Т БАНК

PKL

Андрей Зонов
Руководитель группы
Stuff iOS

БАНК

Андрей Зонов

Stuff iOS | Руководитель группы

@IOS_BROADCAST

linkedin.com/in/avzonov

t.me/avzonov

Pkl (Пикл)

- Язык программирования для связи статических ЯП
- Язык созданный для работы с конфигурациями
- Внимание на безопасности и поддержке
- Поддержка языков и IDE на старте

Сегодня

Какую проблему решает PKL

Основные альтернативы

Примеры использования

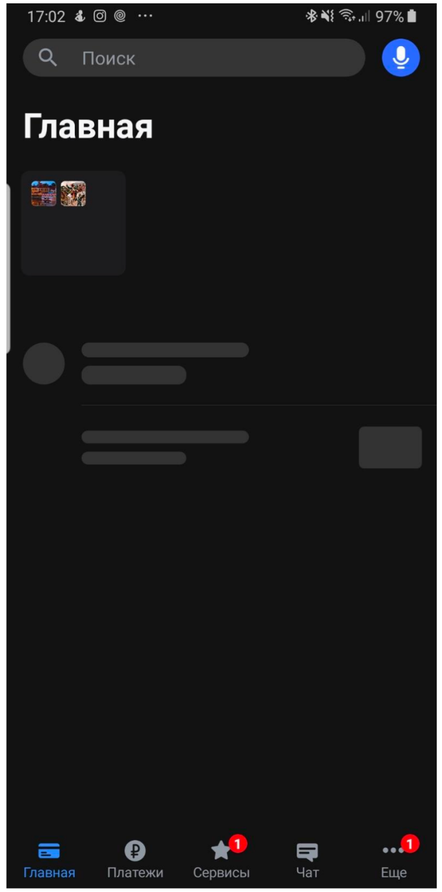
Вывод

Какую проблему решает PKI?

Работа с JSON без тестов

Зачем?

Всё упало

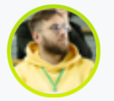


Смотреть запись

ДОКЛАД

Зал 1

«А у нас сейчас всё норм работает?» или Что такое observability мобильного приложения



Даниэль Халиулин

Тинькофф

RU

Инфраструктура

Зачем?

Всё упало

API

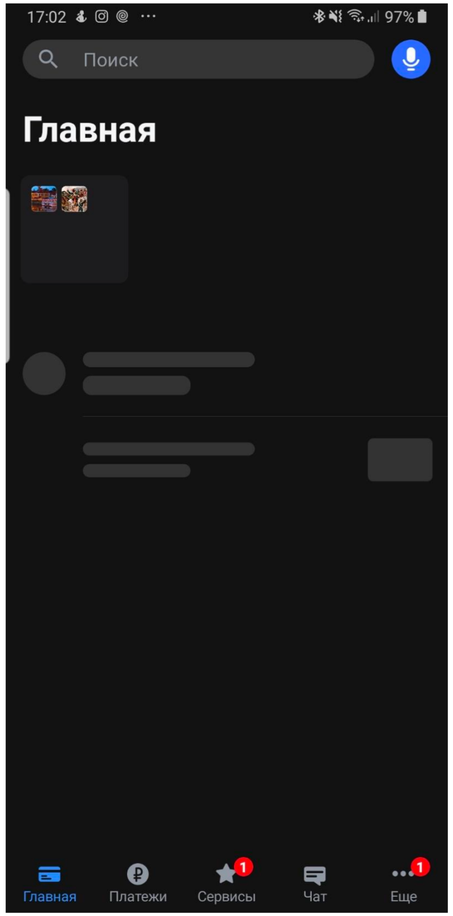
Кэши

Метрики

Базы данных

Бэкенды

Последние релизы сервисов



Смотреть запись

ДОКЛАД

Зал 1

«А у нас сейчас всё норм работает?» или Что такое observability мобильного приложения

Даниэль Халиулин

Тинькофф

RU

Инфраструктура

Зачем?

Всё упало

API

Кэши

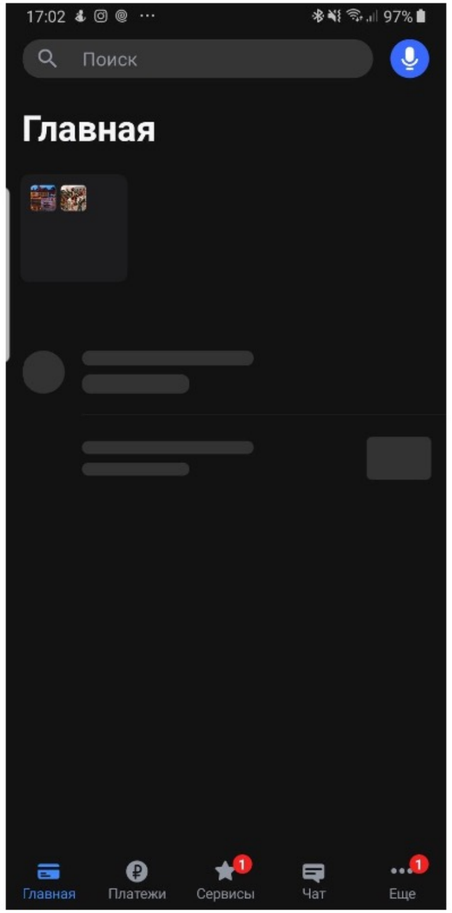
Метрики

Базы данных

Бэкенды

Последние релизы сервисов

Изменения в конфигах

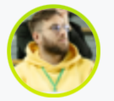


Смотреть запись

ДОКЛАД

Зал 1

«А у нас сейчас всё норм работает?» или Что такое observability мобильного приложения



Даниэль Халиулин

Тинькофф

RU

Инфраструктура

Зачем?

Всё упало

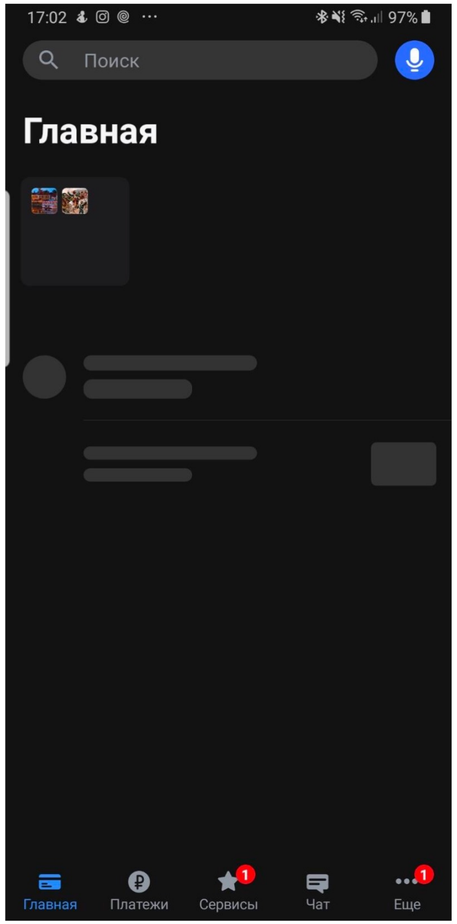
Ожидание

```
{
    "A": "aaaa",
    "B": "bbbbb",
    "C": "ccccc"
}
```

обязательное

Реальность

```
{
    "A": 42,
    "C": "ccccc"
    "D": [
    ]
}
```

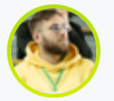


Смотреть запись

ДОКЛАД

Зал 1

«А у нас сейчас всё норм работает?» или Что такое observability мобильного приложения



