available 5.4.2 -> 5.5.2
Run the following to update
yarn add --dev prisma@latest
yarn add @prisma/client@latest

✨ Done in 0.88s.
```


Сгенерировали миграцию

```
vmaliutin@Vadims-MacBook-Pro-2:~/Projects/Personal/codegen-ninja 1
> yarn db:migrate
yarn run v1.22.19
$ prisma migrate dev
Environment variables loaded from .env
Prisma schema loaded from prisma/schema.prisma
Datasource "db": PostgreSQL database "postgres", schema "public" at "localhost:5432"

Applying migration `20231014145657_init`

The following migration(s) have been applied:

migrations/
├─ 20231014145657_init/
│  └─ migration.sql
└─
✓ Enter a name for the new migration: ...
Applying migration `20231026134701_`

The following migration(s) have been created and applied from new schema changes:

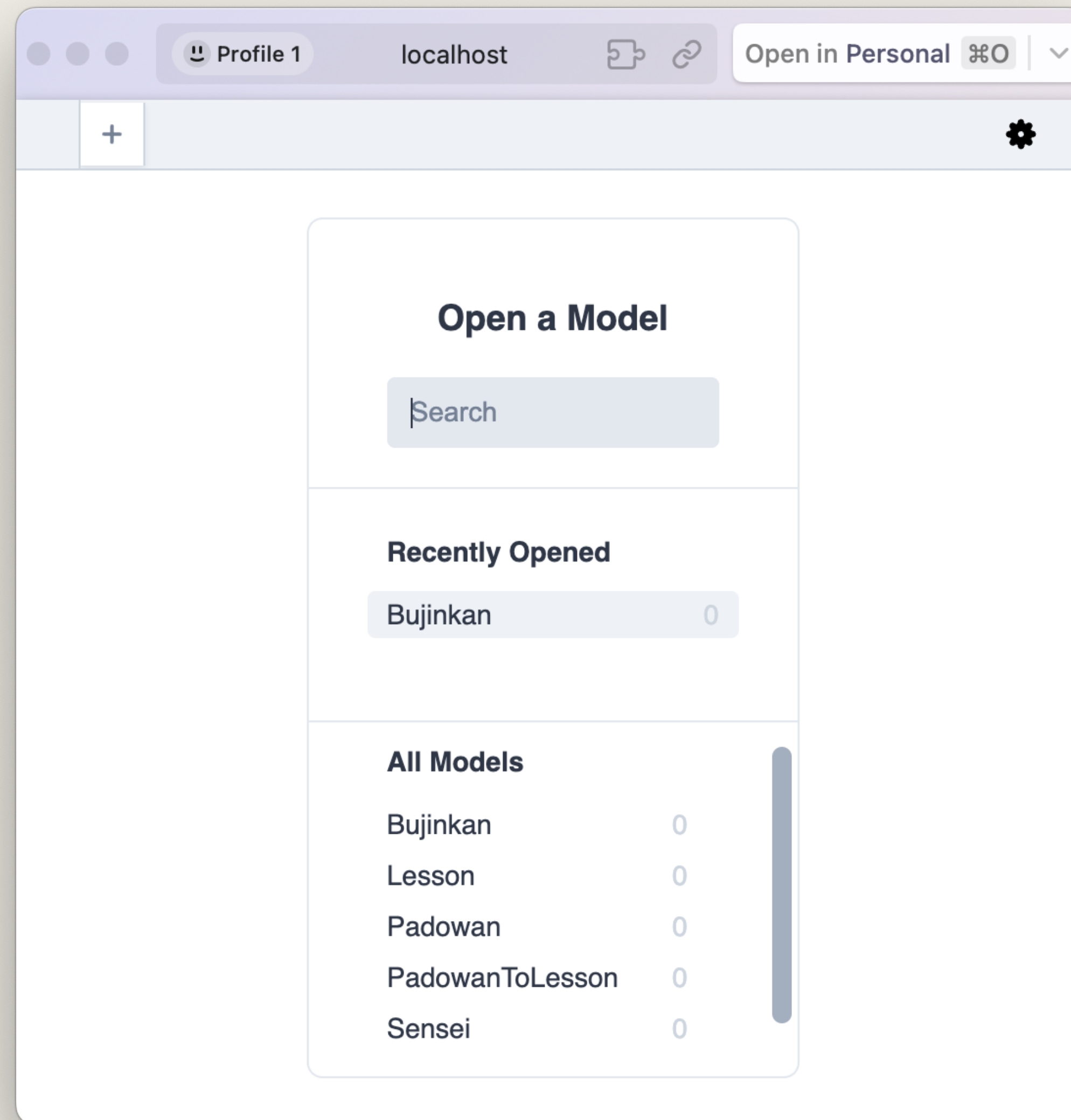
migrations/
├─ 20231026134701_/
│  └─ migration.sql
└─

Your database is now in sync with your schema.

✓ Generated Prisma Client (v5.4.2) to ./node_modules/@prisma/client in 60ms

✨ Done in 7.91s.
```


Получили БД Ч.1



Получили БД Ч.2

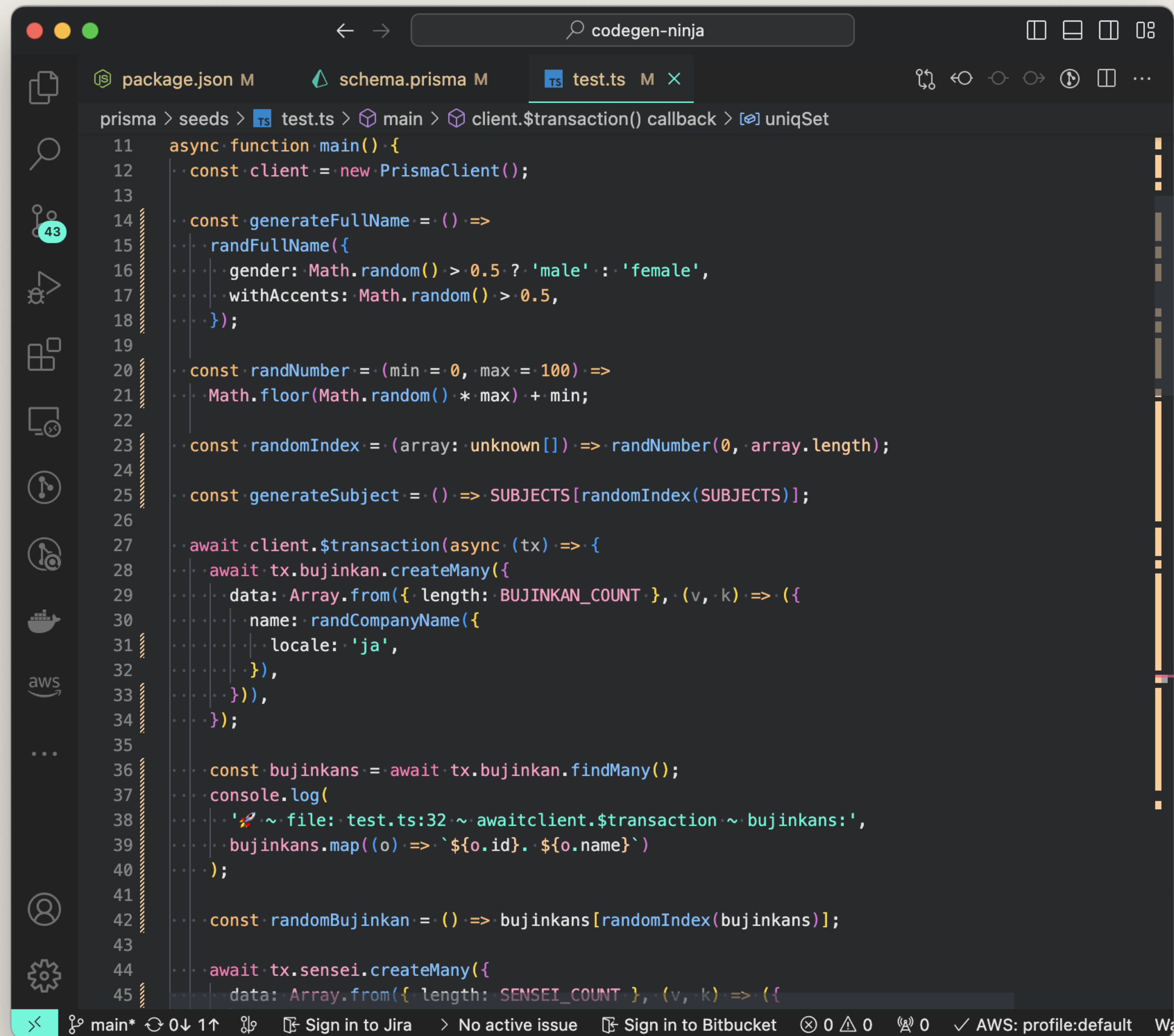
The screenshot shows a web application interface for a database. At the top, there's a browser window with a tab labeled 'Profile 1' and the address 'localhost'. Below the browser, there's a navigation bar with a 'Lesson' tab and a '+', a refresh button, 'Filters None', 'Fields All', 'Showing 0 of 0', and an 'Add record' button. The main area displays a table with the following columns: 'id #', 'bujinkanId #', 'bujinkan {}', 'lessonTime', 'padowans []', 'subject', 'sensei {}', and 'senseid #'. The table is empty, and a message at the bottom states 'There are no rows in this table'.

<input type="checkbox"/>	id #	bujinkanId #	bujinkan {}	lessonTime	padowans []	subject	sensei {}	senseid #
There are no rows in this table								

Минусы

- Не удобный rollback
- Может не быть вашей бд

Скрипт для сидинга



```
prisma > seeds > test.ts > main > client.$transaction() callback > uniqSet
11  async function main() {
12    const client = new PrismaClient();
13
14    const generateFullName = () =>
15      randFullName({
16        gender: Math.random() > 0.5 ? 'male' : 'female',
17        withAccents: Math.random() > 0.5,
18      });
19
20    const randNumber = (min = 0, max = 100) =>
21      Math.floor(Math.random() * max) + min;
22
23    const randomIndex = (array: unknown[]) => randNumber(0, array.length);
24
25    const generateSubject = () => SUBJECTS[randomIndex(SUBJECTS)];
26
27    await client.$transaction(async (tx) => {
28      await tx.bujinkan.createMany({
29        data: Array.from({ length: BUJINKAN_COUNT }, (v, k) => ({
30          name: randCompanyName({
31            locale: 'ja',
32          }),
33        })),
34      });
35
36      const bujinkans = await tx.bujinkan.findMany();
37      console.log(
38        '~ file: test.ts:32 ~ await client.$transaction ~ bujinkans:',
39        bujinkans.map((o) => `${o.id}. ${o.name}`)
40      );
41
42      const randomBujinkan = () => bujinkans[randomIndex(bujinkans)];
43
44      await tx.sensei.createMany({
45        data: Array.from({ length: SENSEI_COUNT }, (v, k) => ({
```


Генерируем данные

```
vmaliutin@Vadims-MacBook-Pro-2:~/Projects/Personal/codegen-ninja ㉿#1
]
~ file: test.ts:32 ~ awaitclient.$transaction ~ lessons: [
  '19. Hiding',      '18. Masking',      '14. Hiding',
  '13. Masking',     '14. Hiding',      '11. Hiding',
  '19. Escape',      '13. Masking',     '19. Hiding',
  '18. Assassination', '13. Escape',      '18. Escape',
  '15. Masking',     '19. Hiding',      '18. InfoGathering',
  '14. Assassination', '11. Escape',      '11. Escape',
  '17. HandFight',   '17. InfoGathering', '13. Diversion',
  '11. Assassination', '14. Assassination', '12. Masking',
  '16. Hiding',      '17. InfoGathering', '17. Assassination',
  '20. Escape',      '18. Escape',      '19. Diversion',
  '18. Masking',     '11. Masking',     '14. Assassination',
  '19. Assassination', '16. Escape',      '12. Masking',
  '11. InfoGathering', '17. Masking',     '14. Masking',
  '13. Masking',     '19. Hiding',      '19. InfoGathering',
  '20. Diversion',   '11. Hiding',      '18. Masking',
  '13. Masking',     '15. Diversion',   '20. Escape',
  '17. InfoGathering', '14. Assassination', '14. Hiding',
  '18. Diversion',   '11. InfoGathering', '19. Masking',
  '15. Masking',     '13. Hiding',      '12. Hiding',
  '20. HandFight',   '13. InfoGathering', '13. Masking'
]
🌱 The seed command has been executed.
🌟 Done in 1.03s.
~/Pr/P/codegen-ninja on main ↕1 !5 ?10 >
```

Получаем в БД

The screenshot shows a database management interface with a table of lesson records. The interface includes a browser-like header with 'localhost', a toolbar with 'Filters', 'Fields', 'Showing 60 of 60', and an 'Add record' button. The table has columns for 'id #', 'bujinkanId #', 'bujinkan {}', 'lessonTime', 'padowans []', and 'subject'. The data rows show lesson IDs 61 through 66, each associated with a bujinkan ID, a lesson time, a number of padowans, and a subject.

<input type="checkbox"/>	id #	bujinkanId #	bujinkan {}	lessonTime	padowans []	subject
<input type="checkbox"/>	61	19	Bujinkan	2023-02-22T01:30:44.9...	7 PadowanToLesson	Hiding
<input type="checkbox"/>	62	18	Bujinkan	2023-04-09T17:38:24.1...	6 PadowanToLesson	Masking
<input type="checkbox"/>	63	14	Bujinkan	2023-06-06T12:35:42.7...	6 PadowanToLesson	Hiding
<input type="checkbox"/>	64	13	Bujinkan	2023-03-10T09:56:45.5...	3 PadowanToLesson	Masking
<input type="checkbox"/>	65	14	Bujinkan	2023-07-12T17:37:30.6...	1 PadowanToLesson	Hiding
<input type="checkbox"/>	66	11	Bujinkan	2023-06-17T23:57:16.3...	7 PadowanToLesson	Hiding

БД в Сервис

hasura.io

Hasura v2.34.0-cloud.1 | API | **DATA** | ACTIONS | REMOTE SCHEMAS | EVENTS | MONITORING | sincere

Data Manager

Databases (1) Manage

- codegen-ninja
 - public
 -
 - Bujinkan**
 - Lesson
 - Padowan
 - PadowanToLesson
 - Sensei
 - _prisma_migrations

Bujinkan Try GraphQL Create REST Endpoints **New**

Browse Rows | Insert Row | Modify | Relationships | Permissions

Filters
No Filters Present
Add

Sort
No sort conditions present.
Add

Run query Export data

Prev 10 rows Next

	<input type="checkbox"/>	id	name	Lessons
Copy Edit Delete Refresh	<input type="checkbox"/>	1	Orn, Spencer and Kiehn	View
Copy Edit Delete Refresh	<input type="checkbox"/>	2	Reichel Group	View
Copy Edit Delete Refresh	<input type="checkbox"/>	3	Mertz, Gusikowski and Lemke	View
Copy Edit Delete Refresh	<input type="checkbox"/>	4	Gutkowski Inc	View
Copy Edit Delete Refresh	<input type="checkbox"/>	5	Fahey, Leannon and Gleichner	View

SQL

Native Queries

GraphQL Endpoint

POST https://

Relay API

Request Headers

GraphQL Prettify History Explorer Cache Code Exporter REST Derive action Analyze

```

1 # Try out GraphQL queries here
2 query GetTenBujinkans {
3   ninjas {
4     Bujinkan(limit: 10) {
5       name
6       id
7     Lessons {
8       id
9       subject
10      Sensei {
11        id
12        name
13      }
14     PadowanToLessons {
15       Padowan {
16         id
17         name
18       }
19     }
20   }
21 }

```

QUERY VARIABLES