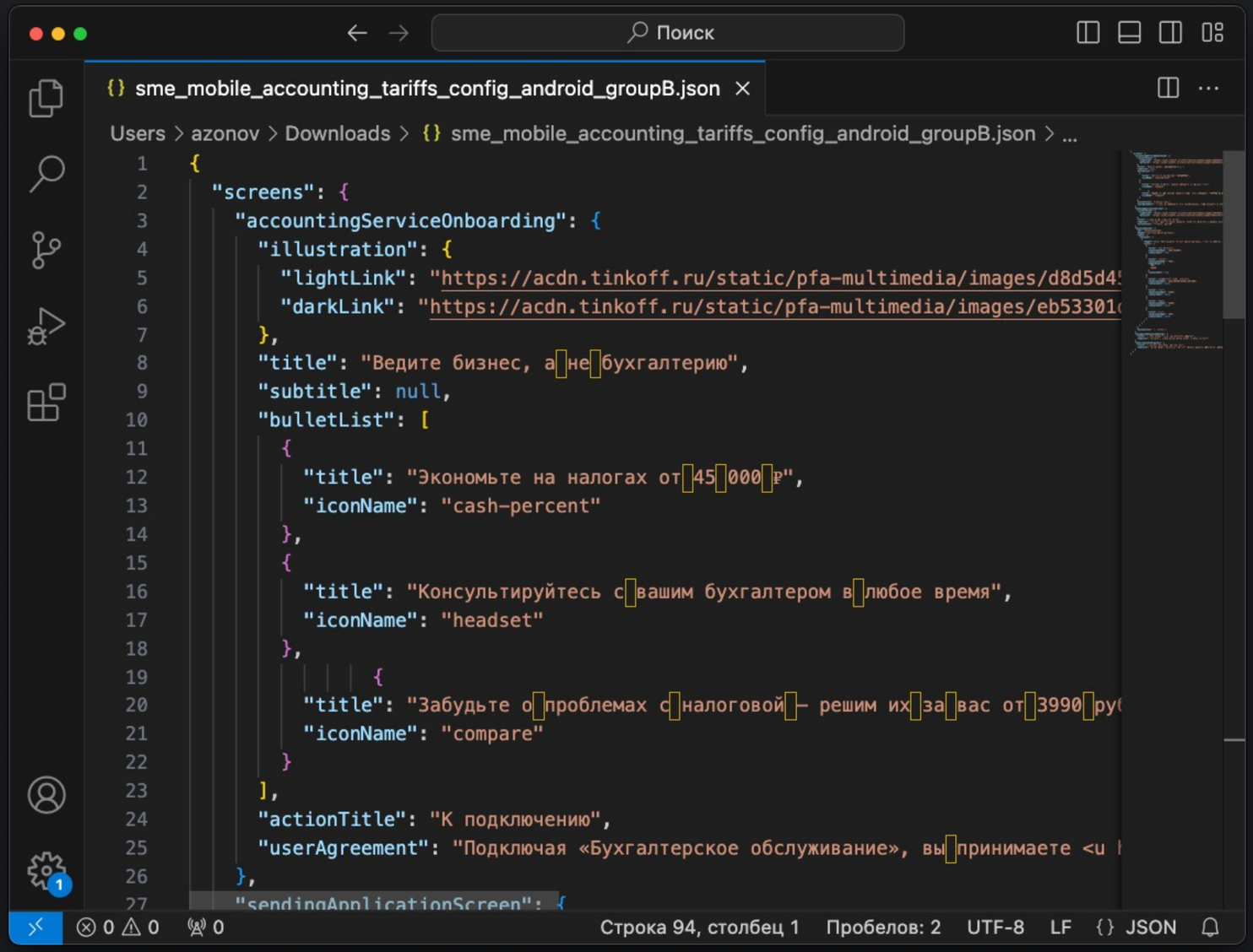
Даниэль Халиулин
Тинькофф

RU

Инфраструктура

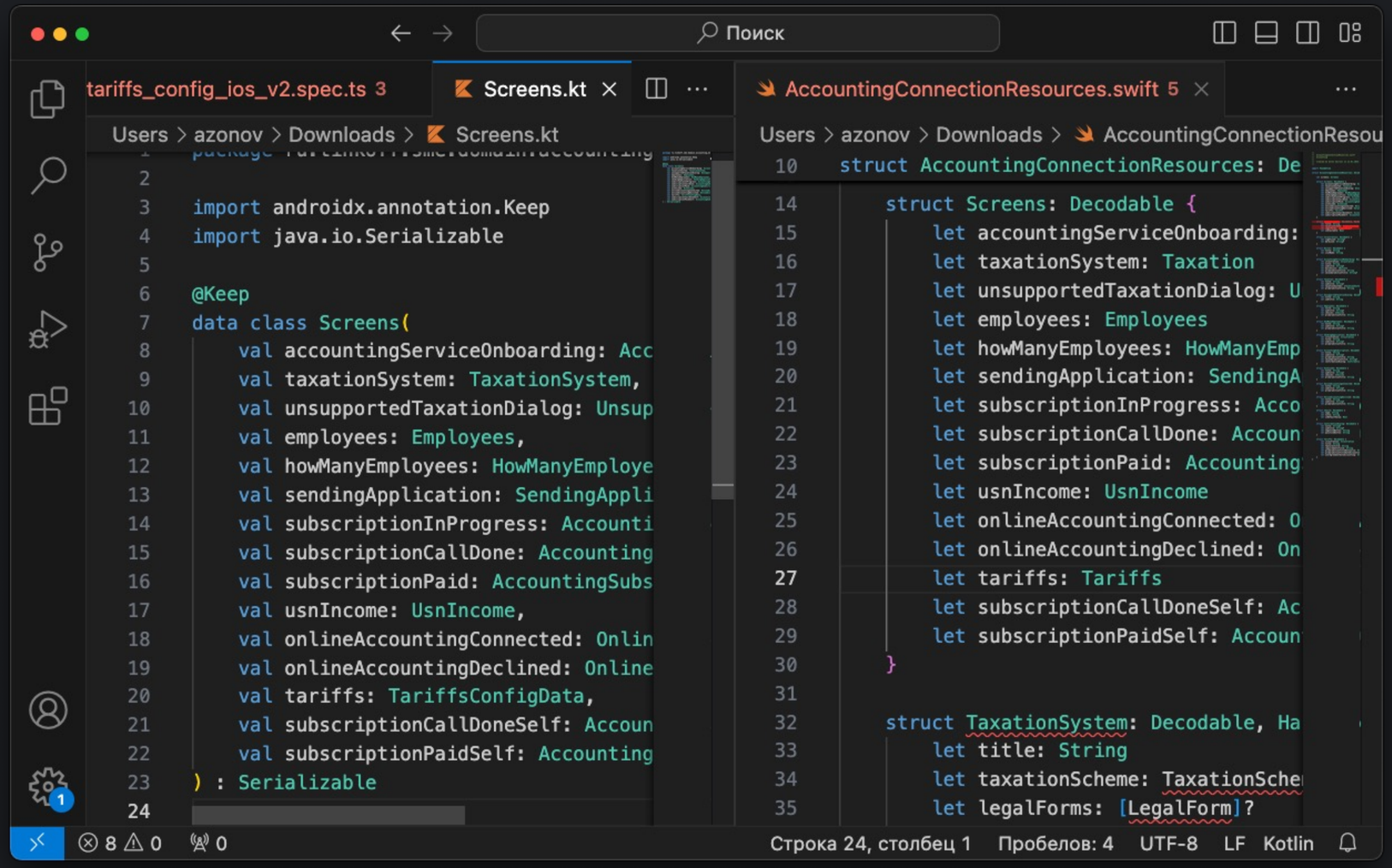
Внедрение валидации контрактов

Как обезопасить прод?



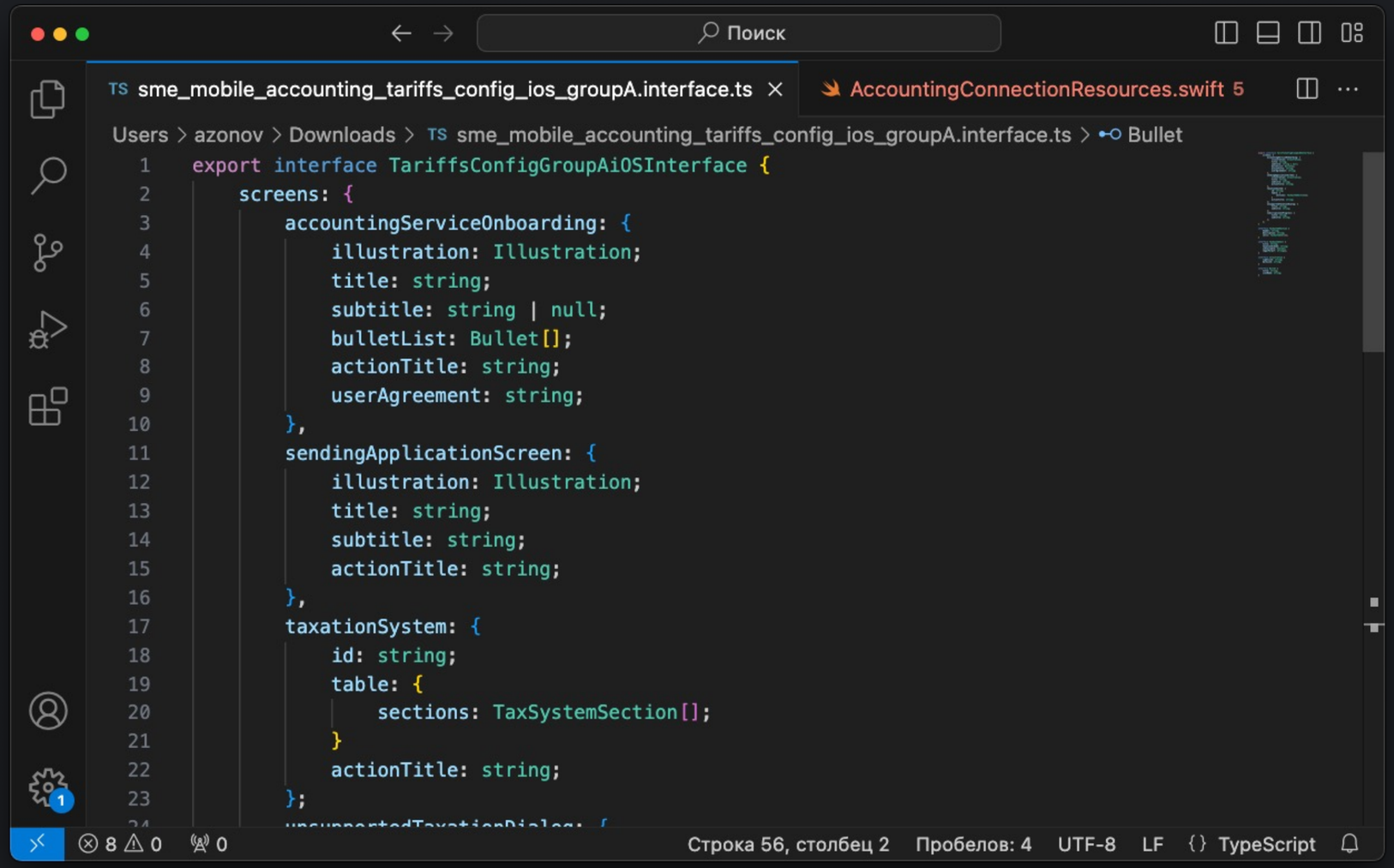
- Json

Как обезопасить прод?



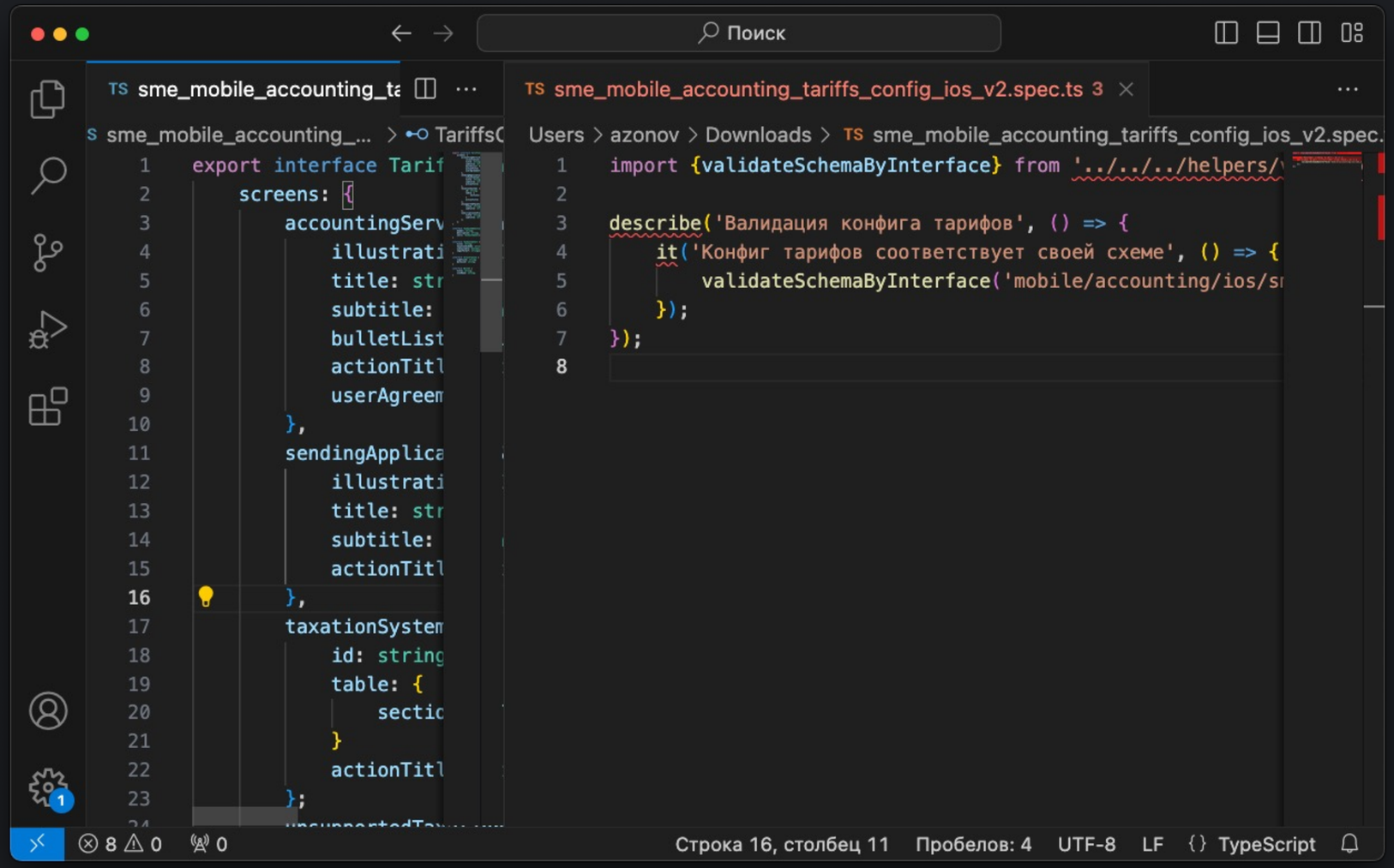
- Json
- Swift
- Kotlin
- JS

TypeScript?