```

{
  "data": {
    "ninjas": {
      "Bujinkan": [
        {
          "name": "Orn, Spencer and Kiehn",
          "id": 1,
          "Lessons": [
            {
              "id": 9,
              "subject": "Escape",
              "Sensei": {
                "id": 21,
                "name": "Götz Halldórsson"
              },
              "PadowanToLessons": [
                {
                  "Padowan": {
                    "id": 20,
                    "name": "Mercy Suleiman"
                  }
                }
              ]
            }
          ]
        }
      ]
    }
  }
}

```

RESPONSE TIME 964 ms RESPONSE SIZE 21166 bytes

< Lessons Lesson X

An array relationship

PadowanToLessons_aggregate(
 distinct_on: [PadowanToLesson_select_column!]
 limit: Int
 offset: Int
 order_by: [PadowanToLesson_order_by!]
 where: PadowanToLesson_bool_exp
): PadowanToLesson_aggregate!
An aggregate relationship

Sensei: Sensei!
An object relationship

bujinkanId: Int!

id: Int!

lessonTime: timestamp!

senseiId: Int!

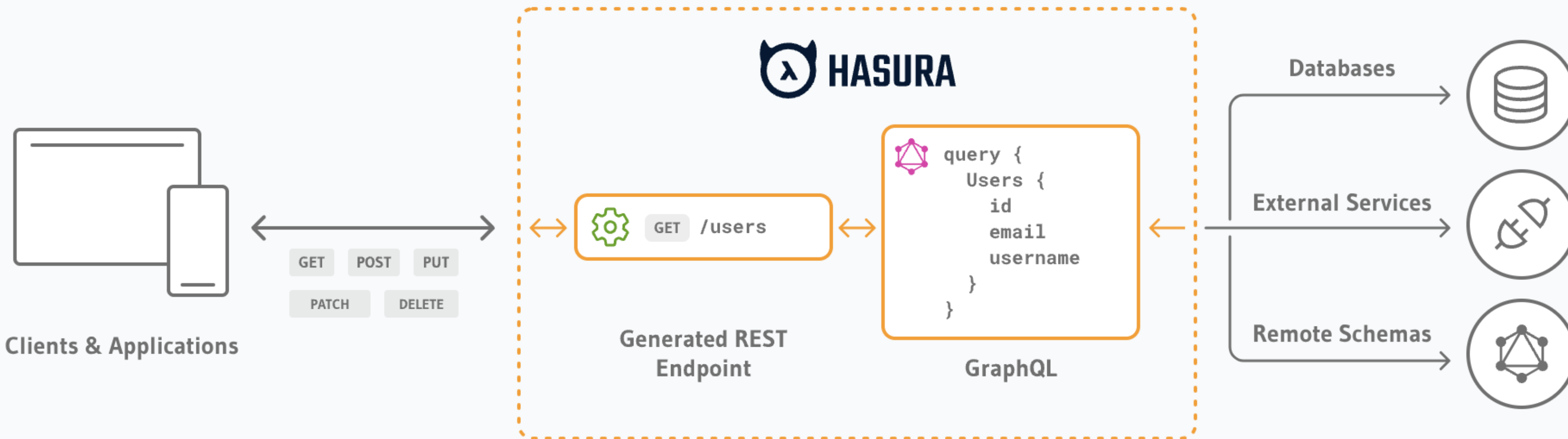
subject: Subject!

REST Endpoints

Create REST

Create Rest endpoints on the top of existing GraphQL queries and mutations

What are REST endpoints?



REST endpoints allow for the creation of a REST interface to your saved GraphQL queries and mutations. Endpoints are accessible from `/api/rest/*` and inherit the authorization and permission structure from your associated GraphQL nodes. To create a new endpoint simply test your query in GraphiQL then click the REST button on GraphiQL to configure a URL. [\(Learn More\)](#)

Минусы

- Слишком много генерируют
- Дополнительные заботы о безопасности

Генераторы архитектуры

jhipster.tech



































Greetings, Java Hipster!

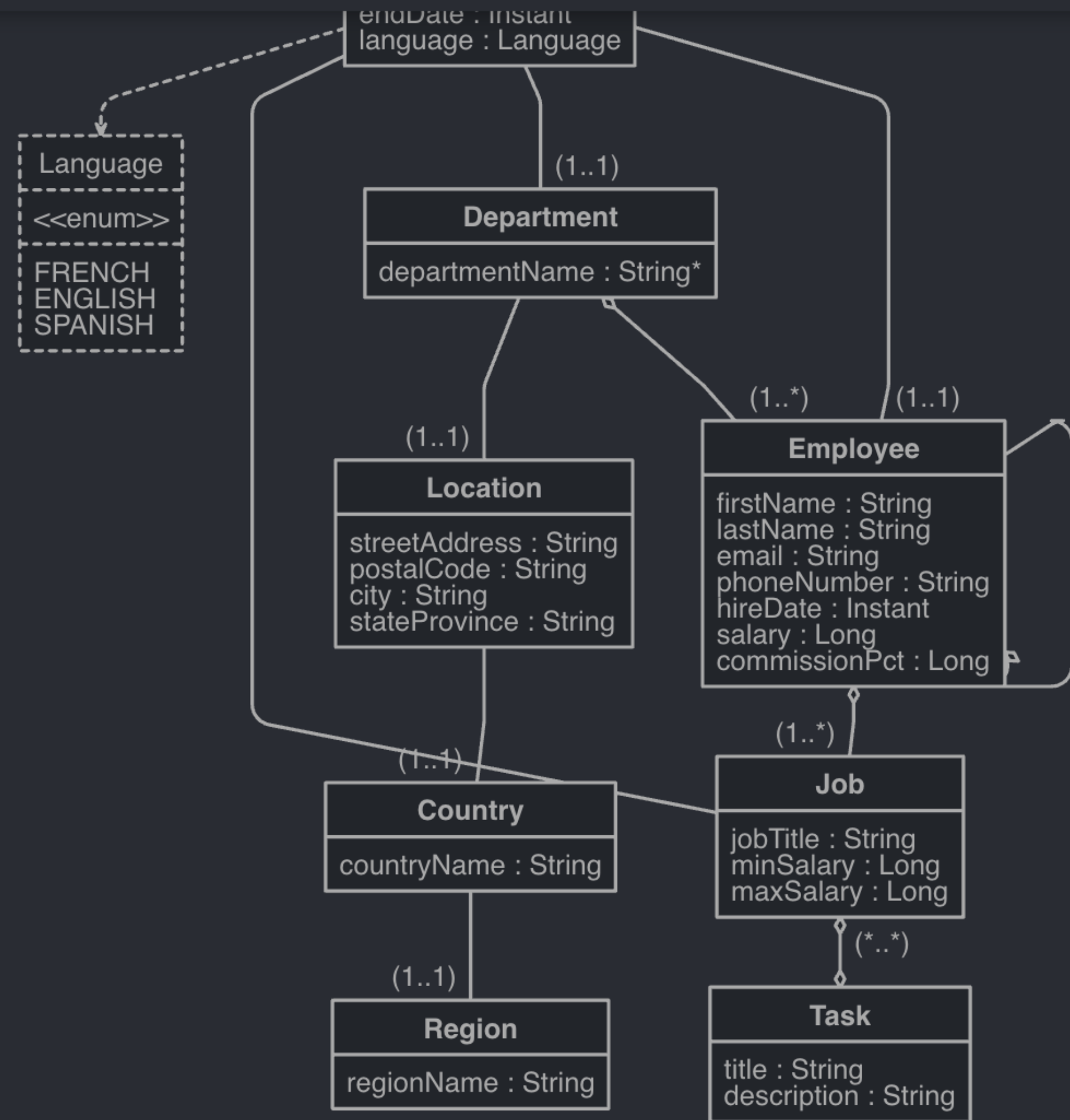
JHipster is a development platform to quickly generate, develop, & deploy modern web applications & microservice architectures.

JHipster is on [Open Collective](#), if you ❤️ JHipster consider becoming a [sponsor](#) or a [backer](#).

```

1
2 entity Region {
3   regionName String
4 }
5
6 entity Country {
7   countryName String
8 }
9
10 // an ignored comment
11 /** not an ignored comment */
12 entity Location {
13   streetAddress String
14   postalCode String
15   city String
16   stateProvince String
17 }
18
19 entity Department {
20   departmentName String required
21 }
22

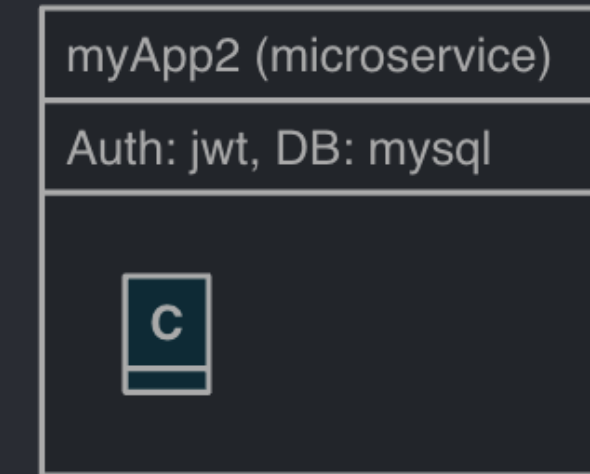
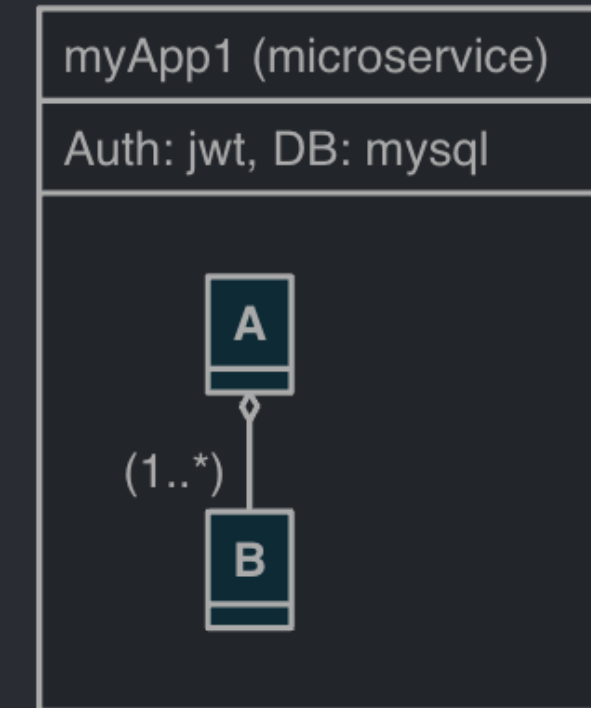
```




```

1
2 application {
3   config {
4     baseName myApp
5     applicationType gateway
6     packageName com.myapp
7     authenticationType jwt
8     prodDatabaseType mysql
9     clientFramework react
10  }
11  entities *
12 }
13
14 application {
15   config {
16     baseName myApp1
17     applicationType microservice
18     packageName com.myapp
19     authenticationType jwt
20     prodDatabaseType mysql
21  }
22  entities A, B

```



8080

Which *type* of authentication would you like to use?

JWT authentication (stateless, with a token) ▾

Which *type* of database would you like to use?

SQL (H2, MySQL, MariaDB, PostgreSQL, Oracle, MSSQL) ▾

Which *production* database would you like to use?

PostgreSQL ▾

Which *development* database would you like to use?

H2 with disk-based persistence ▾

Do you want to use the Spring cache abstraction?

Yes, with Redis (single server) ▾

Do you want to use Hibernate 2nd level cache?

Yes ▾

Would you like to use Maven or Gradle for building the backend?

Maven ▾

Which other technologies would you like to use?

API first development using OpenAPI-generator

Client side options

Which *Framework* would you like to use for the client?

React



Do you want to generate the admin UI?

Yes



Internationalization options

Would you like to enable internationalization support?

Yes



Please choose the native language of the application

English



Please choose additional languages to install

Bengali
Catalan
Chinese (Simplified)
Chinese (Traditional)
Czech



Testing options

Генерируем бэкенд

Основные инструменты



20 инструментов для GraphQL

- Кодген
- Плагины
- Прокси сервера
- и многое другое

the-guild.dev

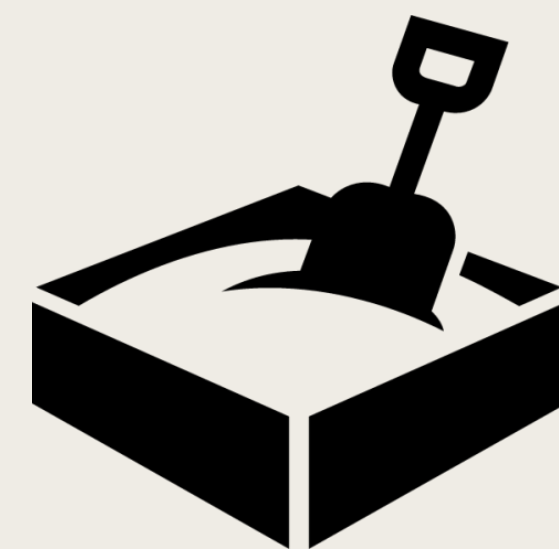
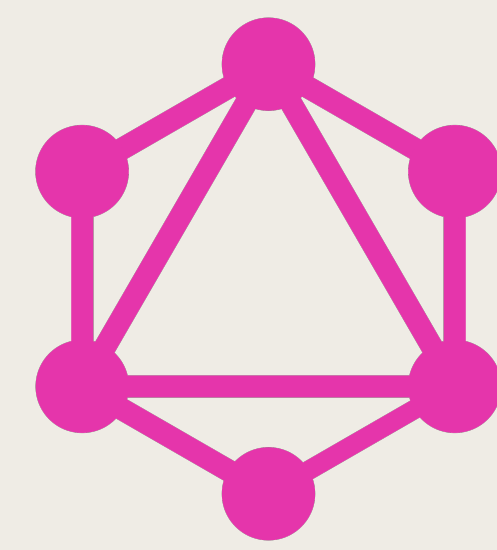


Доступных генераторов

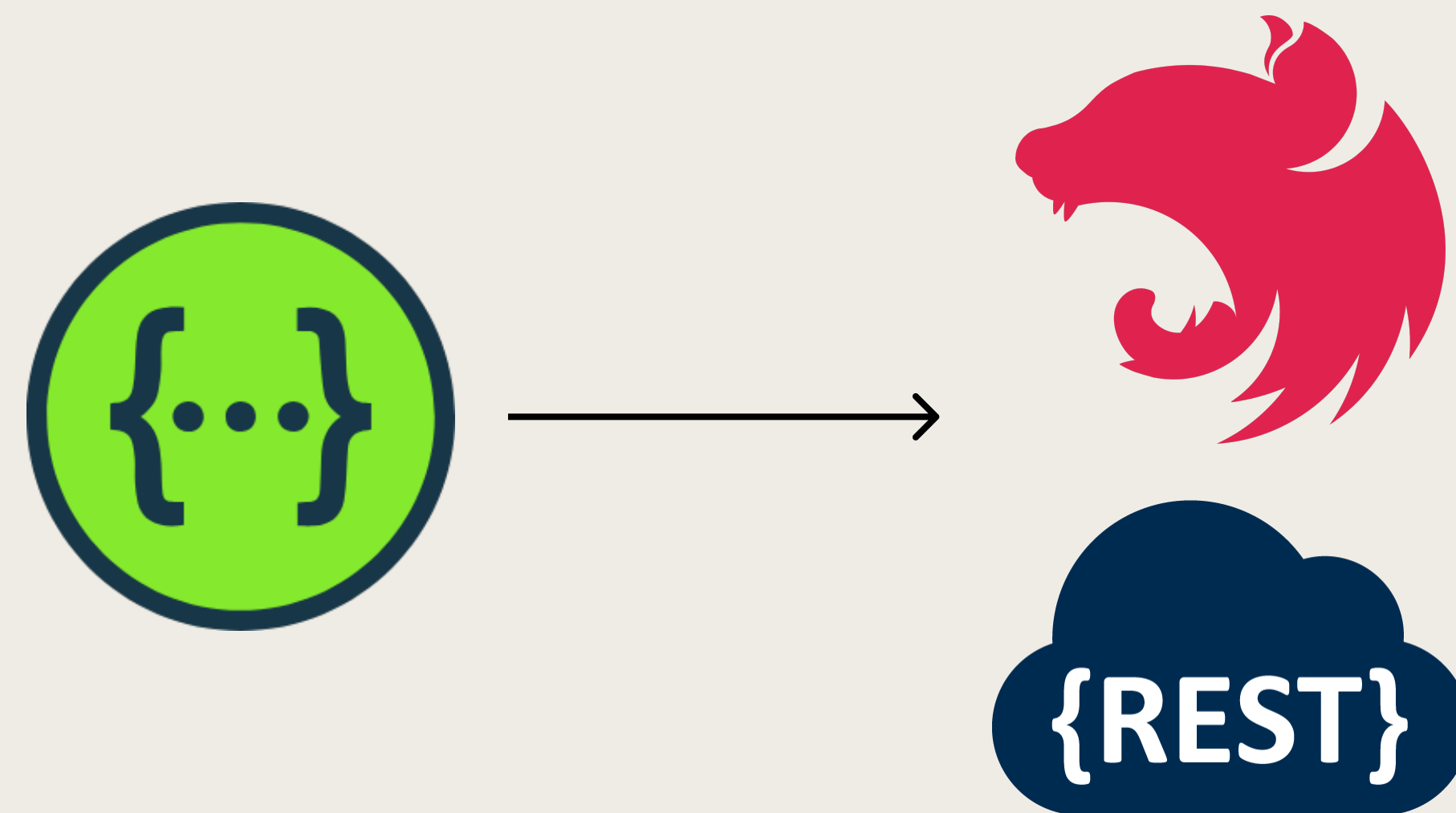
- 71 для клиента
- 56 для сервера
- 9 для документации
- 7 для схем

openapi-generator.tech

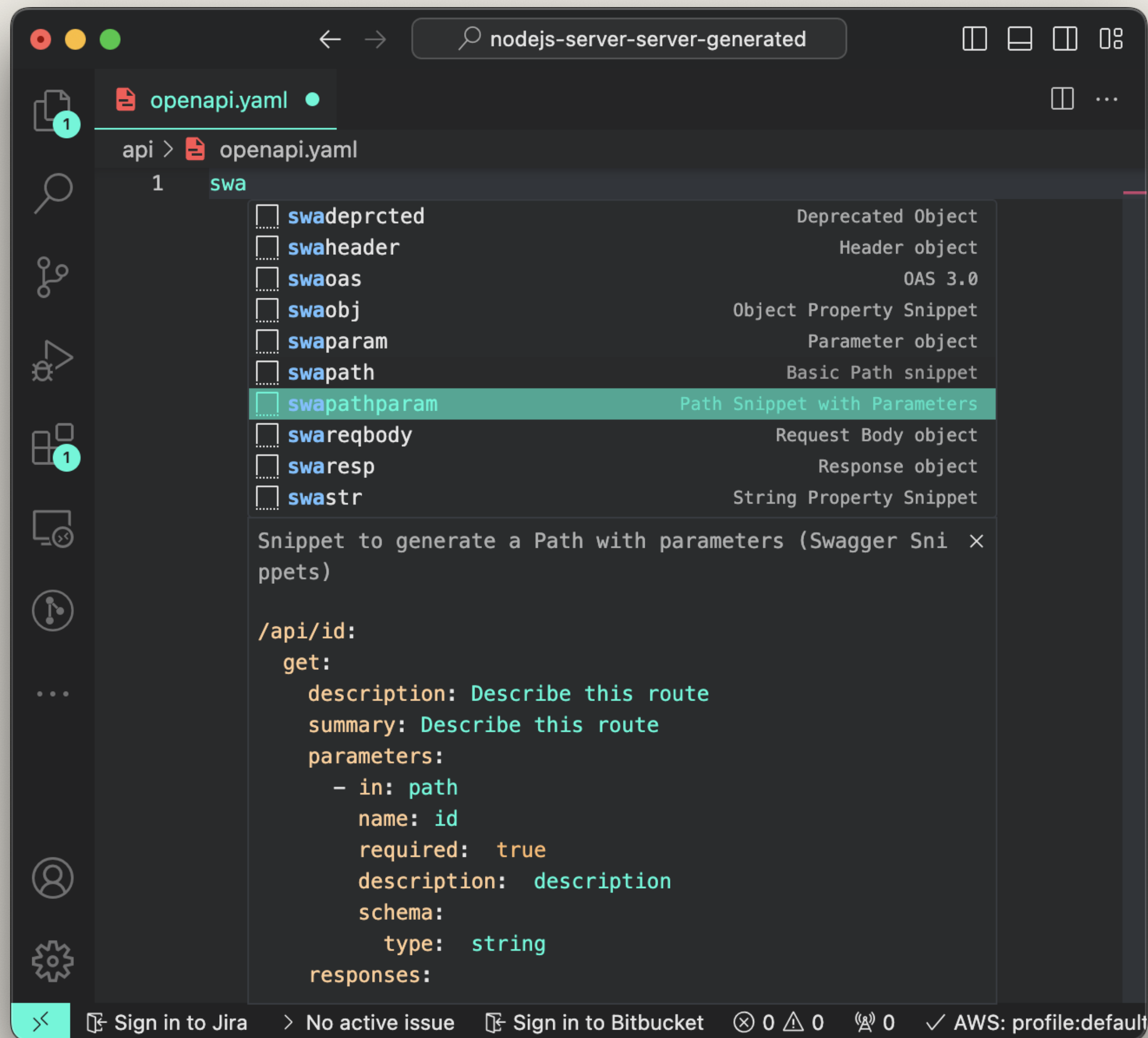
Что будем генерировать здесь?



Кейс: прототипируем сервис



Пишем Swagger и используем снипеты



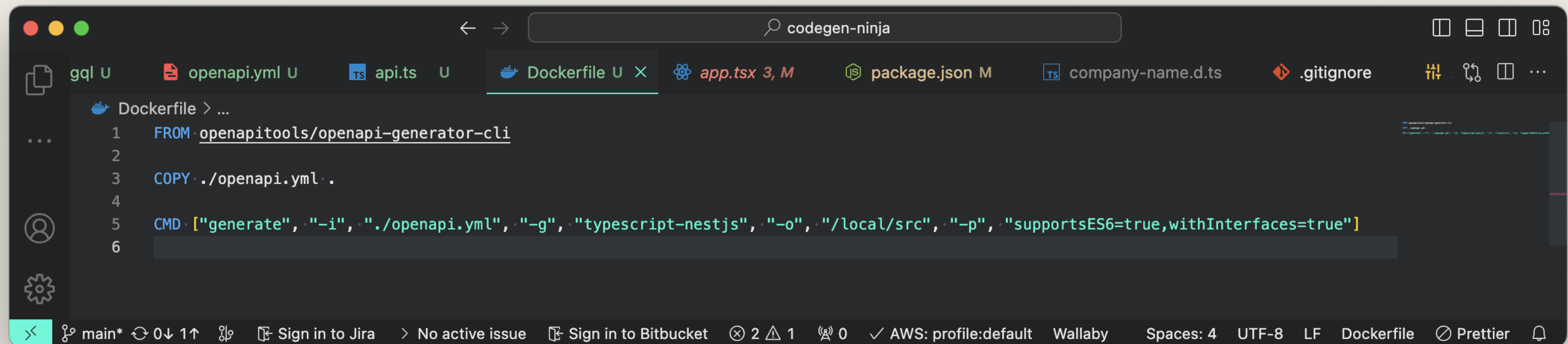
Получаем схему

The image shows a code editor window with the following elements:

- Search Bar:** Contains the text "codegen-ninja".
- File Explorer:** Shows a breadcrumb trail: `openapi.yml > {} components > {} securitySchemes > {} api_key > abc in`.
- Main Editor:** Displays the content of `openapi.yml`, which is an OpenAPI 3.0 schema for Swagger Petstore. The content includes:

```
1 openapi: 3.0.3
2 info:
3   title: Swagger Petstore - OpenAPI 3.0
4   description: |-
5     This is a sample Pet Store Server based on the OpenAPI 3.0 specification. You
6     Swagger at [https://swagger.io] (https://swagger.io). In the third iteration o
7     You can now help us improve the API whether it's by making changes to the def
8     That way, with time, we can improve the API in general, and expose some of th
9
10    _If you're looking for the Swagger 2.0/OAS 2.0 version of Petstore, then clic
11
12    Some useful links:
13    - [The Pet Store repository] (https://github.com/swagger-api/swagger-petstore)
14    - [The source API definition for the Pet Store] (https://github.com/swagger-ap
15  termsOfService: http://swagger.io/terms/
16  contact:
17    email: apiteam@swagger.io
18  license:
19    name: Apache 2.0
20    url: http://www.apache.org/licenses/LICENSE-2.0.html
21  version: 1.0.11
22 externalDocs:
23   description: Find out more about Swagger
24   url: http://swagger.io
25 servers:
26   - url: https://petstore3.swagger.io/api/v3
27 tags:
```
- Right Panel:** Shows a preview of the OpenAPI schema as a JSON object.
- Bottom Bar:** Contains navigation icons and status information: `main*`, `0↓1↑`, `Sign in to Jira`, `No active issue`, `Sign in to Bitbucket`, `0`, `0`, `0`, and `AWS: profile:default`.

Генерируем NestJS Service



The screenshot shows a code editor window with a dark theme. The browser address bar at the top displays "codegen-ninja". The editor's tab bar includes "gql U", "openapi.yml U", "api.ts U", "Dockerfile U X", "app.tsx 3, M", "package.json M", "company-name.d.ts", and ".gitignore". The active file is "Dockerfile", which contains the following content:

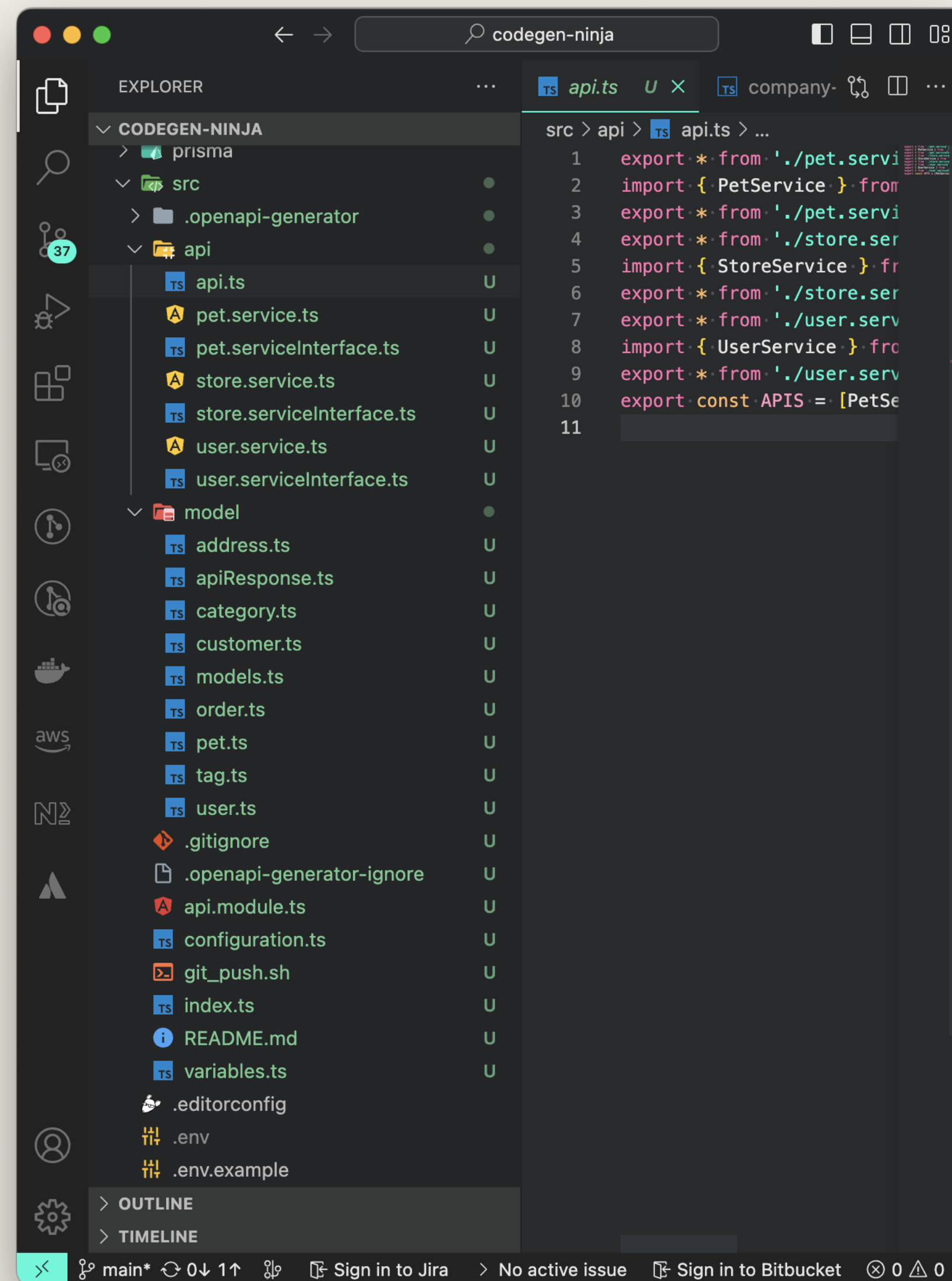
```
Dockerfile > ...
1 FROM openapitools/openapi-generator-cli
2
3 COPY ./openapi.yml .
4
5 CMD ["generate", "-i", "./openapi.yml", "-g", "typescript-nestjs", "-o", "/local/src", "-p", "supportsES6=true,withInterfaces=true"]
6
```

The bottom status bar of the editor shows the following information: "main*" (with refresh and up/down arrows), "Sign in to Jira", "No active issue", "Sign in to Bitbucket", "2 errors, 1 warning, 0 info", "AWS: profile:default", "Wallaby", "Spaces: 4", "UTF-8", "LF", "Dockerfile", "Prettier", and a notification bell icon.

Генерируем NestJS Service

```
vmaliutin@Vadims-MacBook-Pro-2:~/Projects/Personal/codegen-ninja 1
✨ Done in 6.29s.
> yarn codegen
yarn run v1.22.19
$ docker build . -t codegen-service && docker run --rm -v $PWD:/local codegen-service && tsc
[+] Building 7.5s (5/5) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 218B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/openapitools/openapi-generator-cli:latest 2.0s
=> [1/1] FROM docker.io/openapitools/openapi-generator-cli@sha256:aa8b02c8cdaef08f9478d967a5ea2640e94e6d63 5.3s
=> => resolve docker.io/openapitools/openapi-generator-cli@sha256:aa8b02c8cdaef08f9478d967a5ea2640e94e6d63 0.0s
=> => sha256:cd4bbffc676138346e63cd4806a5da2691f8841609da893e7fa25f6fb4cb5216 5.78kB / 5.78kB 0.0s
=> => sha256:71209d5eb534b9e48223962276993c68559f68e230f73c8a0efc2a2998362bd9 49.23MB / 49.23MB 1.7s
=> => sha256:e31ca5ccee8fca6610f14b5ed35ac33bb5f545532b6583e1461037a083c3d87b 7.72MB / 7.72MB 0.4s
=> => sha256:32de8e1f96ccb825d3be85704be7218044a19ff05ea1eea0222e8c942fbf6f8f 9.77MB / 9.77MB 0.3s
=> => sha256:aa8b02c8cdaef08f9478d967a5ea2640e94e6d63f34892ec72ef9458cf7ed305 685B / 685B 0.0s
=> => sha256:7c8360fc942623d2ad4800934ee9e28442679b1923609217c86be146717bec66 1.89kB / 1.89kB 0.0s
=> => sha256:55122ede21178db0d7b4344f167dcbda7d103c4b77dfa15b37411fc005e009c8 5.51MB / 5.51MB 0.8s
=> => sha256:82da056785d1770185dc9c78ca7d278ef3051b0651306f888fc1f8d0129dbef6 211B / 211B 0.6s
=> => sha256:2bc66ba0403fd0d18ced4a3b1b068ff3a161d6acb9b8e4327f007e1e448cb7b8 45.07MB / 45.07MB 2.0s
=> => sha256:07a4472344f9b38c836fcda3b77b9ecba12c8331173dfd5644f0e528c27f5a3b 23.85MB / 23.85MB 1.8s
=> => sha256:ac6c26eaa59778810990f4f328bc1a71eee2661d71a828cc7a2eb593118e58cc 1.12kB / 1.12kB 2.0s
=> => extracting sha256:71209d5eb534b9e48223962276993c68559f68e230f73c8a0efc2a2998362bd9 1.3s
=> => extracting sha256:e31ca5ccee8fca6610f14b5ed35ac33bb5f545532b6583e1461037a083c3d87b 0.2s
=> => extracting sha256:32de8e1f96ccb825d3be85704be7218044a19ff05ea1eea0222e8c942fbf6f8f 0.1s
=> => extracting sha256:55122ede21178db0d7b4344f167dcbda7d103c4b77dfa15b37411fc005e009c8 0.1s
=> => extracting sha256:82da056785d1770185dc9c78ca7d278ef3051b0651306f888fc1f8d0129dbef6 0.0s
```


Полученная структура



codegen-ninja

EXPLORER

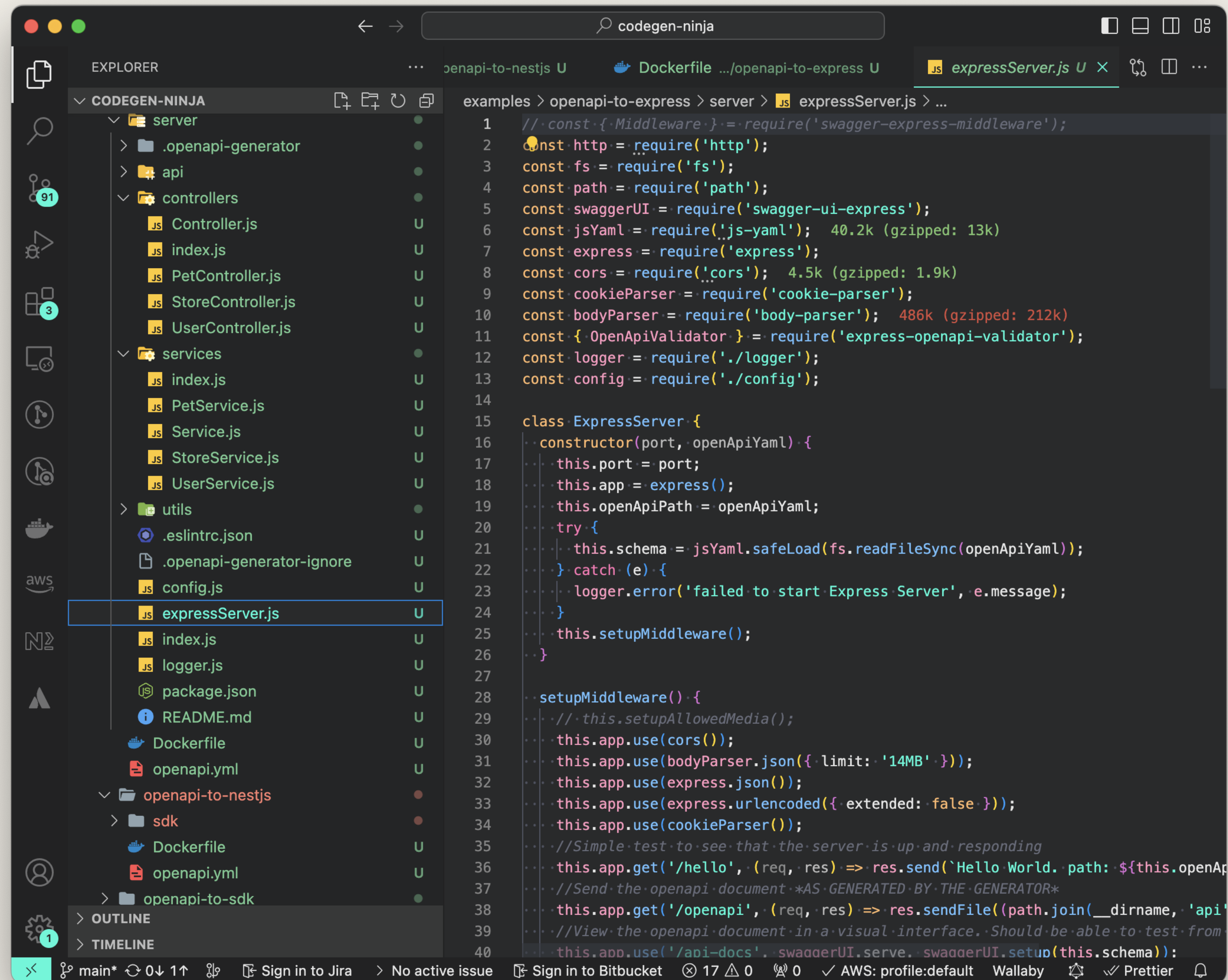
- CODEGEN-NINJA
 - node_modules
 - prisma
 - src
 - .openapi-generator
 - api
 - api.ts
 - pet.service.ts
 - pet.serviceInterfa...
 - store.service.ts 9+, U
 - store.serviceInter...
 - user.service.ts
 - user.serviceInterf...
 - model
 - address.ts
 - apiResponse.ts
 - category.ts
 - customer.ts
 - models.ts
 - order.ts
 - pet.ts
 - tag.ts
 - users.ts

src > api > store.service.ts > ...

```
76 .....);
77 .....}
78 ...../**
79 .....*/ Returns pet inventories by status
80 .....*/ Returns a map of status codes to quantities
81 .....*/ @param observe set whether or not to return the data Observable as the body, response or events. defaults to returning the body.
82 .....*/ @param reportProgress flag to report request and response progress.
83 .....*/
84 .....public getInventory(): Observable<AxiosResponse<{ [key: string]: number; }>>;
85 .....public getInventory(): Observable<any> {
86 .....    Object is possibly 'undefined'. ts(2532)
87 .....    let
88 .....        (property) StoreService.configuration: Configuration
89 .....        // a View Problem (⌘F8) No quick fixes available
90 .....        if (this.configuration.apiKeys["api_key"]) {
91 .....            headers['api_key'] = this.configuration.apiKeys["api_key"];
92 .....        }
93 .....
94 .....        // to determine the Accept header
95 .....        let httpHeaderAccepts: string[] = [
96 .....            'application/json'
97 .....        ];
98 .....        const httpHeaderAcceptSelected: string | undefined = this.configuration.selectHeaderAccept(httpHeaderAccepts);
99 .....        if (httpHeaderAcceptSelected != undefined) {
100 .....            headers['Accept'] = httpHeaderAcceptSelected;
101 .....        }
102 .....
103 .....        // to determine the Content-Type header
104 .....        const consumes: string[] = [
```

main* 0↓ 1↑ Sign in to Jira > No active issue Sign in to Bitbucket 11 △ 0 0 AWS: profile:default Wallaby Spaces: 4 UTF-8 LF {} TypeScript Prettier

А как же API?



The image shows a screenshot of the Visual Studio Code editor. The Explorer sidebar on the left displays a project structure for 'CODEGEN-NINJA'. The main editor window shows the code for 'expressServer.js'.