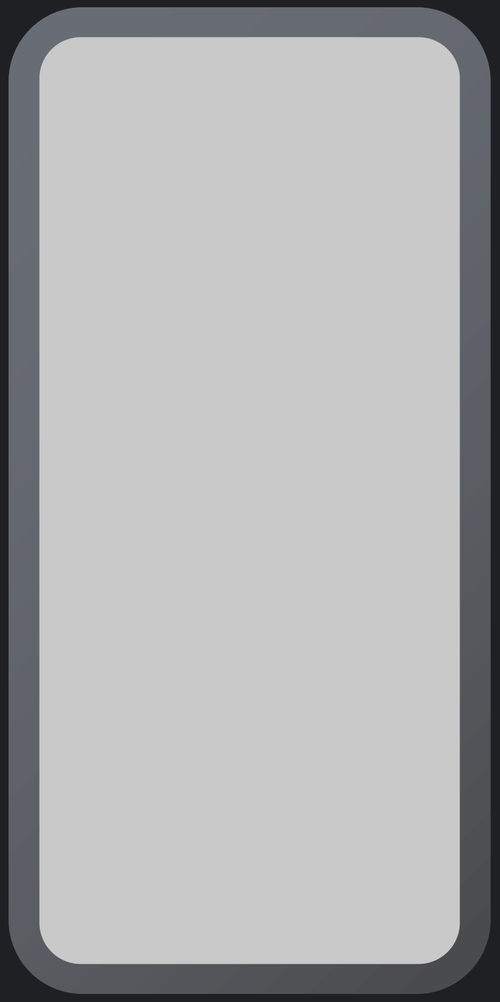
- Json
- Swift
- Kotlin
- JS
- TypeScript

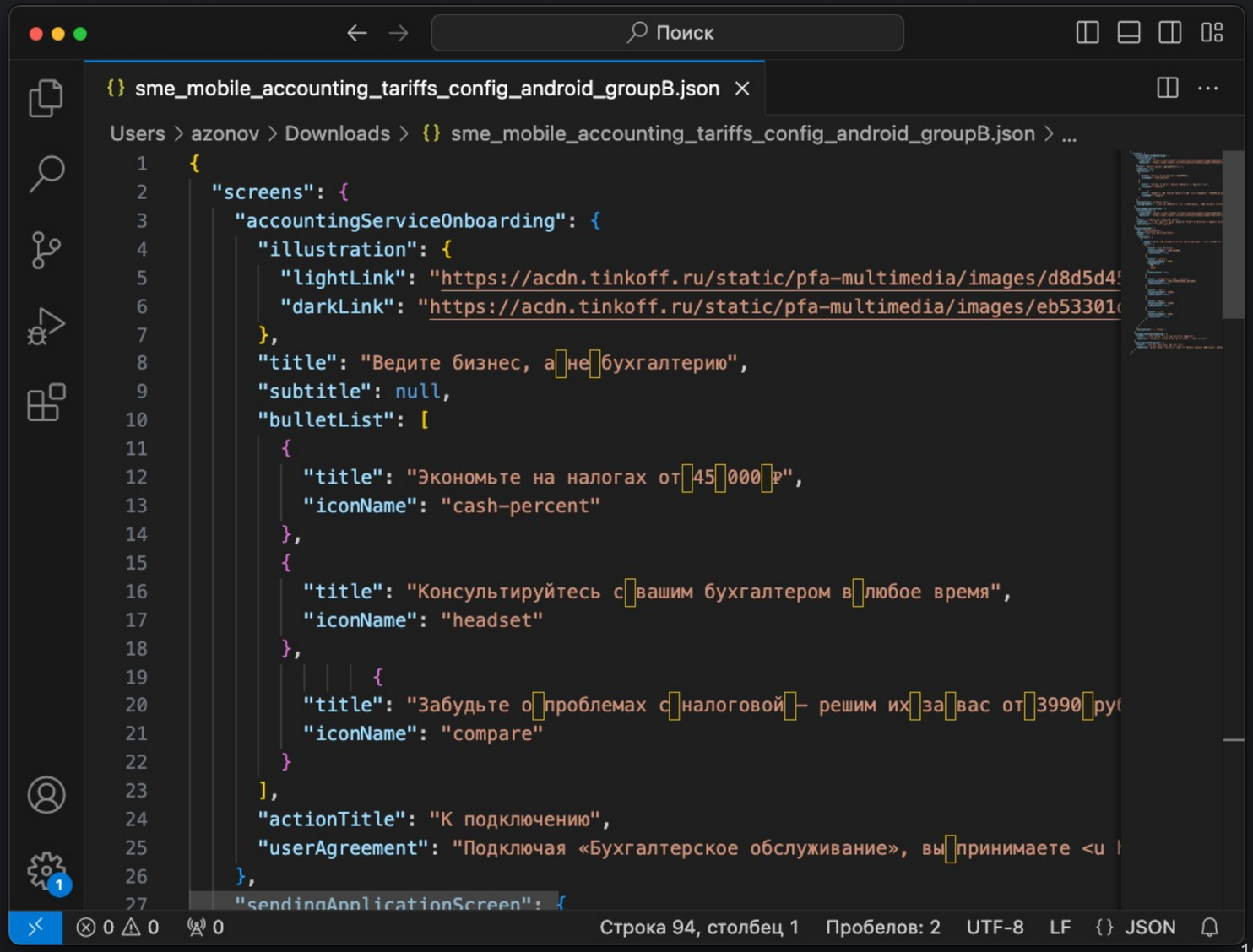
TypeScript?



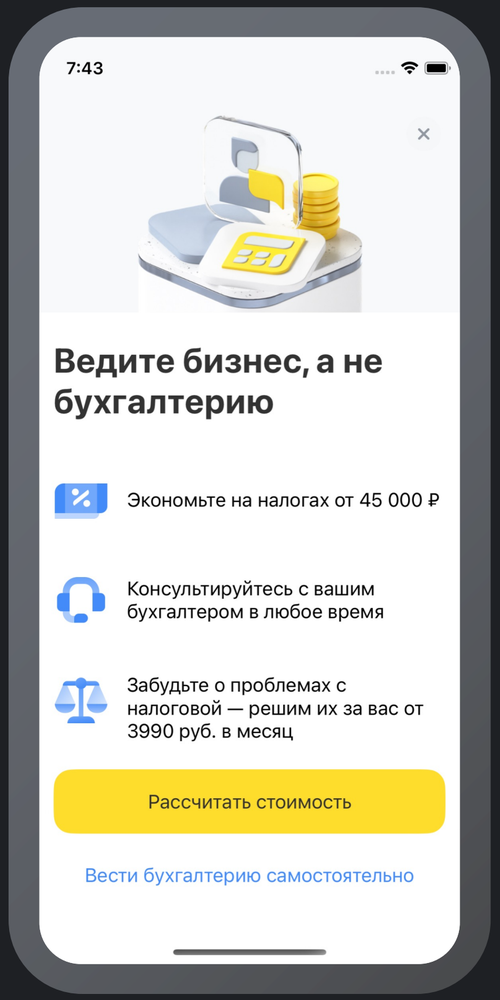
- Json
- Swift
- Kotlin
- JS
- TypeScript

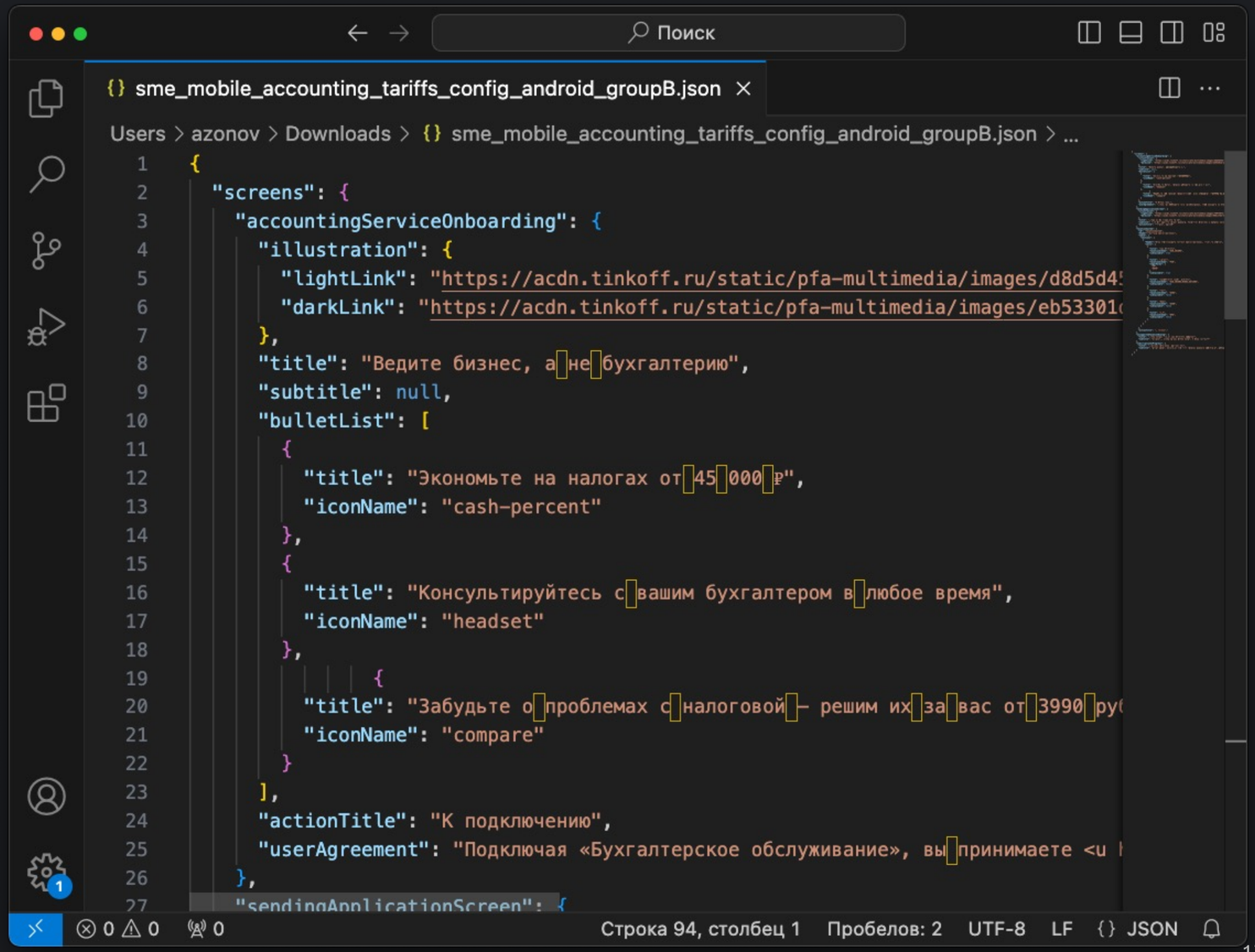
Вы представляете что видит пользователь?



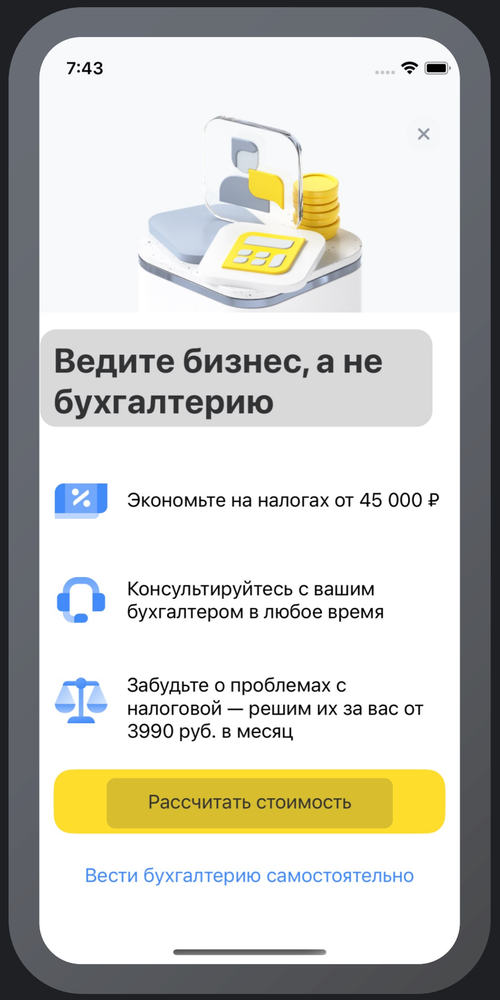


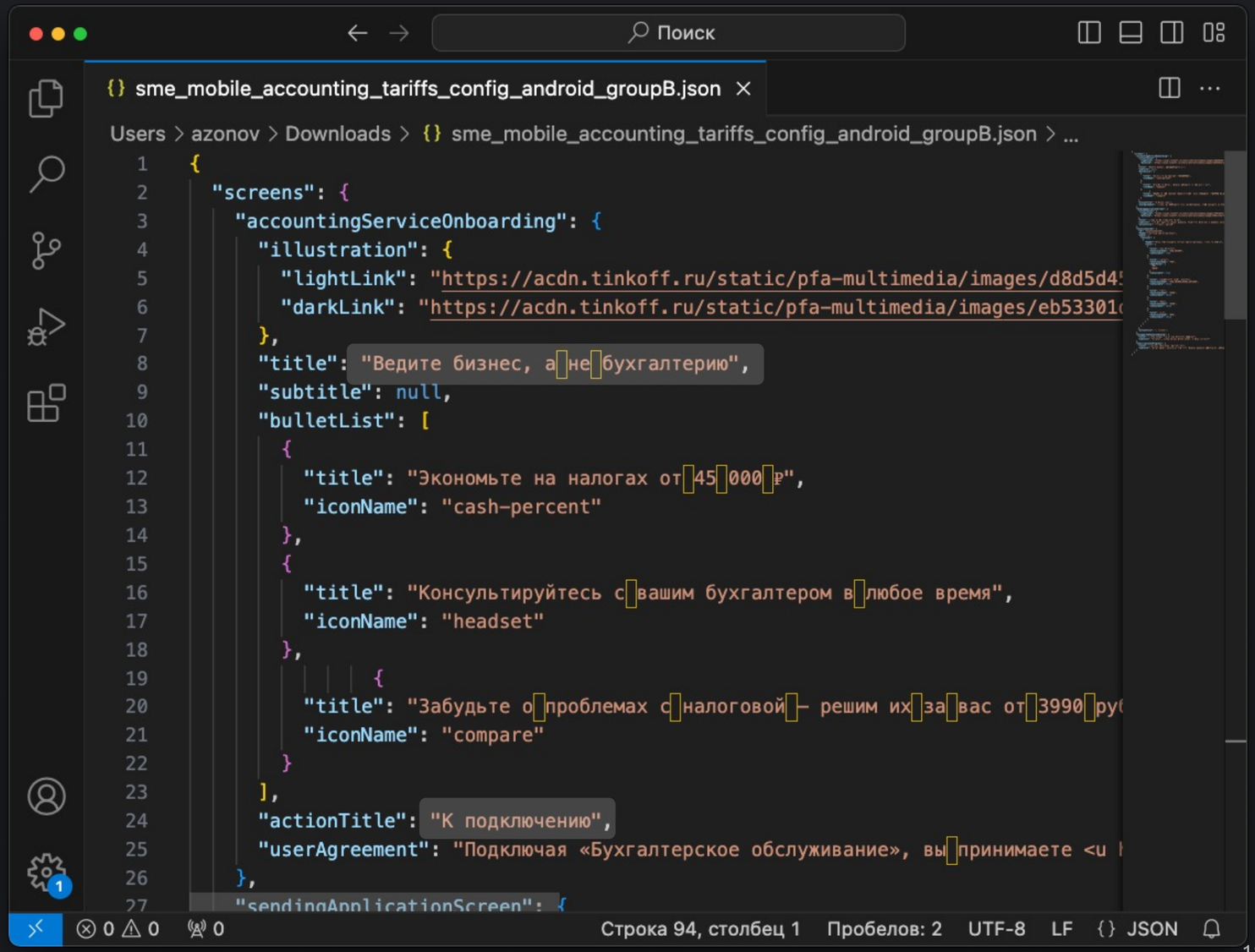
Вы представляете что видит пользователь?





Ограничения на поля



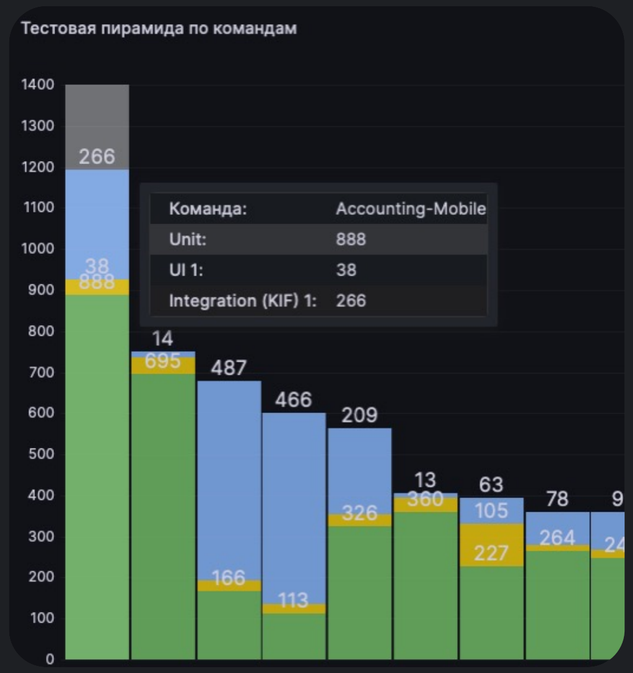


Внедрение валидации контрактов

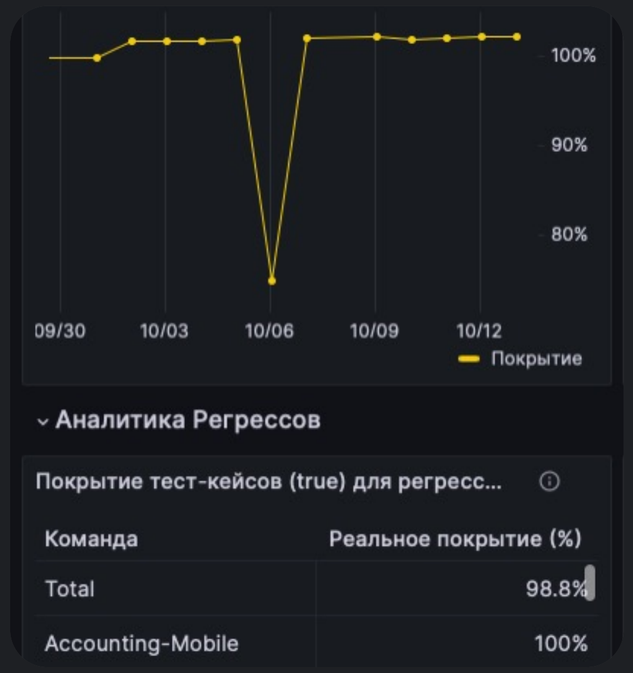
- Контракты
- Тесты
- Вероятность ошибки парсинга
- Только для конфигов
- Сложно понять что меняешь

Внедряем кодогенерацию

Тестирование



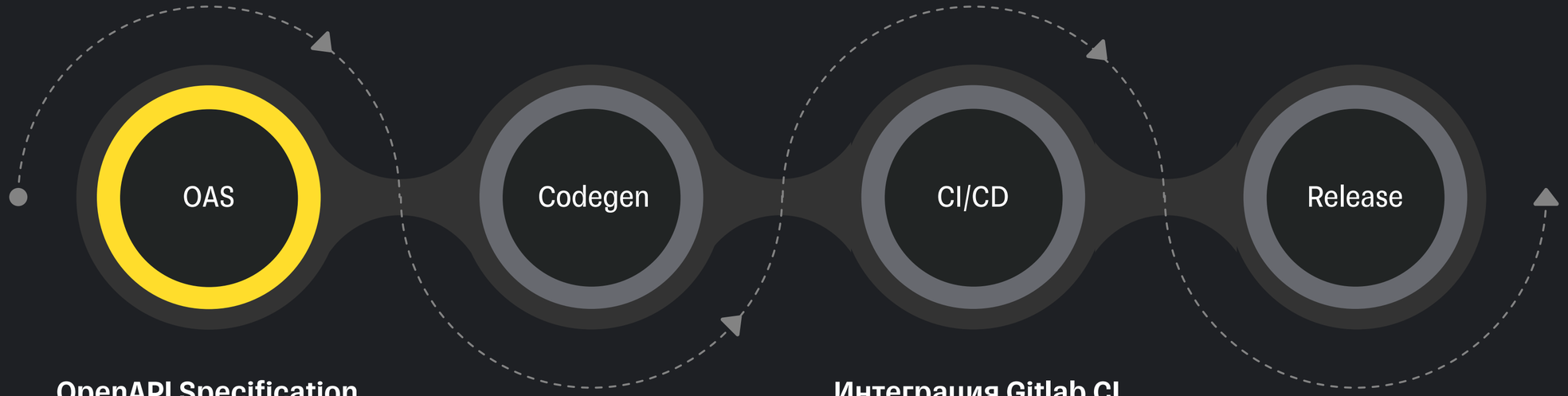
Тестовая пирамида



Покрытие

Стабильность

Разработка через кодогенерацию



OpenAPI Specification

Как единственный источник правды для клиента и сервера

API сразу после спеки

Генерация замоканного API на сервере по спекам для параллельной разработки клиента и сервера