```
examples > openapi-to-express > server > JS expressServer.js > ...
1 // const { Middleware } = require('swagger-express-middleware');
2 const http = require('http');
3 const fs = require('fs');
4 const path = require('path');
5 const swaggerUI = require('swagger-ui-express');
6 const jsYaml = require('js-yaml'); 40.2k (gzipped: 13k)
7 const express = require('express');
8 const cors = require('cors'); 4.5k (gzipped: 1.9k)
9 const cookieParser = require('cookie-parser');
10 const bodyParser = require('body-parser'); 486k (gzipped: 212k)
11 const { OpenApiValidator } = require('express-openapi-validator');
12 const logger = require('./logger');
13 const config = require('./config');
14
15 class ExpressServer {
16   constructor(port, openApiYaml) {
17     this.port = port;
18     this.app = express();
19     this.openApiPath = openApiYaml;
20     try {
21       this.schema = jsYaml.safeLoad(fs.readFileSync(openApiYaml));
22     } catch (e) {
23       logger.error('failed to start Express Server', e.message);
24     }
25     this.setupMiddleware();
26   }
27
28   setupMiddleware() {
29     // this.setupAllowedMedia();
30     this.app.use(cors());
31     this.app.use(bodyParser.json({ limit: '14MB' }));
32     this.app.use(express.json());
33     this.app.use(express.urlencoded({ extended: false }));
34     this.app.use(cookieParser());
35     // Simple test to see that the server is up and responding
36     this.app.get('/hello', (req, res) => res.send(`Hello World. path: ${this.openApiPath}`));
37     // Send the openapi document *AS GENERATED BY THE GENERATOR*
38     this.app.get('/openapi', (req, res) => res.sendFile(path.join(__dirname, 'api-docs', 'openapi.yml')));
39     // View the openapi document in a visual interface. Should be able to test from
40     this.app.use('/api-docs', swaggerUI.serve, swaggerUI.setup(this.schema));
```


Быстрый способ поиграться editor.swagger.io

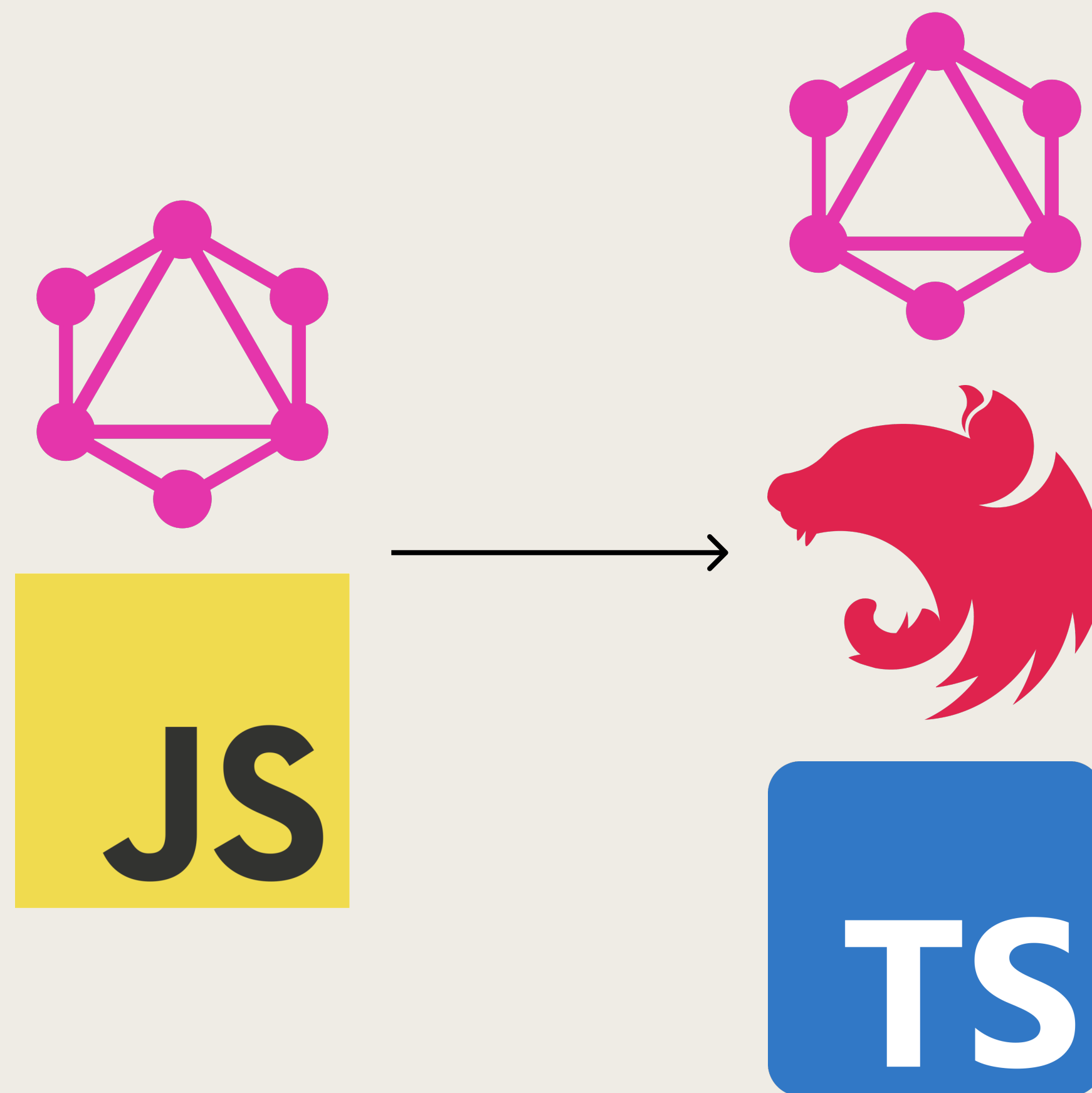
The screenshot displays the Swagger Editor web application. On the left, a code editor shows the OpenAPI 3.0.3 definition for a Petstore API. The code includes an 'info' section with a title, description, and version, and a 'servers' section. On the right, a dropdown menu is open, listing various server frameworks such as 'aspnetcore', 'go-server', 'inflector', 'java-vertx', 'jaxrs-cxf', 'jaxrs-cxf-cdi', 'jaxrs-di', 'jaxrs-jersey', 'jaxrs-resteasy', 'jaxrs-resteasy-eap', and 'jaxrs-spec'. Below the dropdown, there are links for 'Apache 2.0' and 'Find out more about Swagger'. At the bottom right, there is a 'Servers' dropdown menu with 'https://petstore3.swagger.io/api/v3' selected and an 'Authorize' button with a lock icon.

```
1 openapi: 3.0.3
2 info:
3   title: Swagger Petstore - OpenAPI 3.0
4   description: |-
5     This is a sample Pet Store Server based on the
6     OpenAPI 3.0 specification. You can see the API
7     Swagger at [https://swagger.io](https://swagger.io)
8     Swagger at [https://swagger.io](https://swagger.io)
9     Swagger at [https://swagger.io](https://swagger.io)
10    Swagger at [https://swagger.io](https://swagger.io)
11    Swagger at [https://swagger.io](https://swagger.io)
12    Swagger at [https://swagger.io](https://swagger.io)
13    Swagger at [https://swagger.io](https://swagger.io)
14    Swagger at [https://swagger.io](https://swagger.io)
15    Swagger at [https://swagger.io](https://swagger.io)
16    Swagger at [https://swagger.io](https://swagger.io)
17    Swagger at [https://swagger.io](https://swagger.io)
18    Swagger at [https://swagger.io](https://swagger.io)
19    Swagger at [https://swagger.io](https://swagger.io)
20    Swagger at [https://swagger.io](https://swagger.io)
21    Swagger at [https://swagger.io](https://swagger.io)
22    Swagger at [https://swagger.io](https://swagger.io)
23    Swagger at [https://swagger.io](https://swagger.io)
24    Swagger at [https://swagger.io](https://swagger.io)
25    Swagger at [https://swagger.io](https://swagger.io)
```

Вывод

у OpenAPI куда развиваться

Кейс: мигрируем с JS на TS





Schema

Reference

SDL

API schema

90 types 595 fields

<https://spacex-production.up.railway.app/>

Oct 15, 2021 at 6:32 AM GMT+1

```

1132
1133     """
1134     semi_major_axis_km: Float
1135   }
1136
1137   type Query {
1138     """
1139     capsule(id: ID!): Capsule
1140
1141     """
1142     capsules(find: CapsulesFind, limit: Int, offset:
1143     Int, order: String, sort: String): [Capsule]
1144
1145     """
1146     capsulesPast(find: CapsulesFind, limit: Int,
1147     offset: Int, order: String, sort: String): [Capsule]

```

Copy Raw



codegen-ninja

```
1127 .....
1128   · regime: String
1129 .....
1130 .....
1131   · reused: Boolean
1132 .....
1133 .....
1134   · semi_major_axis_km: Float
1135 }
1136
1137 type Query {
1138 .....
1139   · capsule(id: ID!): Capsule
1140 .....
1141 .....
1142   · capsules(find: CapsulesFind,
1143 .....
1144 .....
1145   · capsulesPast(find: CapsulesFir
1146 .....
1147 .....
1148   · capsulesUpcoming(find: Capsule
1149 .....
1150 .....
1151   · company: Info
1152 .....
1153 .....
1154   · core(id: ID!): Core
1155 .....
1156 .....

1 generates:
2   · types.ts:
3   · · · plugins:
4   · · · · - typescript-type-graphql

import * as TypeGraphQL from 'type-graph Untitled-3 9+
1222   · period_min?: Maybe<Scalars['Float']>;
1223
1224   · @TypeGraphQL.Field(type => TypeGraphQL.Float, { nullable: true })
1225   · raan?: Maybe<Scalars['Float']>;
1226
1227   · @TypeGraphQL.Field(type => String, { nullable: true })
1228   · reference_system?: Maybe<Scalars['String']>;
1229
1230   · @TypeGraphQL.Field(type => String, { nullable: true })
1231   · regime?: Maybe<Scalars['String']>;
1232
1233   · @TypeGraphQL.Field(type => Boolean, { nullable: true })
1234   · reused?: Maybe<Scalars['Boolean']>;
1235
1236   · @TypeGraphQL.Field(type => TypeGraphQL.Float, { nullable: true })
1237   · semi_major_axis_km?: Maybe<Scalars['Float']>;
1238 };
1239
1240 export type Query = {
1241   · __typename?: 'Query';
1242   · capsule?: Maybe<Capsule>;
1243   · capsules?: Maybe<Array<Maybe<Capsule>>>;
1244   · capsulesPast?: Maybe<Array<Maybe<Capsule>>>;
1245   · capsulesUpcoming?: Maybe<Array<Maybe<Capsule>>>;
1246   · company?: Maybe<Info>;
1247   · core?: Maybe<Core>;
1248   · cores?: Maybe<Array<Maybe<Core>>>;
1249   · coresPast?: Maybe<Array<Maybe<Core>>>;
1250   · coresUpcoming?: Maybe<Array<Maybe<Core>>>;
1251   · dragon?: Maybe<Dragon>;
```

main* 0↓ 1↑ Sign in to Jira > No active issue Sign in to Bitbucket 65 0 1 0 AWS: profile:default Wallaby Spaces: 2 UTF-8 LF GraphQL Prettier


Комментарии

- Конвертирует не прям в NestJS, а в type-graphql
- Некоторые типы не понимает и возвращает any, что генерирует невалидный GQL
- Тоже можно использовать для прототипирования


Как фиксировать?

- Написать плагин
- Ручками поменять один раз, если не нужно регулярно делать похожее

Что еще умеет?




GraphQL Code Generator
Generation of typed queries, mutations, subscriptions and typed GraphQL resolvers

[Docs](#) [Plugins](#) 

Explore Plugins & Presets

preset plugin typescript csharp dart flutter flow java utilities mongodb nhost angular react svelte next apollo urql vue kotlin android
reason relay jsdoc resolvers hasura validation yup zod




Trending

NAME	LAST UPDATE
------	-------------

Recently Updated

NAME	LAST UPDATE
------	-------------



Пишем логику

Default (GPT-3.5)

hey gpt, I want to write backend service using nestjs for ninja online school. Could you write logic for my endpoints?

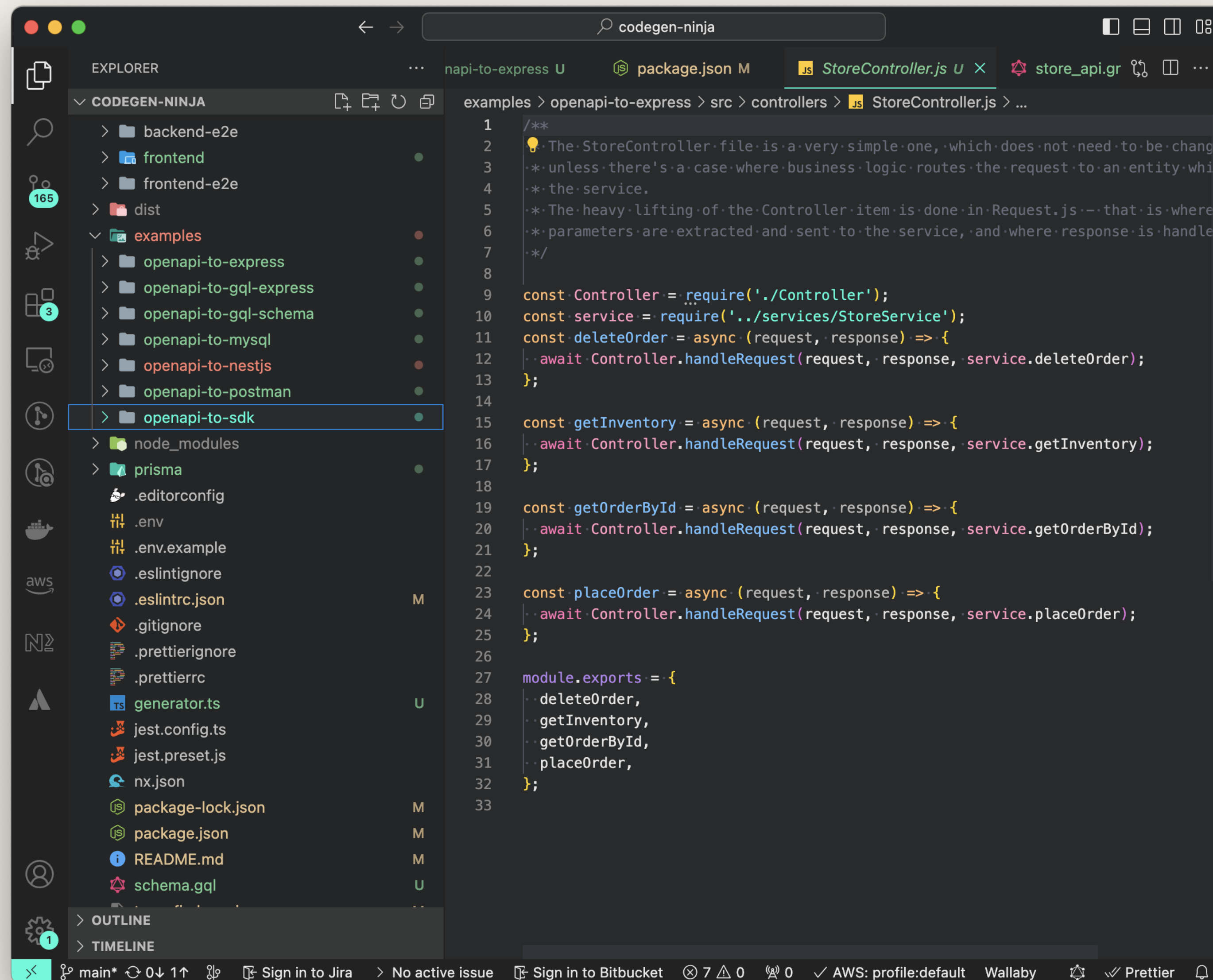
Of course, I can help you get started with writing the logic for your backend service using NestJS for a Ninja online school. NestJS is a popular framework for building scalable and maintainable server-side applications. Below, I'll provide you with some example endpoints and their corresponding logic to give you a starting point. You can expand upon this foundation to meet the specific requirements of your project.

Before you start, make sure you have NestJS installed and set up in your project.

Send a message

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT September 25 Version](#)

Больше примеров



Генерируем
бэкенд документацию

Swagger OpenAPI

The screenshot displays the Swagger Editor interface for a pet API. The top navigation bar includes the Swagger Editor logo, a menu with 'File', 'Edit', and 'Insert', and buttons for 'Generate Server', 'Generate Client', and 'About'. A green button on the right says 'Try our new Editor'. The main content area is titled 'pet Everything about your Pets' with a 'Find out more' link. Two endpoints are listed: a PUT endpoint for updating a pet and a POST endpoint for adding a new pet. The POST endpoint is selected and expanded, showing its description 'Add a new pet to the store', a 'Parameters' section with 'No parameters', and a 'Request body' section with 'application/json' selected. Below this, there is an 'Example Value' section with a JSON object: { "id": 10, "name": "doggie", "category": { "id": 1, "name": "Dogs" }, "photoUrls": [...] }.

Swagger Editor
Supported by SMARTBEAR

File Edit Insert Generate Server Generate Client About Try our new Editor

pet Everything about your Pets Find out more

PUT /pet Update an existing pet

POST /pet Add a new pet to the store

Add a new pet to the store

Parameters Try it out

No parameters

Request body required application/json

Create a new pet in the store

Example Value Schema

```
{
  "id": 10,
  "name": "doggie",
  "category": {
    "id": 1,
    "name": "Dogs"
  },
  "photoUrls": [
```

GQL Playground

The screenshot displays the GraphQL Playground interface. At the top, there is a tab labeled 'cat' and a plus sign for adding new tabs. The address bar shows 'http://localhost:3000/graphql'. Below the address bar are buttons for 'PRETTIFY', 'HISTORY', 'COPY CURL', and 'SHARE PLAYGROUND'. The main area is split into two panes. The left pane contains a GraphQL query:

```
1 {  
2   cat(id: 1) {  
3     id,  
4     name  
5   }  
6 }  
7
```

The right pane shows the JSON response:

```
{  
  "data": {  
    "cat": {  
      "id": 1,  
      "name": "Cat"  
    }  
  }  
}
```

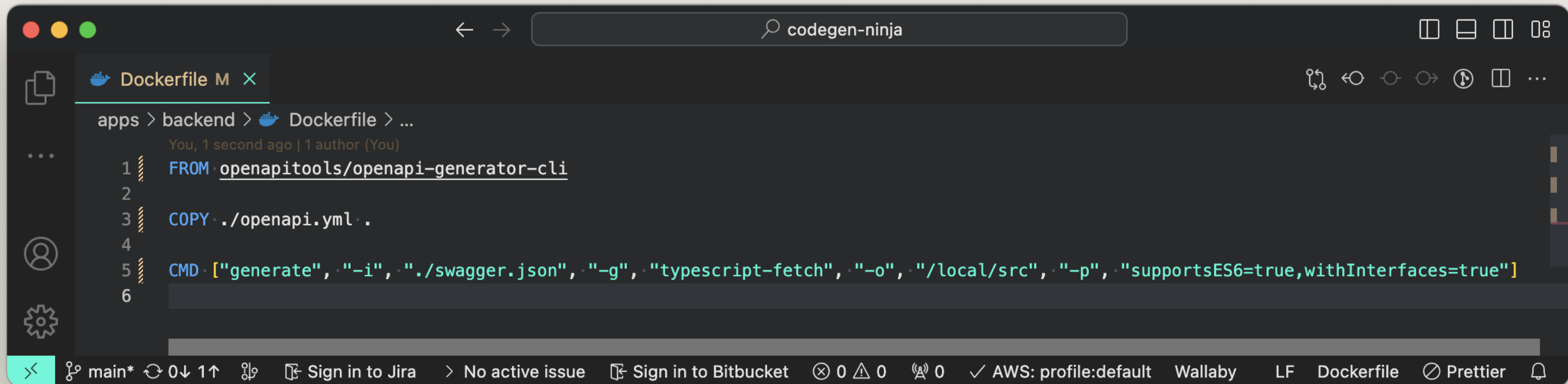
At the bottom of the interface, there are sections for 'QUERY VARIABLES' and 'HTTP HEADERS', both currently empty. A 'SCHEMA' button is visible on the right side of the response pane.

Комментарии

- Самим писать декораторы / комментарии
- Swagger чуть сложнее настройка, чем в GraphQL

Документация - клево
А как еще лучше?

Генерируем SDK



The image shows a code editor window with a dark theme. The title bar at the top contains window control buttons (red, yellow, green) on the left, a search bar with the text "codegen-ninja" in the center, and window management icons (maximize, close, etc.) on the right. The editor's tab bar shows a single tab titled "Dockerfile M" with a close button. The breadcrumb navigation indicates the current file path: "apps > backend > Dockerfile > ...". The main editing area displays the following Dockerfile content:

```
1 FROM openapitools/openapi-generator-cli
2
3 COPY ./openapi.yml .
4
5 CMD ["generate", "-i", "./swagger.json", "-g", "typescript-fetch", "-o", "/local/src", "-p", "supportsES6=true,withInterfaces=true"]
6
```

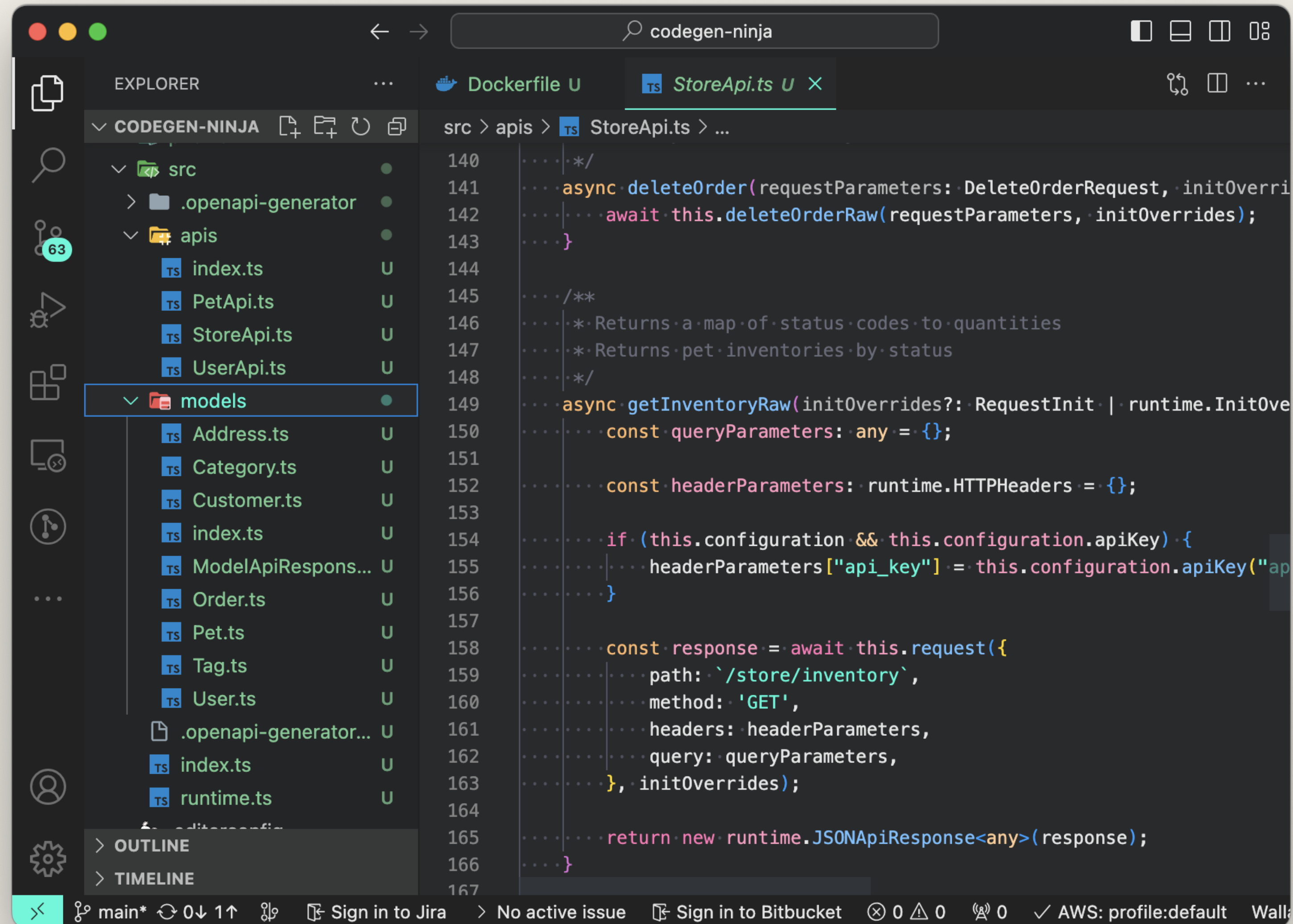
At the bottom of the editor, a status bar provides various information: a file explorer icon, the current branch "main*", commit statistics "0↓ 1↑", a search icon, "Sign in to Jira", "No active issue", "Sign in to Bitbucket", error and warning counts "0 0", a microphone icon, "AWS: profile:default", "Wallaby", "LF", "Dockerfile", "Prettier", and a notification bell icon.

Запустили докер

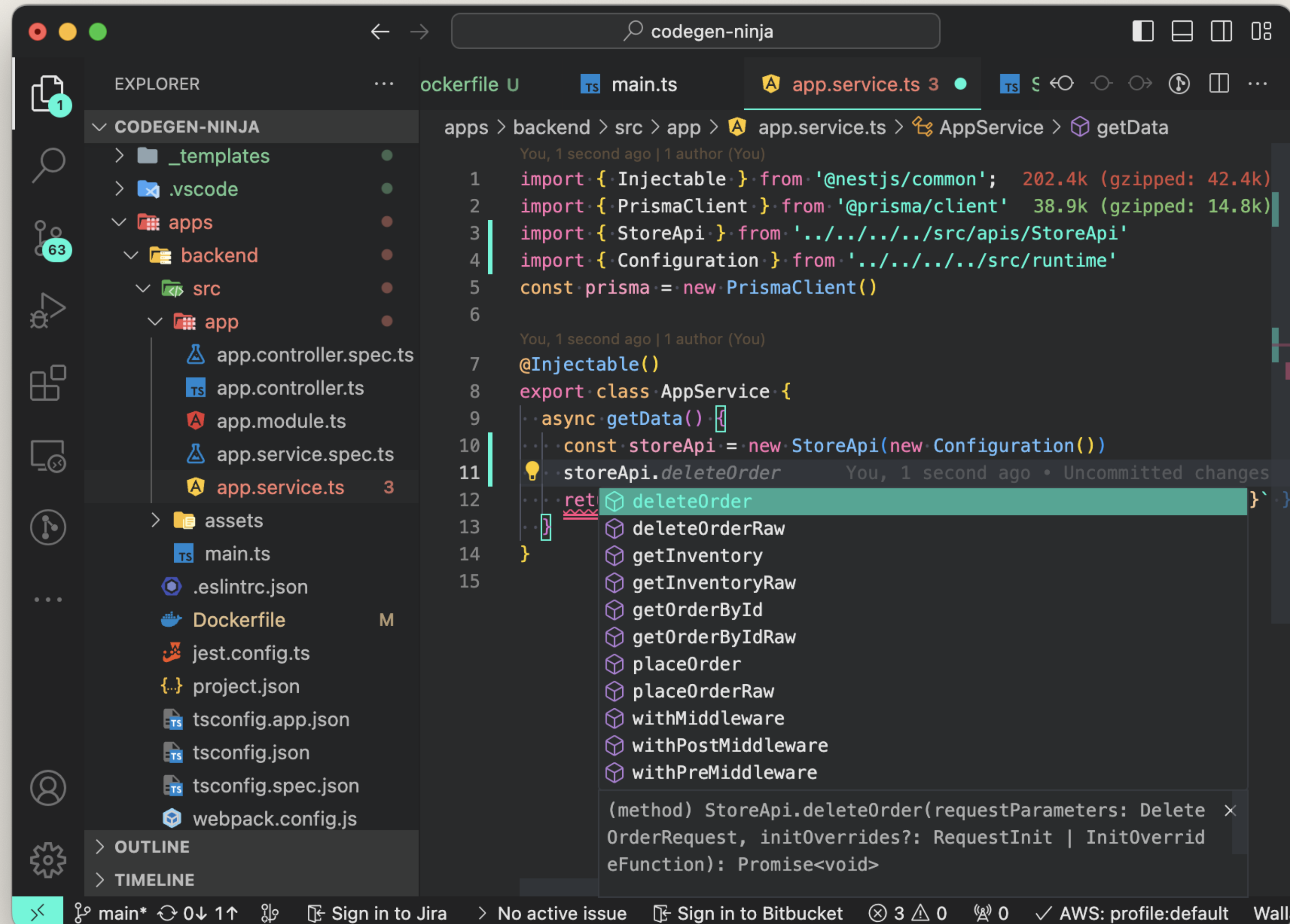
```
vmaliutin@Vadims-MacBook-Pro-2:~/Projects/Personal/codegen-ninja ㄿ#1
> yarn codegen
yarn run v1.22.19
$ docker build . -t codegen-service && docker run --rm -v $PWD:/local codegen-service
[+] Building 0.7s (7/7) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 37B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/openapitools/openapi-generator-cli:latest 0.6s
=> [internal] load build context 0.0s
=> => transferring context: 34B 0.0s
=> [1/2] FROM docker.io/openapitools/openapi-generator-cli@sha256:d8fd7b785278449f0ec7d3e 0.0s
=> CACHED [2/2] COPY ./openapi.yml . 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:82b5078588851dd7043cfa8feaa2edb776f6f748ed93534b8fd153923145 0.0s
=> => naming to docker.io/library/codegen-service 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
[main] INFO o.o.codegen.DefaultGenerator - Generating with dryRun=false
[main] INFO o.o.codegen.DefaultGenerator - OpenAPI Generator: typescript-fetch (client)
[main] INFO o.o.codegen.DefaultGenerator - Generator 'typescript-fetch' is considered stable.
[main] INFO o.o.c.l.AbstractTypeScriptClientCodegen - Hint: Environment variable 'TS_POST_PROCESSING_FILE' (optional) not defined. E.g. to format the source code, please try 'export TS_POST_PROCESSING_FILE="/usr/local/bin/prettier --write"' (Linux/Mac)
[main] INFO o.o.c.l.AbstractTypeScriptClientCodegen - Note: To enable file post-processing, 'enablePostProcessFile' must be set to `true` (--enable-post-process-file for CLI).
```

Получили код



Используем сгенеренный код



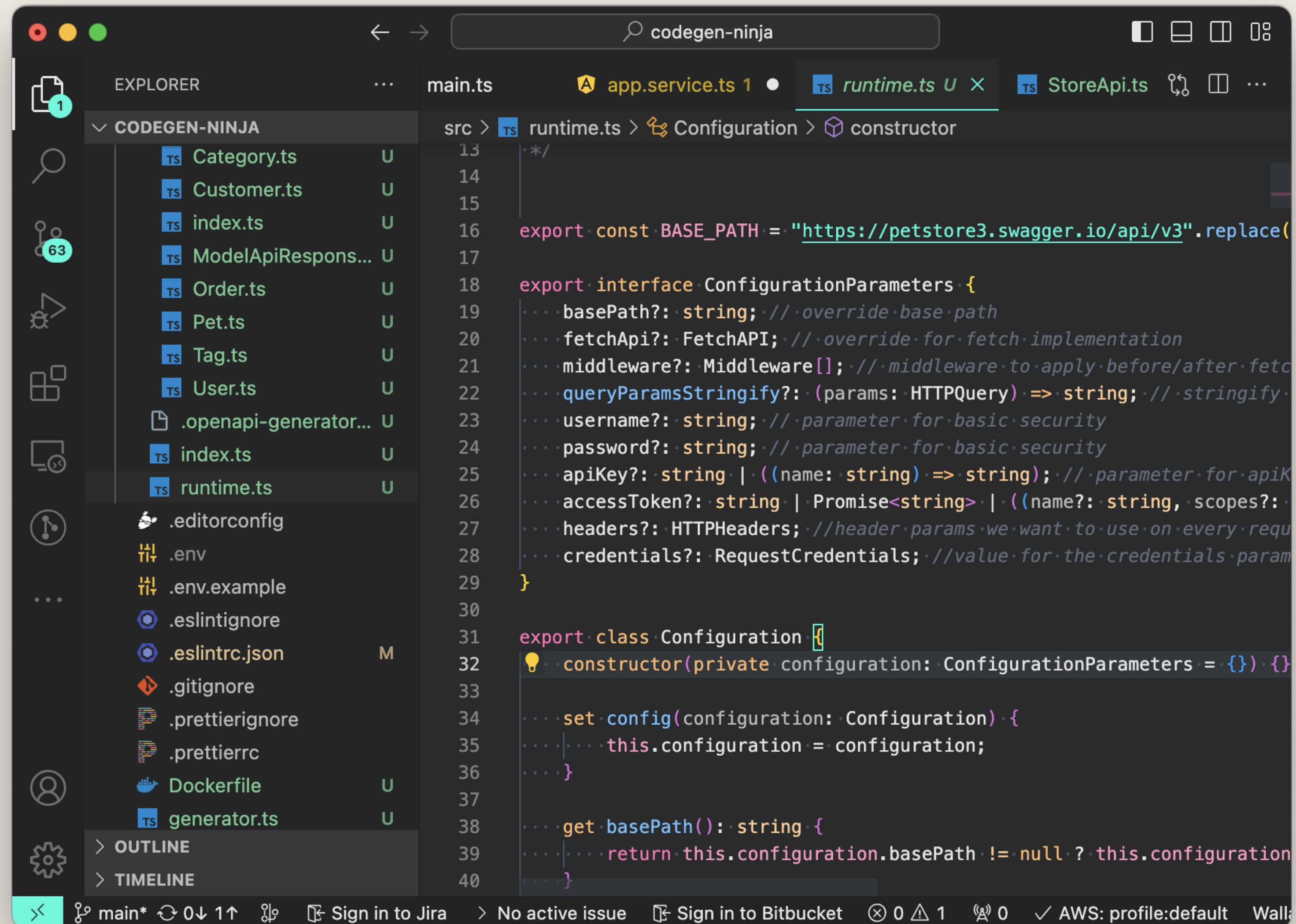
The image shows a screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows the project structure for 'CODEGEN-NINJA', with the file 'app.service.ts' selected. The main editor area displays the content of 'app.service.ts', which includes imports for '@nestjs/common', '@prisma/client', 'StoreApi', and 'Configuration'. A class 'AppService' is defined with an '@Injectable()' decorator and an 'async getData()' method. The 'getData()' method calls 'storeApi.deleteOrder()'. A dropdown menu is open over the 'deleteOrder' call, listing various methods from the 'StoreApi' interface, including 'deleteOrderRaw', 'getInventory', 'getInventoryRaw', 'getOrderById', 'getOrderByIdRaw', 'placeOrder', 'placeOrderRaw', 'withMiddleware', 'withPostMiddleware', and 'withPreMiddleware'. The status bar at the bottom shows 'main*' and 'AWS: profile:default'.

```
1 import { Injectable } from '@nestjs/common'; 202.4k (gzipped: 42.4k)
2 import { PrismaClient } from '@prisma/client' 38.9k (gzipped: 14.8k)
3 import { StoreApi } from '../../../../src/apis/StoreApi'
4 import { Configuration } from '../../../../src/runtime'
5 const prisma = new PrismaClient()
6
7 @Injectable()
8 export class AppService {
9   async getData() {
10     const storeApi = new StoreApi(new Configuration())
11     storeApi.deleteOrder
12     return deleteOrder
13   }
14 }
15
```

- deleteOrderRaw
- getInventory
- getInventoryRaw
- getOrderById
- getOrderByIdRaw
- placeOrder
- placeOrderRaw
- withMiddleware
- withPostMiddleware
- withPreMiddleware

(method) StoreApi.deleteOrder(requestParameters: DeleteOrderRequest, initOverrides?: RequestInit | InitOverrideFunction): Promise<void>

Кастомизируем поведение



The image shows a screenshot of the Visual Studio Code editor interface. The top bar displays the search bar with 'codegen-ninja' and window management icons. The Explorer sidebar on the left shows a file tree for 'CODEGEN-NINJA' with files like 'Category.ts', 'Customer.ts', 'index.ts', 'ModelApiResponse...', 'Order.ts', 'Pet.ts', 'Tag.ts', 'User.ts', and various configuration files. The main editor area shows the 'runtime.ts' file with the following TypeScript code:

```
src > runtime.ts > Configuration > constructor
13  */
14
15
16  export const BASE_PATH = "https://petstore3.swagger.io/api/v3".replace(
17
18  export interface ConfigurationParameters {
19    basePath?: string; // override base path
20    fetchApi?: FetchAPI; // override for fetch implementation
21    middleware?: Middleware[]; // middleware to apply before/after fetch
22    queryParamsStringify?: (params: HTTPQuery) => string; // stringify
23    username?: string; // parameter for basic security
24    password?: string; // parameter for basic security
25    apiKey?: string | ((name: string) => string); // parameter for apiK
26    accessToken?: string | Promise<string> | ((name?: string, scopes?:
27    headers?: HTTPHeaders; // header params we want to use on every requ
28    credentials?: RequestCredentials; // value for the credentials param
29  }
30
31  export class Configuration {
32    constructor(private configuration: ConfigurationParameters = {}) {}
33
34    set config(configuration: Configuration) {
35      this.configuration = configuration;
36    }
37
38    get basePath(): string {
39      return this.configuration.basePath != null ? this.configuration
40    }

```

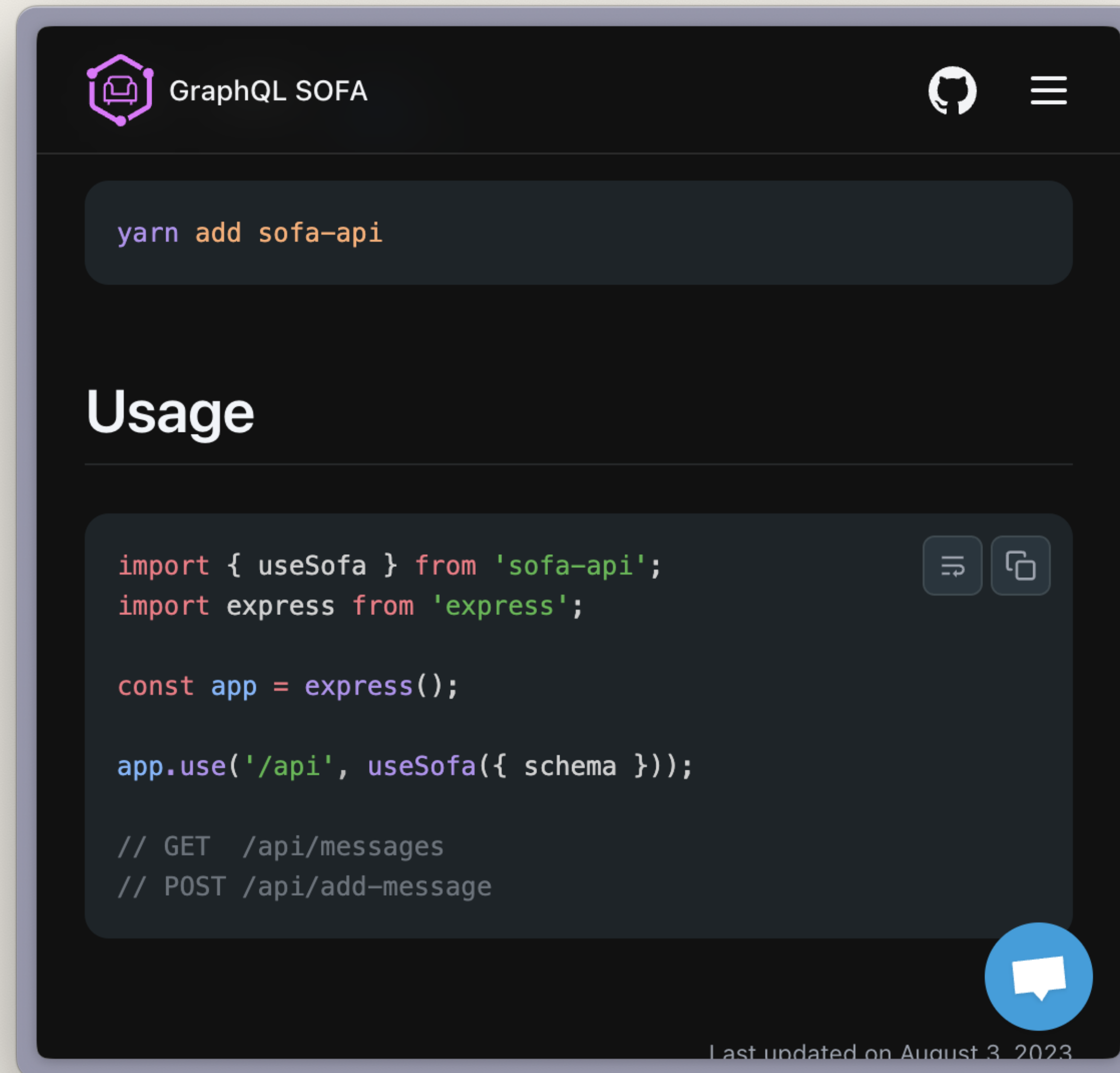
The status bar at the bottom shows the current file 'main.ts', a refresh icon, a download icon, an upload icon, and various utility icons like 'Sign in to Jira', 'No active issue', 'Sign in to Bitbucket', and 'AWS: profile:default'.

Полученное SDK

- Типизировано
- Из коробки мапперы данных на типы и их валидация
- Возможность кастомизации
- Публишим в public / private npm registry
- Храним у себя в репе
- CI/CD на кодген при изменениях в бэкенде

Хочу другой
бэкенд протокол

Кейс: GraphQL → REST



The screenshot shows the GraphQL SOFA documentation page. At the top left is the logo and name 'GraphQL SOFA'. To the right are icons for GitHub and a menu. Below the header is a dark terminal-style box containing the command `yarn add sofa-api`. Underneath is a section titled 'Usage' with a horizontal line below it. A code block contains the following JavaScript code:

```
import { useSofa } from 'sofa-api';
import express from 'express';

const app = express();

app.use('/api', useSofa({ schema }));

// GET /api/messages
// POST /api/add-message
```

 To the right of the code are icons for search and copy. At the bottom right of the code block is a blue circular chat icon. At the very bottom of the page, it says 'Last updated on August 3, 2023'.

GraphQL SOFA

```
yarn add sofa-api
```

Usage

```
import { useSofa } from 'sofa-api';
import express from 'express';

const app = express();

app.use('/api', useSofa({ schema }));

// GET /api/messages
// POST /api/add-message
```

Last updated on August 3, 2023

Кейс: Any → GQL

The screenshot shows the GraphQL Mesh documentation page. The header includes the GraphQL Mesh logo and the text "A fully-featured GraphQL gateway framework". Navigation links for "Docs" and "Examples" are present, along with a GitHub icon. A sidebar on the left lists various source handlers, with "OpenAPI / Swagger" selected and highlighted. The main content area provides instructions on how to get started, including installation commands for npm, pnpm, yarn, and bun. A code block shows the configuration for the OpenAPI handler in a .meshrc.yaml file. A note mentions that the handler is based on the JSON Schema handler. At the bottom, there is a section titled "Overriding default Query/Mutation operations" and a blue chat bubble icon.

GraphQL Mesh
A fully-featured GraphQL gateway framework

Docs Examples GitHub

Source Handlers

- Introduction
- GraphQL
- OpenAPI / Swagger**
- Federation Supergraph
- gRPC / Protobuf
- JSON Schema or Samples
- PostgreSQL / PostGraphile
- SOAP
- MongoDB / Mongoose
- OData / Microsoft Graph
- Apache Thrift
- SQLite / Tuql
- MySQL
- Neo4j

Transforms >

Dark

To get started, install the handler library:

[npm](#) [pnpm](#) [yarn](#) [bun](#)

```
npm i @graphql-mesh/openapi
```

Now, you can use it directly in your Mesh config file / URL:

```
.meshrc.yaml
```

```
sources:
  - name: MyOpenapiApi
    handler:
      openapi:
        source: ./my-schema.json
```

Note that this handler is based on the [JSON Schema handler](#) - so it's configurations will apply here as well.

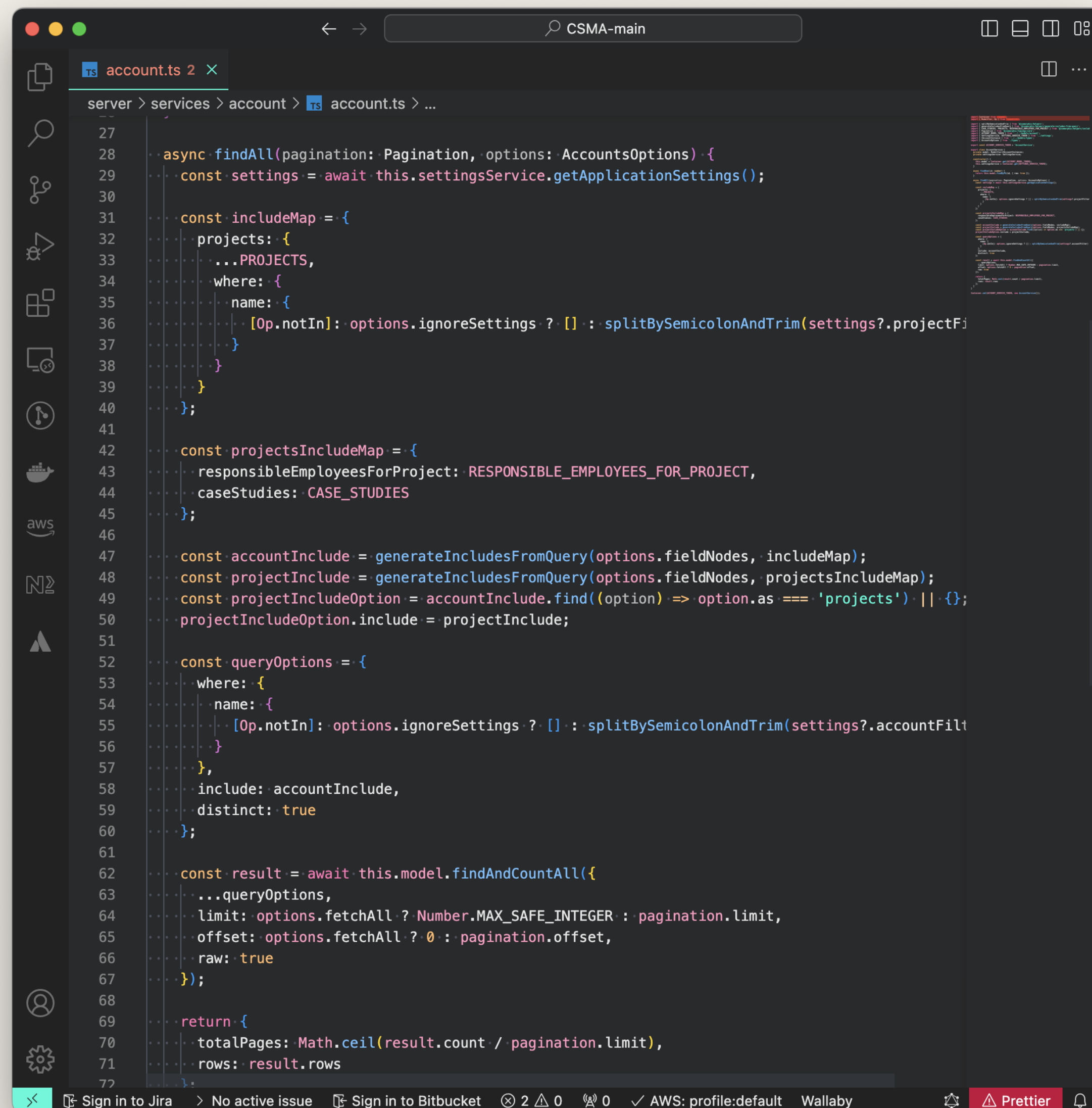
Overriding default
Query/Mutation operations

Кейс: GraphQL → SQL

```
ludes-from-query.ts 3  const schema = { Untitled-1 3  ...
1  const schema = {
2    "fieldName": "RugbyPlayers",
3    "fieldNodes": [
4      {
5        "kind": "Field",
6        "name": {
7          "kind": "Name",
8          "value": "RugbyPlayers",
9        },
10       "arguments": [],
11       "directives": [],
12       "selectionSet": {
13         "kind": "SelectionSet",
14         "selections": [
15           {
16             "kind": "Field",
17             "name": {
18               "kind": "Name",
19               "value": "full_name",
20             },
21           },
22           {
23             "kind": "Field",
24             "name": {
25               "kind": "Name",
26               "value": "club",
27             },
28           },
29           {
30             "kind": "SelectionSet",
31             "selections": [
32               {
33                 "kind": "Field",
34                 "name": {
35                   "kind": "Name",
36                   "value": "full_name",
37                 },
38               },
39               {
40                 "kind": "Field",
41                 "name": {
42                   "kind": "Name",
43                   "value": "club",
44                 },
45               },
46             ],
47             "arguments": [],
48             "directives": [],
49           },
50         ],
51       },
52     },
53   ],
54 },
55 }
```