Интеграция Gitlab CI

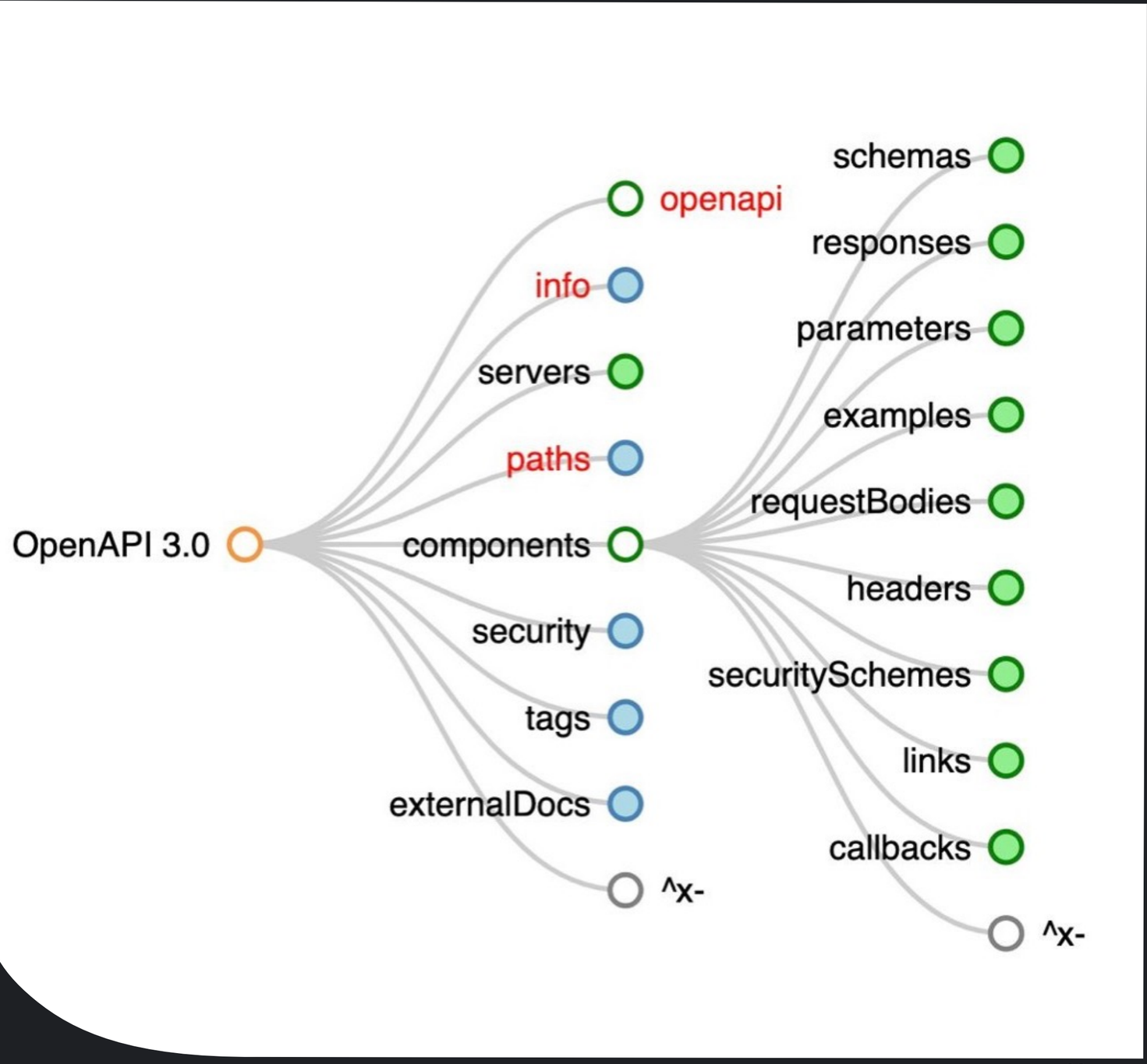
Pipeline который при изменении файла спецификации API будет перегенерировать файлы для сервера и клиентов и говорить есть ли проблемы с совместимостью

TBD

Контрактное тестирование из коробки доступно как потребителям так и создателям API, спим спокойно

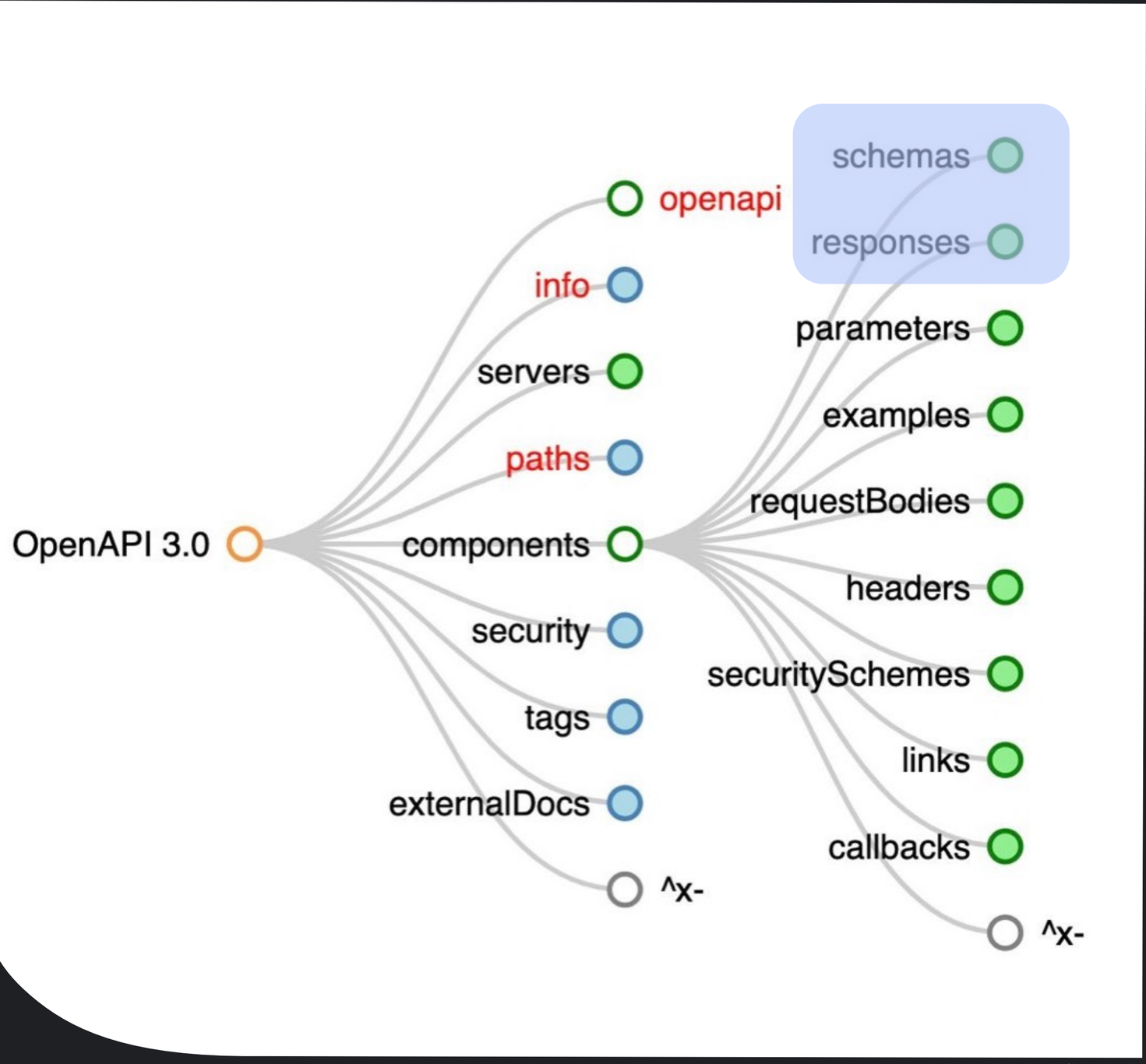
OpenAPI

Документ YAML OpenAPI представляет собой дерево с набором ветвей



OpenAPI

Документ YAML OpenAPI представляет собой дерево с набором ветвей



OpenAPI

```yaml
Category:
  type: object
  properties:
    id:
      type: integer
      format: int64
      example: 1
    name:
      type: string
      example: Dogs
  xml:
    name: category
```

Типы

string, number, integer,

boolean, object, array

OpenAPI

```yaml
Category:
  type: object
  properties:
    id:
      type: integer
      format: int64
      example: 1
    name:
      type: string
      example: Dogs
  xml:
    name: category
```

```yaml
/users/{userId}:
  get:
    parameters:
      - name: userId
        in: path
        required: true
        description: "ID o
        allowEmptyValue: t
        schema:
          type: integer
```

Типы

string, number, integer,

boolean, object, array

Возможность задания ограничений

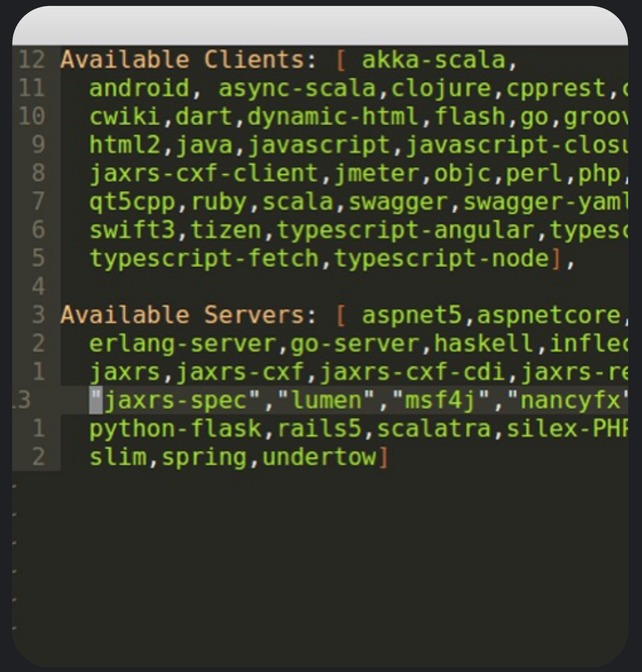
max, min, pattern, format, length,

nullable, enum, required

OpenAPI

```yaml
Category:
  type: object
  properties:
    id:
      type: integer
      format: int64
      example: 1
    name:
      type: string
      example: Dogs
  xml:
    name: category
```

```yaml
/users/{userId}:
  get:
    parameters:
      - name: userId
        in: path
        required: true
        description: "ID o
        allowEmptyValue: t
        schema:
          type: integer
```



Типы

string, number, integer,

boolean, object, array

Возможность задания ограничений

max, min, pattern, format, length,

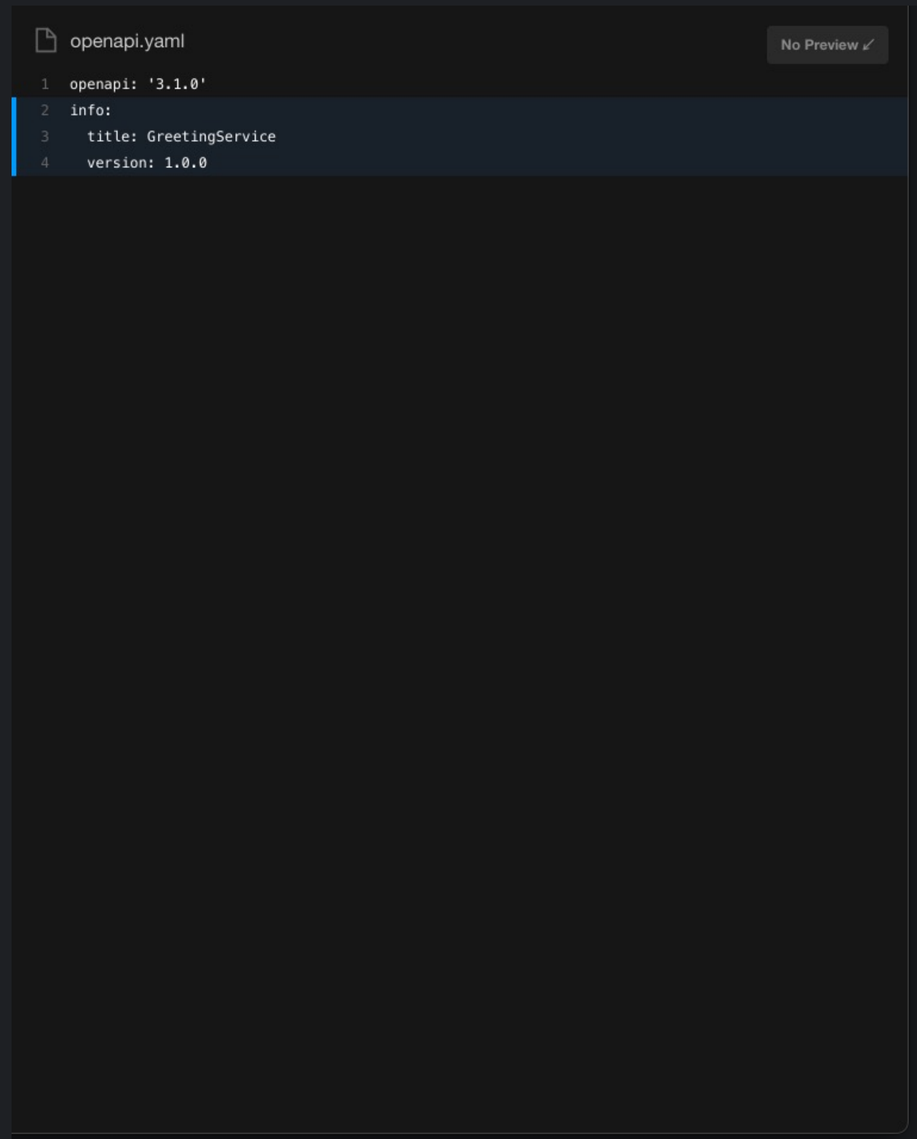
nullable, enum, required

Кодогенерация

Jar исполняемый cli с кастомизацией

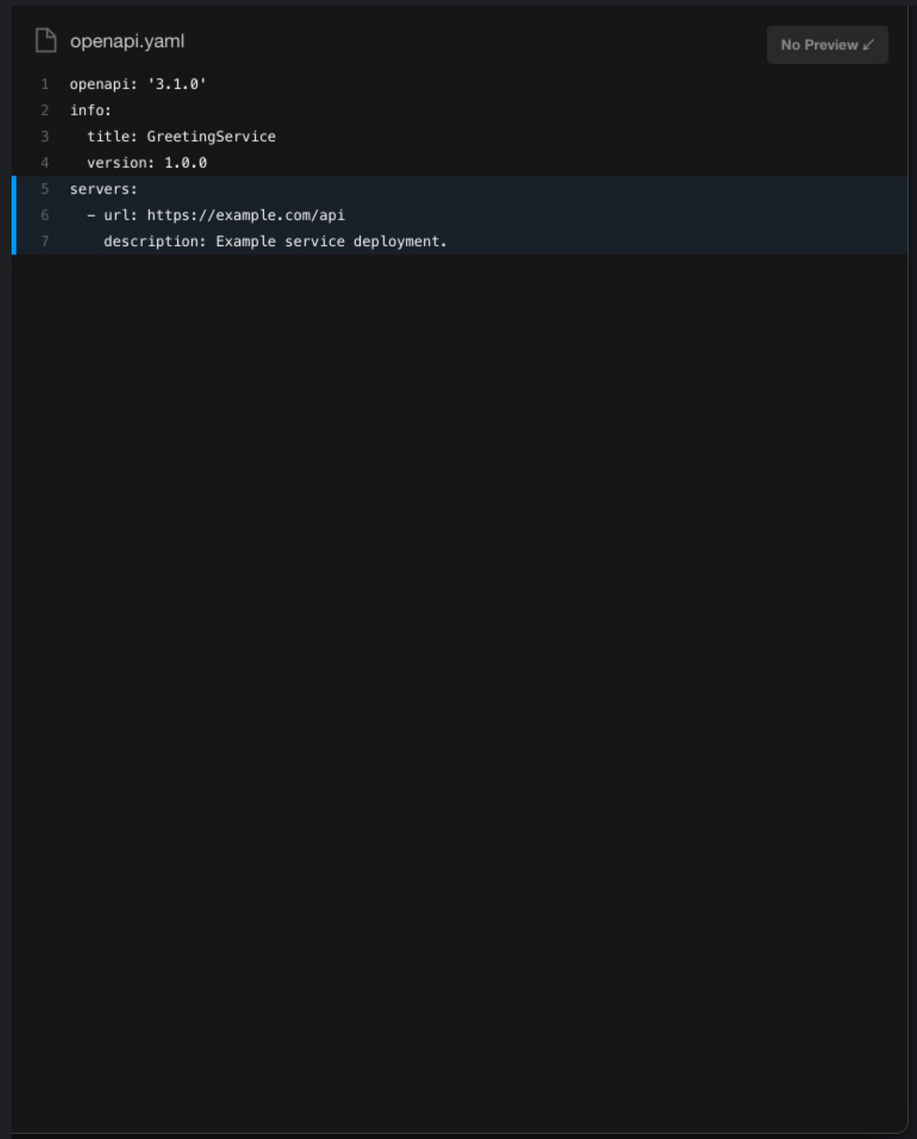
через mustache

OpenAPI



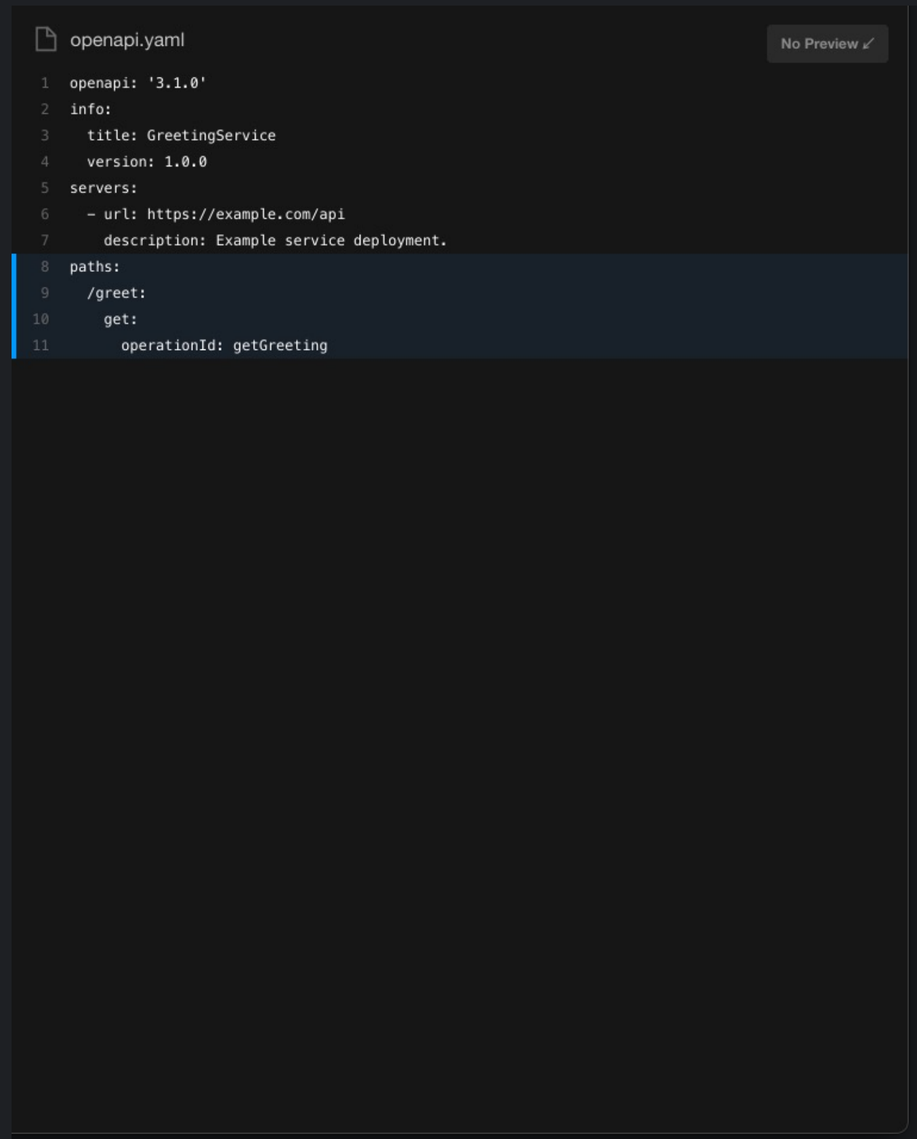
- Версия API и название

OpenAPI



- Версия API и название
- Список URL сервера

OpenAPI



- Версия API и название
- Список URL сервера
- Список путей

OpenAPI

openapi.yaml

No Preview

```yaml
openapi: '3.1.0'
info:
  title: GreetingService
  version: 1.0.0
servers:
  - url: https://example.com/api
    description: Example service deployment.
paths:
  /greet:
    get:
      operationId: getGreeting
      parameters:
      - name: name
        required: false
        in: query
        description: The name used in the returned greeting.
        schema:
          type: string
```