```
account.ts 2  generate-includes-from-query.ts 3  X
isomorphic > helpers > generate-includes-from-query.ts > ...
50  }
51
52  export function generateIncludesFromQuery(fieldNodes: readonly FieldNode[], includeMap: IIncludeMap) {
53    let attributes = [];
54    return fieldNodes.reduce((acc: IncludeOptions[], fieldNode) => {
55      if (!fieldNode.selectionSet) {
56        return acc;
57      }
58
59      attributes = getAttributesForField(fieldNode);
60
61      if (fieldNode.name.value in includeMap) {
62        const valueToInclude = includeMap[fieldNode.name.value];
63        valueToInclude.attributes = attributes;
64        acc.push(valueToInclude);
65        attributes = [];
66      }
67
68      const interimResult = generateIncludesFromQuery(
69        fieldNode.selectionSet.selections as readonly FieldNode[],
70        includeMap
71      );
72
73      return acc.concat([...interimResult]);
74    }, []);
75  }
76
```

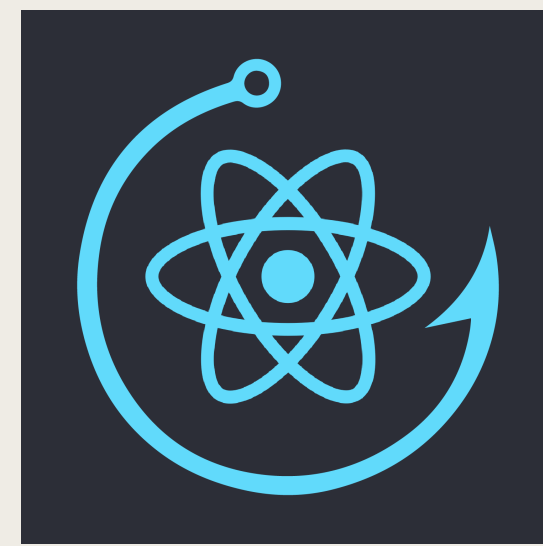
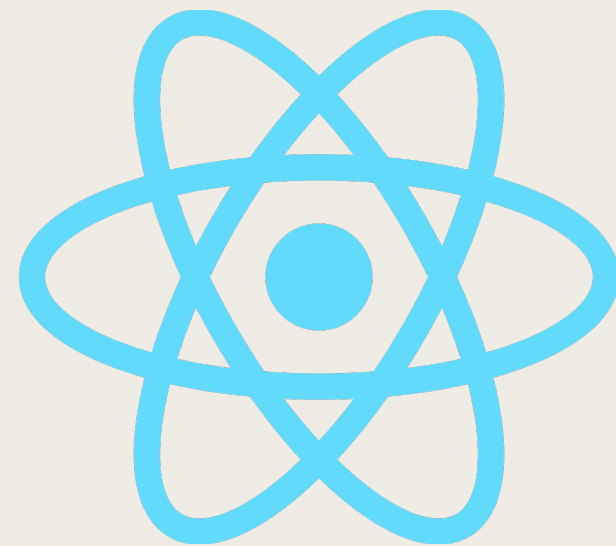

Кейс: GraphQL → SQL



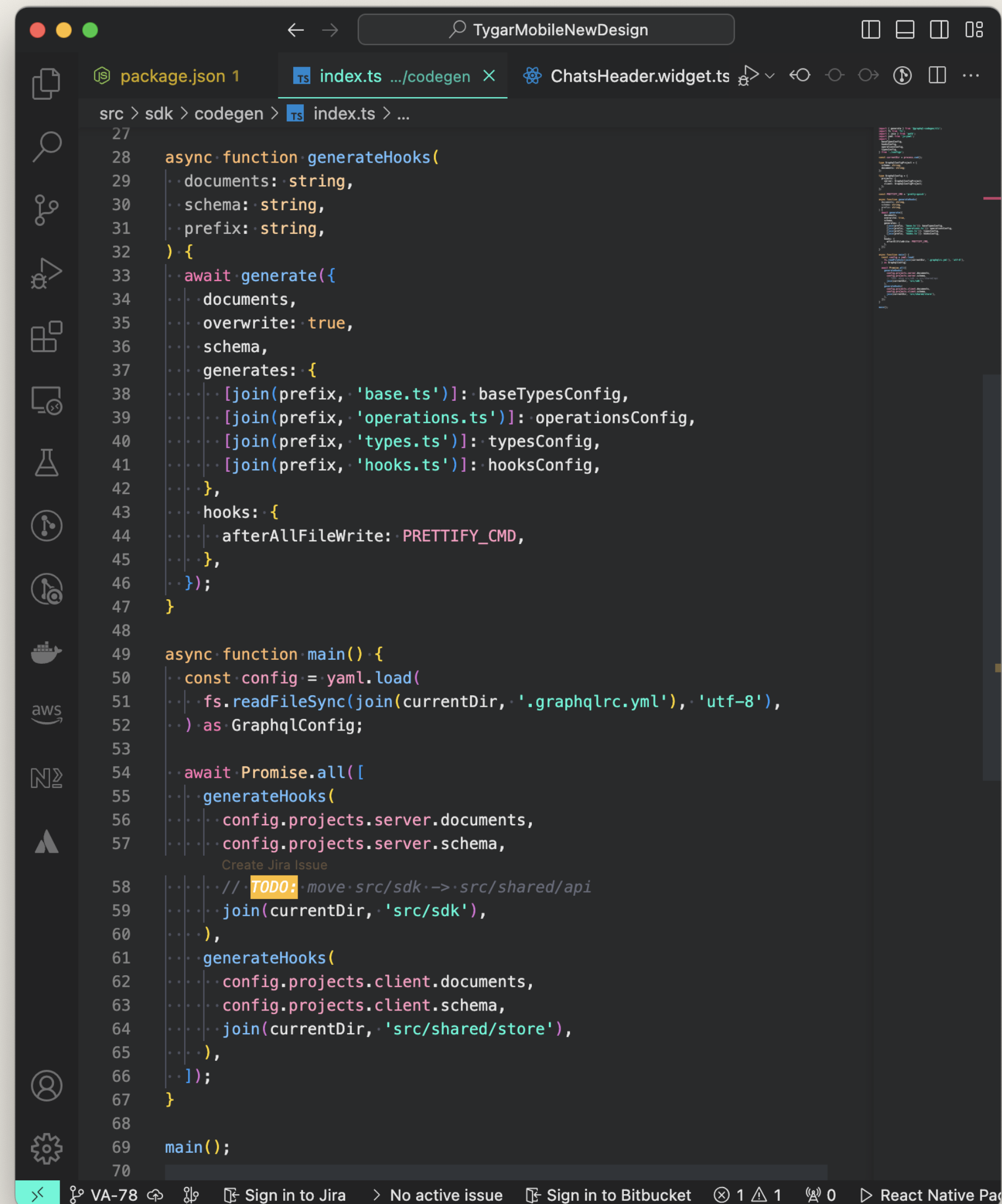
```
server > services > account > account.ts > ...
27
28 ..async findAll(pagination: Pagination, options: AccountsOptions) {
29 ..  const settings = await this.settingsService.getApplicationSettings();
30
31 ..  const includeMap = {
32 ..    projects: {
33 ..      ...PROJECTS,
34 ..      where: {
35 ..        name: {
36 ..          [Op.notIn]: options.ignoreSettings ? [] : splitBySemicolonAndTrim(settings?.projectFi
37 ..        }
38 ..      }
39 ..    }
40 ..  };
41
42 ..  const projectsIncludeMap = {
43 ..    responsibleEmployeesForProject: RESPONSIBLE_EMPLOYEES_FOR_PROJECT,
44 ..    caseStudies: CASE_STUDIES
45 ..  };
46
47 ..  const accountInclude = generateIncludesFromQuery(options.fieldNodes, includeMap);
48 ..  const projectInclude = generateIncludesFromQuery(options.fieldNodes, projectsIncludeMap);
49 ..  const projectIncludeOption = accountInclude.find((option) => option.as === 'projects') || {};
50 ..  projectIncludeOption.include = projectInclude;
51
52 ..  const queryOptions = {
53 ..    where: {
54 ..      name: {
55 ..        [Op.notIn]: options.ignoreSettings ? [] : splitBySemicolonAndTrim(settings?.accountFilt
56 ..      }
57 ..    },
58 ..    include: accountInclude,
59 ..    distinct: true
60 ..  };
61
62 ..  const result = await this.model.findAndCountAll({
63 ..    ...queryOptions,
64 ..    limit: options.fetchAll ? Number.MAX_SAFE_INTEGER : pagination.limit,
65 ..    offset: options.fetchAll ? 0 : pagination.offset,
66 ..    raw: true
67 ..  });
68
69 ..  return {
70 ..    totalPages: Math.ceil(result.count / pagination.limit),
71 ..    rows: result.rows
72 ..  };
73 }
```

The image shows a code editor window with a dark theme. The main area contains TypeScript code for a GraphQL resolver. The code defines a `findAll` function that takes pagination and options. It fetches application settings and constructs query options for a database query. The options include a `where` clause with a `name` filter and an `include` clause. The code uses `generateIncludesFromQuery` to build the include clause based on the provided field nodes. The final query is executed using `findAndCountAll` with pagination parameters. The result is returned as an object with `totalPages` and `rows`.

Фронтенд

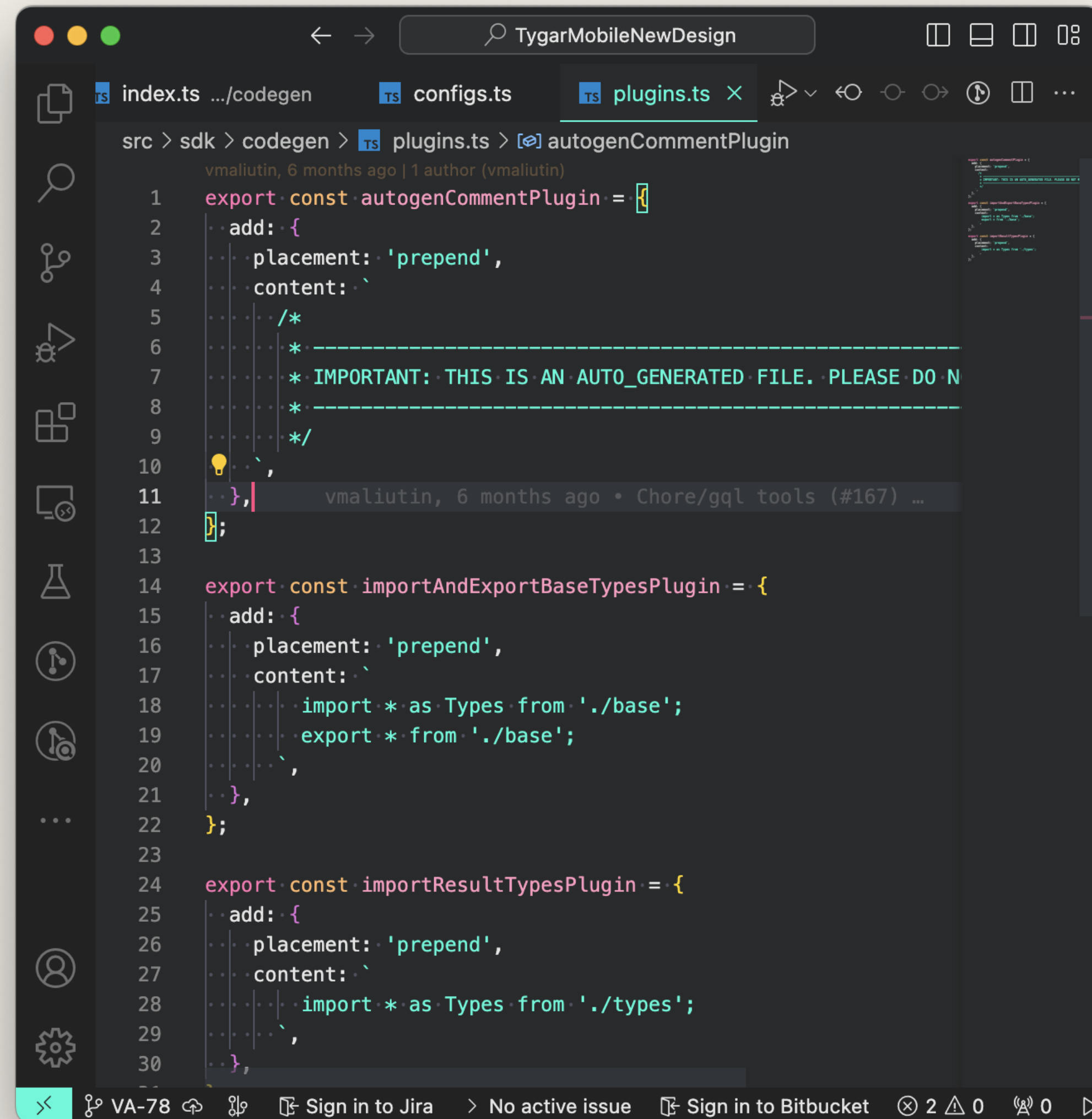


Скрипт для генерации React hooks



```
src > sdk > codegen > index.ts > ...
27
28 async function generateHooks(
29   documents: string,
30   schema: string,
31   prefix: string,
32 ) {
33   await generate({
34     documents,
35     overwrite: true,
36     schema,
37     generates: {
38       [join(prefix, 'base.ts')]: baseTypesConfig,
39       [join(prefix, 'operations.ts')]: operationsConfig,
40       [join(prefix, 'types.ts')]: typesConfig,
41       [join(prefix, 'hooks.ts')]: hooksConfig,
42     },
43     hooks: {
44       afterAllFileWrite: PRETTIFY_CMD,
45     },
46   });
47 }
48
49 async function main() {
50   const config = yaml.load(
51     fs.readFileSync(join(currentDir, '.graphqlrc.yml'), 'utf-8'),
52   ) as GraphQLConfig;
53
54   await Promise.all([
55     generateHooks(
56       config.projects.server.documents,
57       config.projects.server.schema,
58       // TODO: move src/sdk -> src/shared/api
59       join(currentDir, 'src/sdk'),
60     ),
61     generateHooks(
62       config.projects.client.documents,
63       config.projects.client.schema,
64       join(currentDir, 'src/shared/store'),
65     ),
66   ]);
67 }
68
69 main();
70
```


Пишем свои плагины




The image shows a code editor window with the following content:



```
src > sdk > codegen > plugins.ts > autogenCommentPlugin
v maliutin, 6 months ago | 1 author (vmaliutin)
1 export const autogenCommentPlugin = {
2   add: {
3     placement: 'prepend',
4     content: `
5       /*
6       * -----
7       * IMPORTANT: THIS IS AN AUTO_GENERATED FILE. PLEASE DO N
8       * -----
9       */
10    `,
11  },
12 };
13
14 export const importAndExportBaseTypesPlugin = {
15   add: {
16     placement: 'prepend',
17     content: `
18     import * as Types from './base';
19     export * from './base';
20     `,
21  },
22 };
23
24 export const importResultTypesPlugin = {
25   add: {
26     placement: 'prepend',
27     content: `
28     import * as Types from './types';
29     `,
30  },
31 };
```


The editor interface includes a search bar at the top with the text "TygarMobileNewDesign", a sidebar on the left with various icons, and a status bar at the bottom with system and application information.


Используем чужие плагины


Catch the highlights of GraphQLConf 2023! [Click for recordings](#). Or check out our [recap blog post](#). ✕

 **GraphQL Code Generator**
Generation of typed queries, mutations, subscriptions and typed GraphQL resolvers

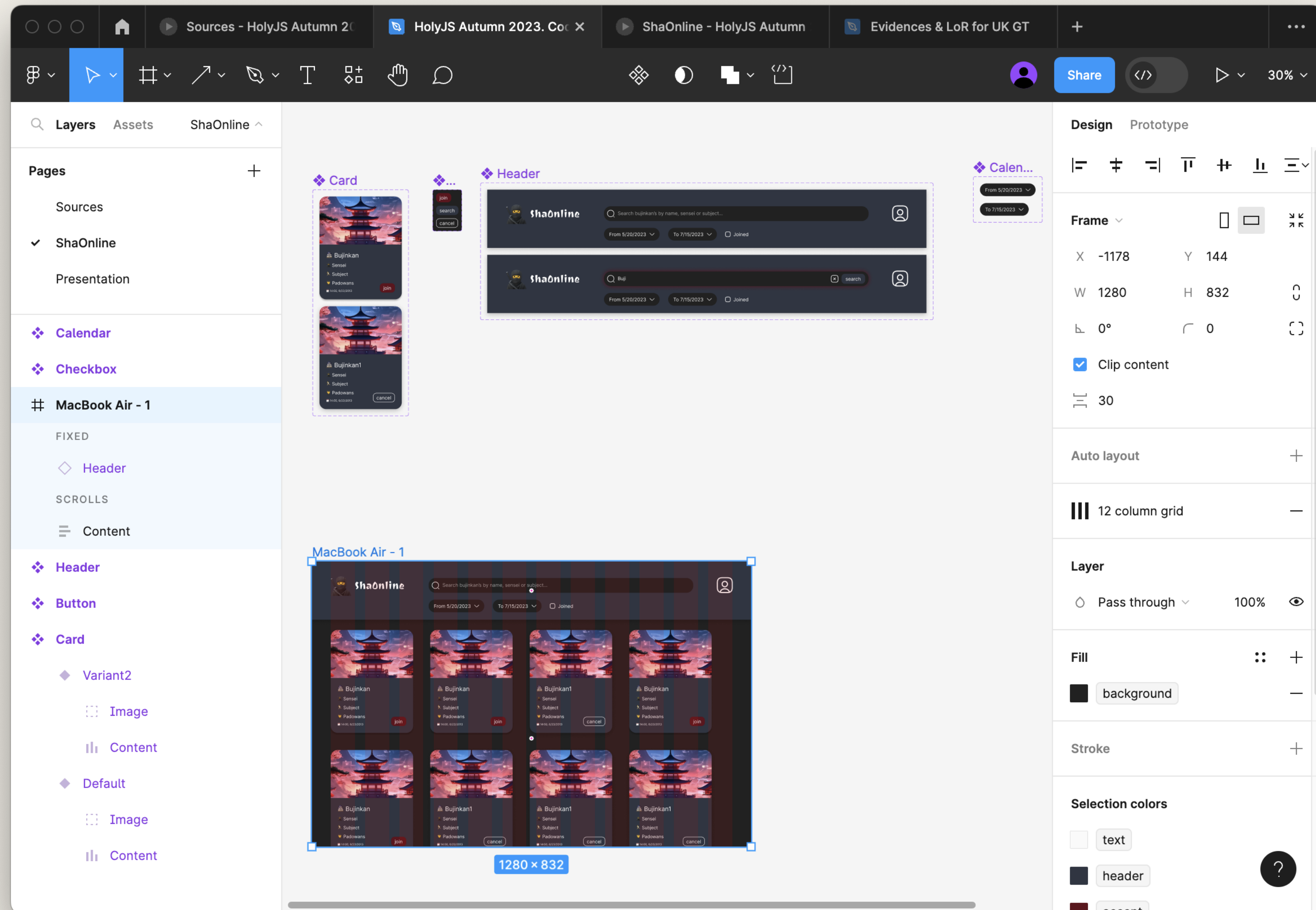
 

 **Add**
GraphQL Code Generator plugin for adding custom content to your output file ➤
plugin

 **TypeScript React Apollo**
GraphQL Code Generator plugin for generating a ready-to-use React Components/HOC/Hooks based on GraphQL operations ➤
plugin typescript react apollo

 **Introspection**
GraphQL Code Generator plugin for generating an introspection JSON file for a GraphQLSchema 💬
plugin utilities

Дизайним проект в Figma



Создаем аккаунт и логинимся в plugin

Locofy

Alright! Are you ready?
Let's create your first project

React Native codegen-ninja

CODE SETTINGS
You can change this later while exporting code

TypeScript JavaScript

Project Environment
 React Native CLI Expo

Style Guide ⓘ
 Style Constants

I'll set it up later **Next**

Экспортируем компонент в код и фиксируем пропсы

The screenshot shows the Figma interface with the 'Header' component selected. The 'Locofy FREE BETA' plugin is open, displaying a list of style props to be fixed. The plugin interface includes a search bar, a list of recommendations, and a table of style props.

6 recommendations Accept All

Feel free to edit before accepting. This is a Figma Component

#2 Component Name Property1Default Skip for now

Instances

Header

Style Props

#	Prop Name	Value	
1	property1DefaultWidth	91.43rem	×
2	property1DefaultHeight	12.14rem	×
3	property1DefaultPadding	2.86rem 0rem 1.43rem	×
4	property1DefaultPosition	absolute	⌵ ×
5	property1DefaultTop	0rem	×
6	property1DefaultLeft	0rem	×
7	hovHeight	1.29rem	×

Reject (1) Accept (1)

Component Name

Done ?

Комментарии

- Редко норм свойства изначально генерирует
- Не всегда автоматом может правильные семантические HTML теги определить
- Но как сконфигурировал - уже помнит и можно дальше править дизайн
- Есть возможность синхронизации с Github и деплой в разные клауды (Netlify, Vercel, Github Pages)

Пишем логику

hey gpt, I want to write backend service using nestjs for ninja online school. Could you write logic for my endpoints?

Of course, I can help you get started with writing the logic for your backend service using NestJS for a Ninja online school. NestJS is a popular framework for building scalable and maintainable server-side applications. Below, I'll provide you with some example endpoints and their corresponding logic to give you a starting point. You can expand upon this foundation to meet the specific requirements of your project.

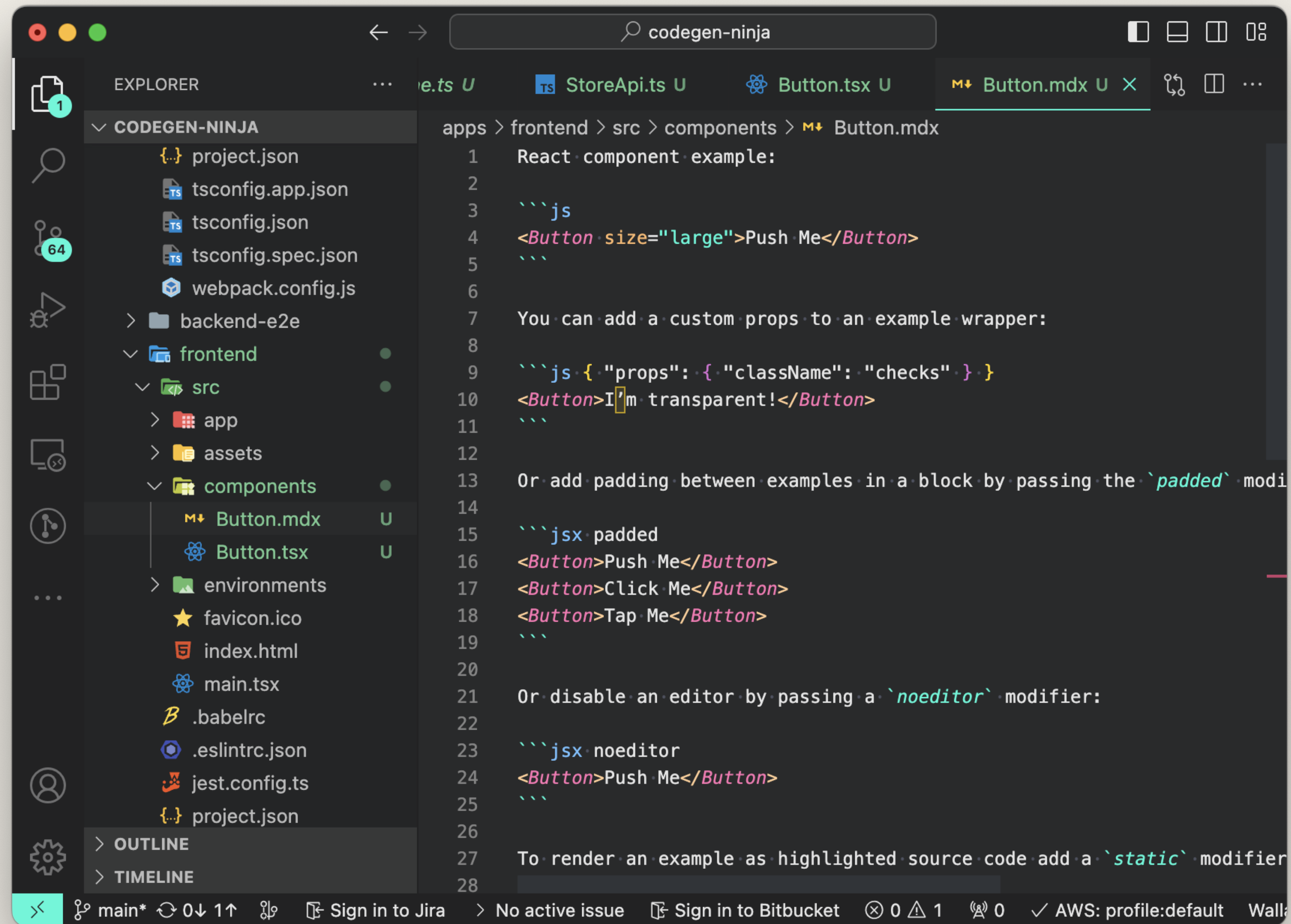
Before you start, make sure you have NestJS installed and set up in your project.

Send a message

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT September 25 Version](#)

Сгенерируем фронтенд доки

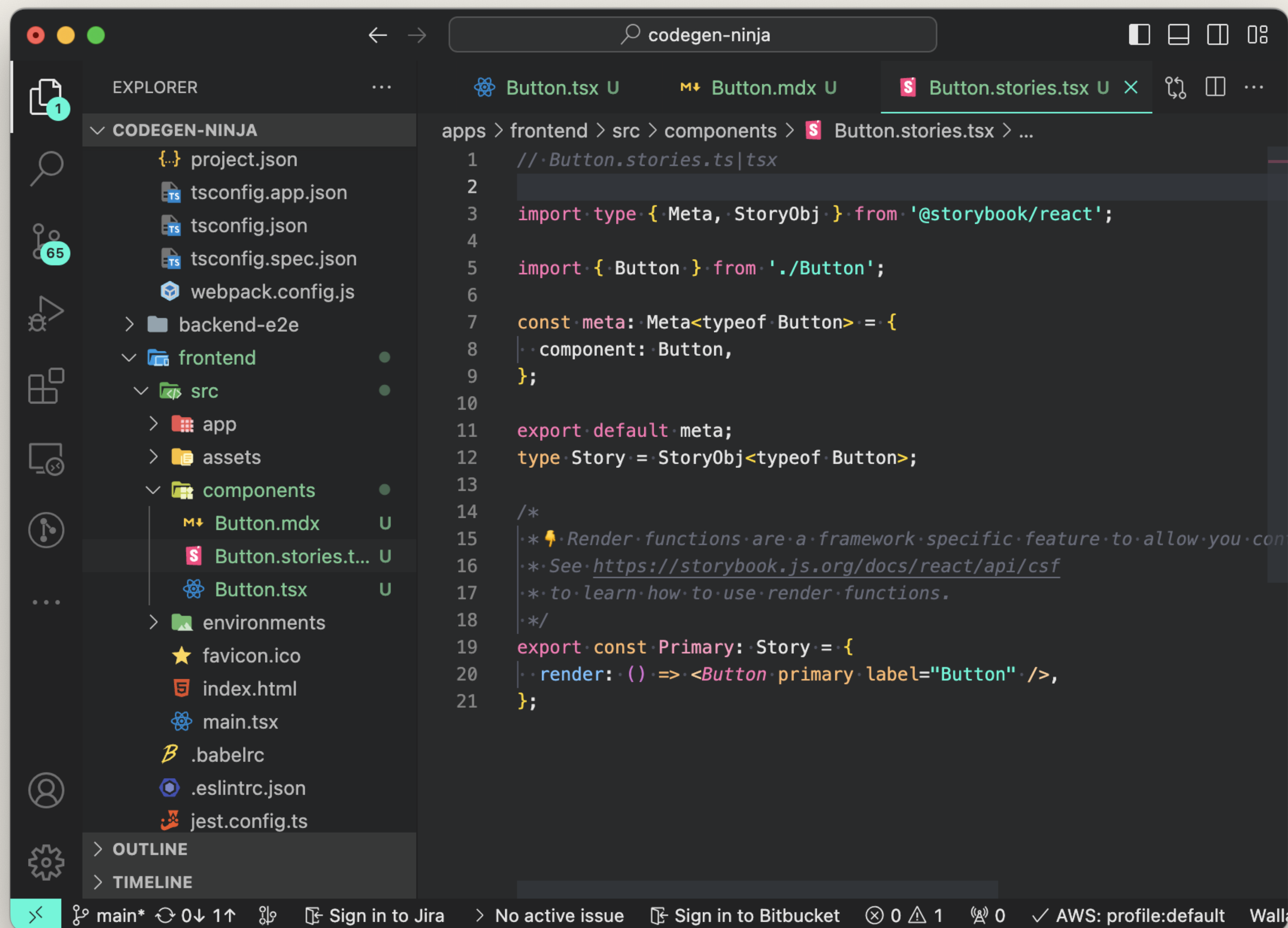
Пишем Markdown



The screenshot shows the VS Code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'CODEGEN-NINJA' with folders like 'backend-e2e', 'frontend', and 'components'. The code editor displays the content of 'Button.mdx' with the following text:

```
1 React component example:
2
3 ```js
4 <Button size="large">Push Me</Button>
5 ```
6
7 You can add a custom props to an example wrapper:
8
9 ```js { "props": { "className": "checks" } }
10 <Button>I'm transparent!</Button>
11 ```
12
13 Or add padding between examples in a block by passing the `padded` modifier:
14
15 ```jsx padded
16 <Button>Push Me</Button>
17 <Button>Click Me</Button>
18 <Button>Tap Me</Button>
19 ```
20
21 Or disable an editor by passing a `noeditor` modifier:
22
23 ```jsx noeditor
24 <Button>Push Me</Button>
25 ```
26
27 To render an example as highlighted source code add a `static` modifier
28
```


Пишем TypeScript



Получаем что-от такое

The image shows the Grommet component library interface. On the left is a sidebar with a search bar and a list of components under 'CONTROLS' and 'DATA'. The main area displays a preview of various components: 'Heading', 'Paragraph', 'Text', 'Anchor', 'Menu', 'Button', 'Select', 'CheckBox', 'CheckBox toggle', 'RadioButton 1', 'RadioButton 2', 'TextInput', and 'TextArea'. Below the preview is a code editor showing the corresponding React code.

```
1 import React, { useState } from 'react';
2
3 import {
4   Accordion,
5   AccordionPanel,
6   Anchor,
7   Box,
8   Button,
9   Calendar,
10  Chart,
11  CheckBox,
12  Clock,
```

Или такое

React Styleguidist
Example Basic Style
Guide
1.0.0

Filter by name

Button
CounterButton
Placeholder
Push Button 🎉
RandomButton
WrappedButton

Button

src/components/Button/Button.js

The only true button.

PROPS & METHODS

Basic button:

Push Me

VIEW CODE

Big pink button:

Click Me

VIEW CODE

And you *can* **use** any [Markdown](#) here.

Fenced code blocks with `js`, `jsx` or `javascript` languages are rendered as a interactive playgrounds:

Fork me on GitHub

А где кодген?



Here is a basic example

index.tsx

```
import { parse } from 'react-docgen';

const code = `
/** My first component */
export default ({ name }: { name: string }) => <div>{{name}}</div>;
`;

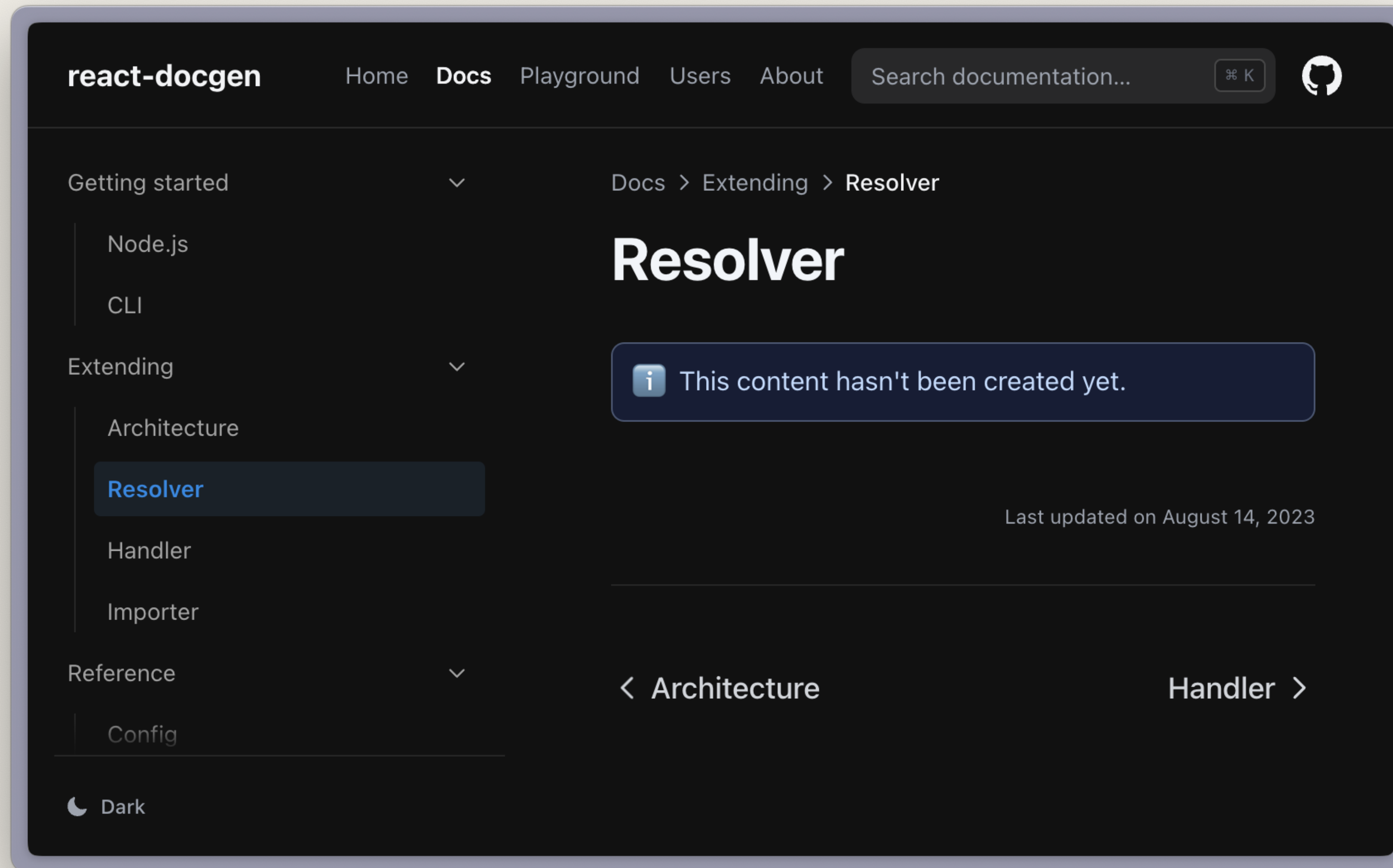
const documentation = parse(code);

console.log(documentation);
```



```
[
  {
    "description": "My first component",
    "methods": [],
    "props": {
      "name": {
        "required": true,
        "tsType": {
          "name": "string"
        },
        "description": ""
      }
    }
  }
]
```


Минус - нет документации



Что можно с этим делать?

☰ README.md

How many buttons? One Two Three

NOTES **PROPS** STORY KNOBS VIEWPORT ACTION LOGGER ACCESSIBILITY BACKGROUNDS


Prop Types

"Button" Component

property	propType	required	default	description
children	ReactNode	yes	-	Text of the button
size	"medium" "large"	-	medium	Size of the button
color	"primary" "secondary" "tertiary" "upsell" "white" "black" "light" "dark" "link" "info" "success" "warning" "danger"	-	-	Color of the button
isOutlined	boolean	-	false	Render the button with an outline
isInverted	boolean	-	false	Render the button with the colors inverted

Installation [↗](#)

```
yarn add storybook-addon-react-docgen
```



React Docgen Integration [↗](#)

Демо



Search bujinkan's by name, sensei or subject...



From 5/20/2023

To 7/15/2023

Joined



Orn, Spencer ...

Götz Halldórsson

Escape

3 padowans

14:00, 6/23/2013

join



Orn, Spencer ...

Yhudiyt Agbaria

Hiding

2 padowans

14:00, 6/23/2013

join



Orn, Spencer ...

Konstantin Serrano

Masking

9 padowans

14:00, 6/23/2013

join



Orn, Spencer ...

Ægir Ózse

Escape

4 padowans

14:00, 6/23/2013

join



Orn, Spencer ...

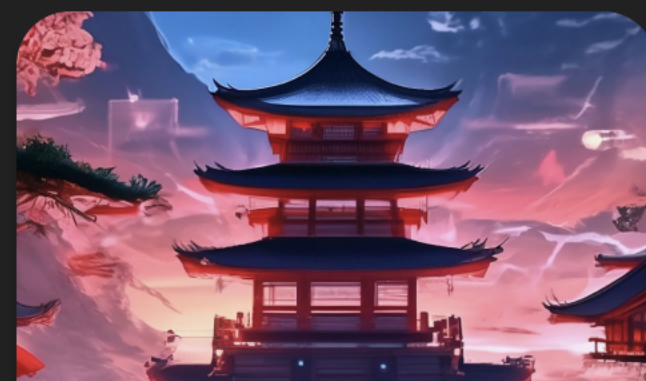
José Kristjánsdóttir

Diversion

1 padowans

14:00, 6/23/2013

join



Orn, Spencer ...

Františka Þorsteinsdóttir

Diversion

7 padowans

14:00, 6/23/2013

join



Orn, Spencer ...

Dörte Jónsdóttir

Assasination

3 padowans

14:00, 6/23/2013

join



Reichel Group

Noam Matthews

Escape

2 padowans

14:00, 6/23/2013


join

CHANGE SCHEMA


Type List

Search Schema...


aggregate variance on columns


Bujinkan  columns and relationships of "Bujinkan"

Bujinkan_aggregate  aggregated selection of "Bujinkan"

Bujinkan_aggregate_fields  aggregate fields of "Bujinkan"


Bujinkan_avg_fields  aggregate avg on columns


Bujinkan_max_fields  aggregate max on columns


Bujinkan_min_fields  aggregate min on columns


Bujinkan_stddev_fields  aggregate stddev on columns

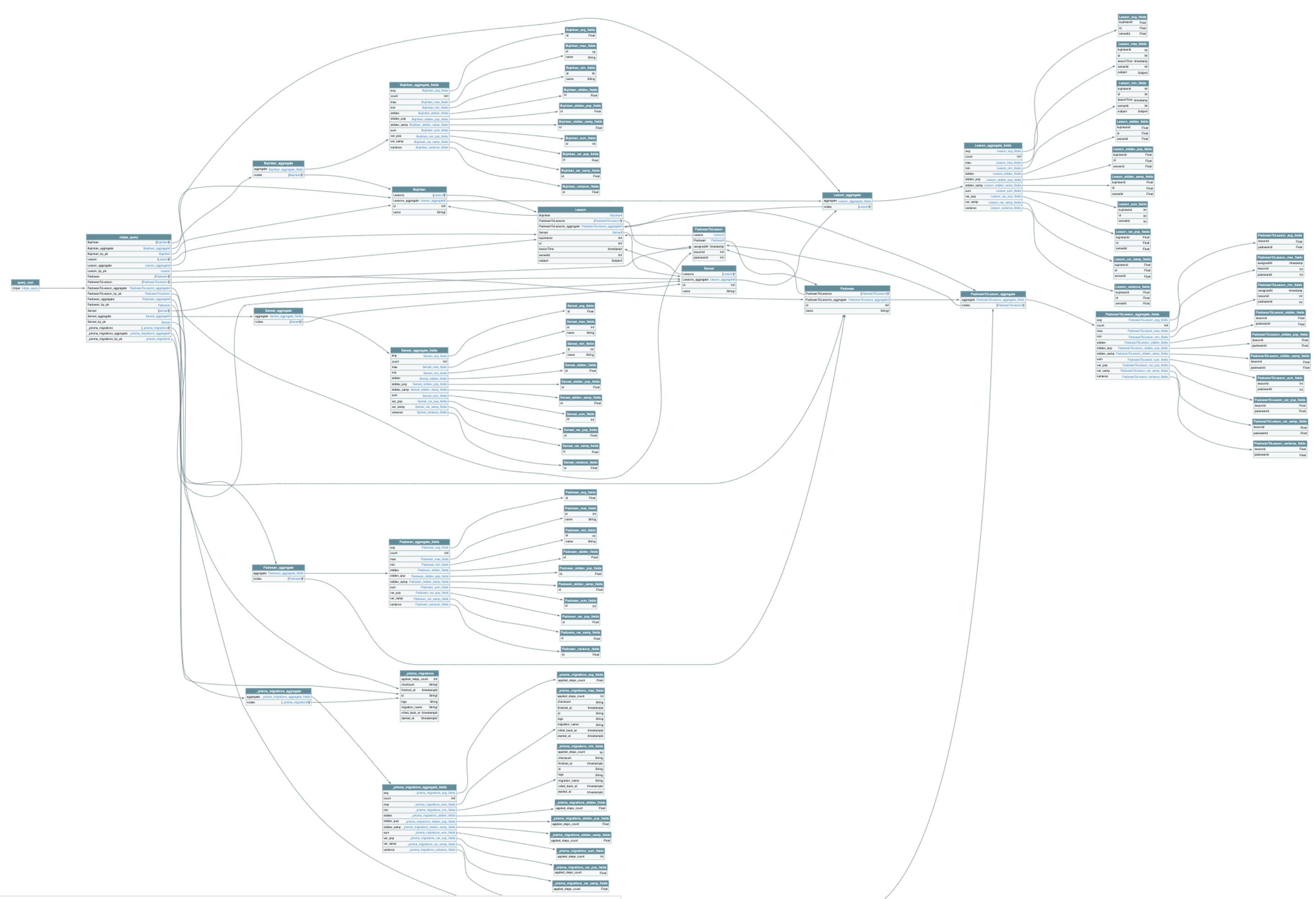
Bujinkan_stddev_pop_fields  aggregate stddev_pop on columns

Bujinkan_stddev_samp_fields  aggregate stddev_samp on columns

Bujinkan_sum_fields  aggregate sum on columns

Bujinkan_var_pop_fields  aggregate var_pop on columns

Bujinkan_var_samp_fields  aggregate var_samp on columns



query_root

- Sort by Alphabet
- Skip Relay
- Skip deprecated
- Show leaf fields

+
RESET
-

ИТОГ

- Что такое кодогенерация
- Зачем она нужна компании, разработчикам и дизайнерам
- Прошлись по всем аспектам разработки и как можно в каждой из них генерировать код (BE, DB, Architecture, FE, Docs, SDK, CI /CD)
- Обзор доступных инструментов и их особенностей
- Демо проект с использованием всех инструментов

Выводы



Вадим Малютин

телеграм: t.me/mfrjke

почта: mltn.dev@yandex.ru

сайт: mltn.dev

Спасибо за внимание!

<https://github.com/mdusaftp/codegen-ninja>