- Версия API и название
- Список URL сервера
- Список путей
- Параметры запроса

OpenAPI

openapi.yaml

No Preview

```yaml
openapi: '3.1.0'
info:
  title: GreetingService
  version: 1.0.0
servers:
  - url: https://example.com/api
    description: Example service deployment.
paths:
  /greet:
    get:
      operationId: getGreeting
      parameters:
      - name: name
        required: false
        in: query
        description: The name used in the returned greeting.
        schema:
          type: string
      responses:
        '200':
          description: A success response with a greeting.
```

- Версия API и название
- Список URL сервера
- Список путей
- Параметры запроса
- Коды ответов

OpenAPI

openapi.yaml — No Preview

```yaml
openapi: '3.1.0'
info:
  title: GreetingService
  version: 1.0.0
servers:
  - url: https://example.com/api
    description: Example service deployment.
paths:
  /greet:
    get:
      operationId: getGreeting
      parameters:
      - name: name
        required: false
        in: query
        description: The name used in the returned greeting.
        schema:
          type: string
      responses:
        '200':
          description: A success response with a greeting.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Greeting'
components:
  schemas:
    Greeting:
      type: object
      properties:
        message:
          type: string
      required:
        - message
```

- Версия API и название
- Список URL сервера
- Список путей
- Параметры запроса
- Коды ответов
- Тело ответа

OpenAPI

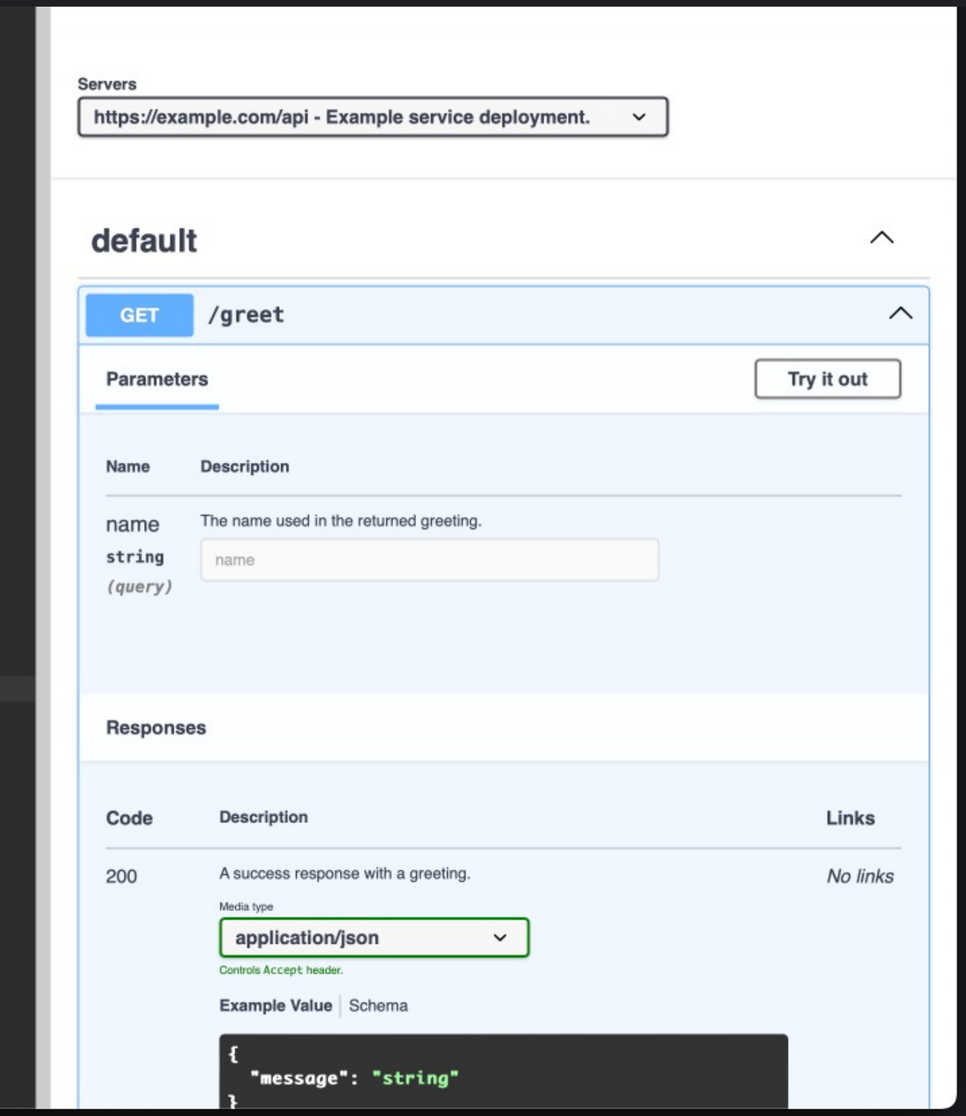
openapi.yaml — No Preview

```yaml
openapi: '3.1.0'
info:
  title: GreetingService
  version: 1.0.0
servers:
  - url: https://example.com/api
    description: Example service deployment.
paths:
  /greet:
    get:
      operationId: getGreeting
      parameters:
      - name: name
        required: false
        in: query
        description: The name used in the returned greeting.
        schema:
          type: string
      responses:
        '200':
          description: A success response with a greeting.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Greeting'
components:
  schemas:
    Greeting:
      type: object
      properties:
        message:
          type: string
      required:
        - message
```

- Версия API и название
- Список URL сервера
- Список путей
- Параметры запроса
- Коды ответов
- Тело ответа

OpenAPI

```yaml
paths:
  /greet:
    get:
      operationId: getGreeting
      parameters:
      - name: name
        required: false
        in: query
        description: The name used in the returned greeting.
        schema:
          type: string
      responses:
        '200':
          description: A success response with a greeting.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Greeting'
components:
  schemas:
    Greeting:
      type: object
      properties:
        message:
          type: string
      required:
        - message
```



Codegen

OpenAPI

main.swift

```swift
import Foundation
import Vapor
import OpenAPIRuntime
import OpenAPIVapor

// Define a type that conforms to the generated protocol.
struct GreetingServiceAPIImpl: APIProtocol {
    func getGreeting(
        _ input: Operations.getGreeting.Input
    ) async throws -> Operations.getGreeting.Output {
        let name = input.query.name ?? "Stranger"
        let greeting = Components.Schemas.Greeting(message: "Hello, \(name
        return .ok(.init(body: .json(greeting)))
    }
}
```

Сервер

Codegen

OpenAPI

main.swift

```swift
import Foundation
import Vapor
import OpenAPIRuntime
import OpenAPIVapor

// Define a type that conforms to the generated protocol.
struct GreetingServiceAPIImpl: APIProtocol {
    func getGreeting(
        _ input: Operations.getGreeting.Input
    ) async throws -> Operations.getGreeting.Output {
        let name = input.query.name ?? "Stranger"
        let greeting = Components.Schemas.Greeting(message: "Hello, \(name
        return .ok(.init(body: .json(greeting)))
    }
}
```

Сервер

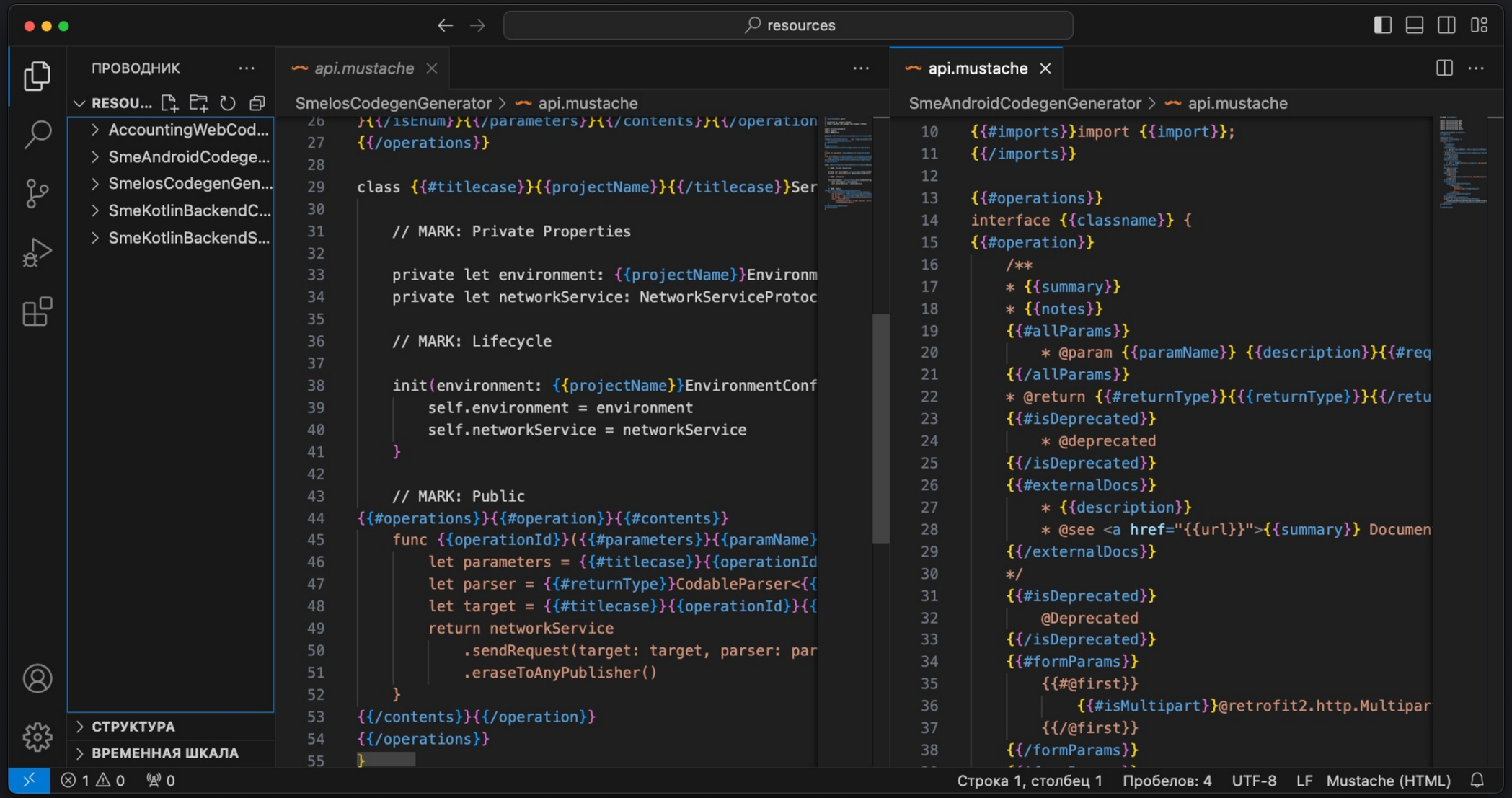
main.swift

No Pr

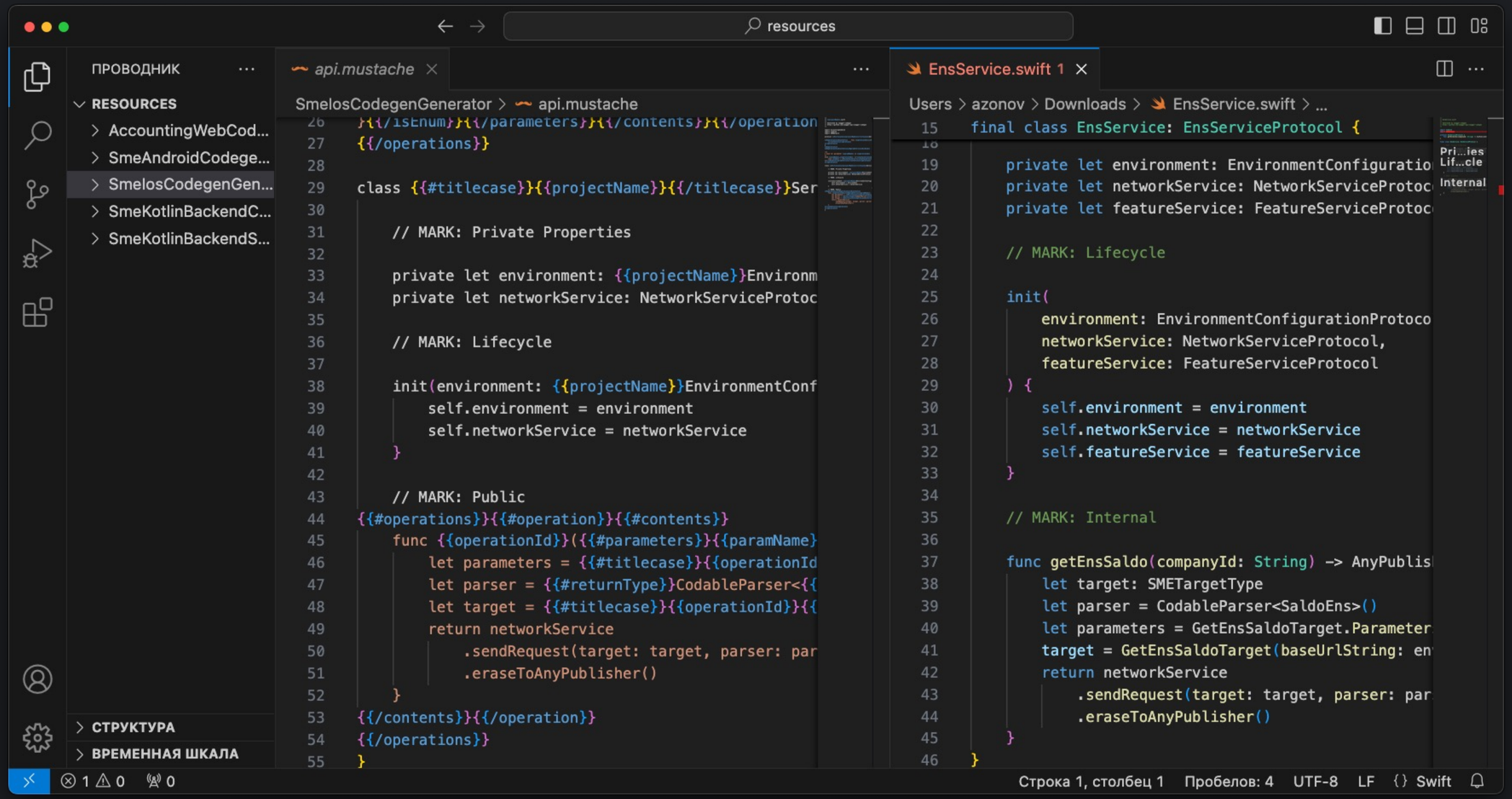
```swift
import OpenAPIRuntime
import OpenAPIURLSession

let client = Client(
    serverURL: try Servers.server2(),
    transport: URLSessionTransport()
)

let response = try await client.getGreeting(query: .init(name: "CLI"))
switch response {
case .ok(let okResponse):
    switch okResponse.body {
    case .json(let greeting):
        print(greeting.message)
    }
case .undocumented(statusCode: let statusCode, _):
    print("😵 undocumented response: \(statusCode)")
}
```

Клиент

OpenAPI Codegen



OpenAPI Codegen



OpenAPI Codegen

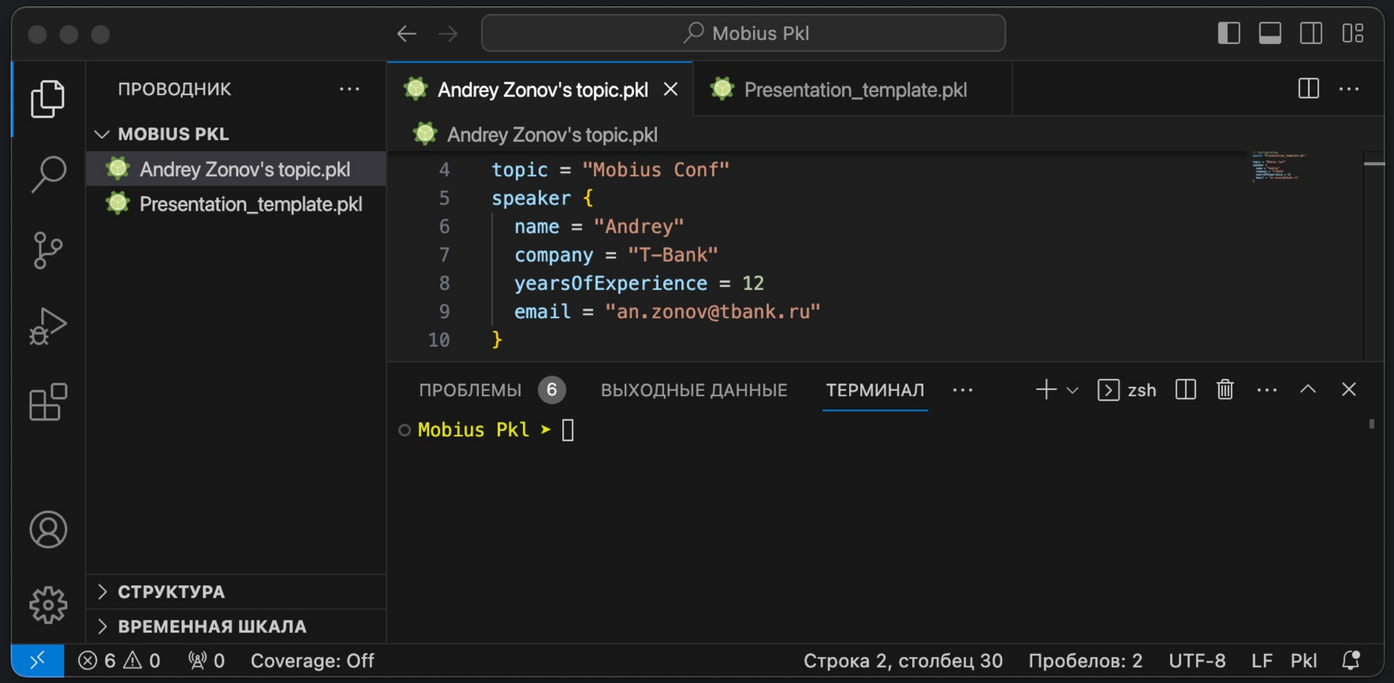
- Единый источник правды
- Не тратим времени на бойлерплейт
- Работа по-контрактам

OpenAPI Codegen

- Множество инструментов
- Не поддерживает валидацию из коробки
- Не для конфигов

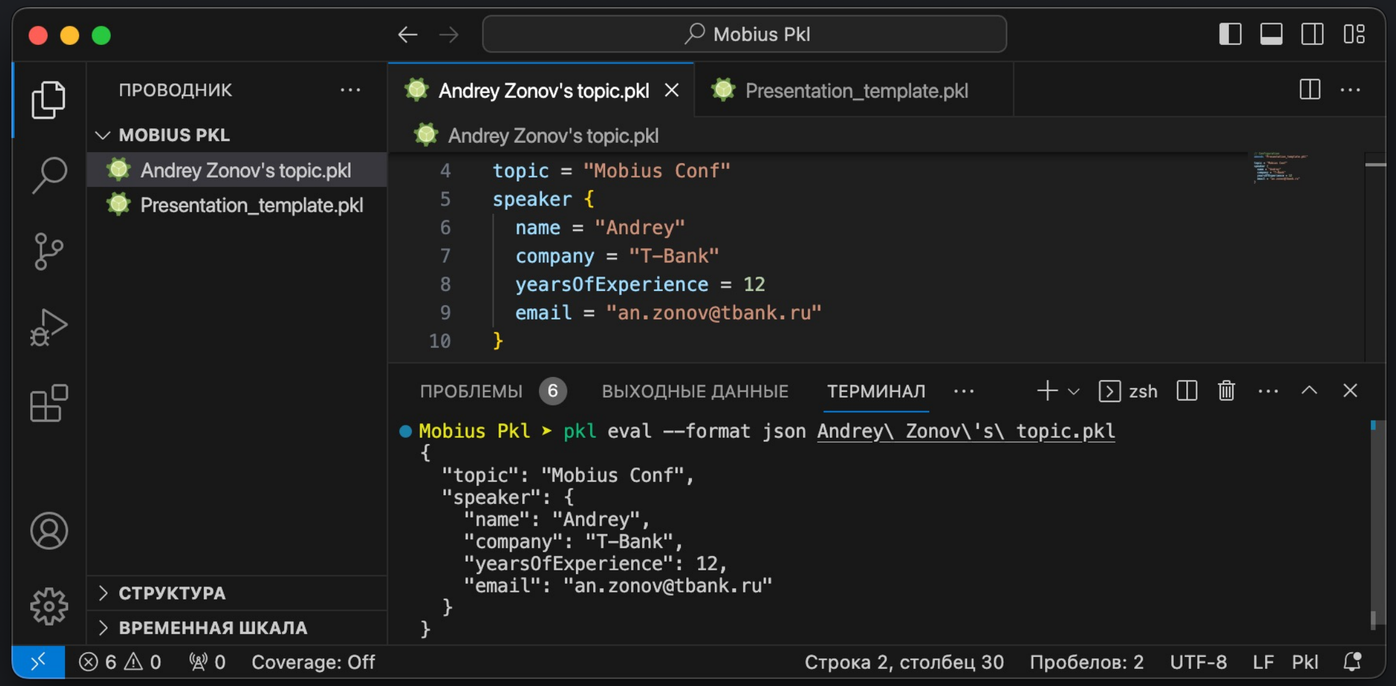
Pkl

Генерация статической конфигурации



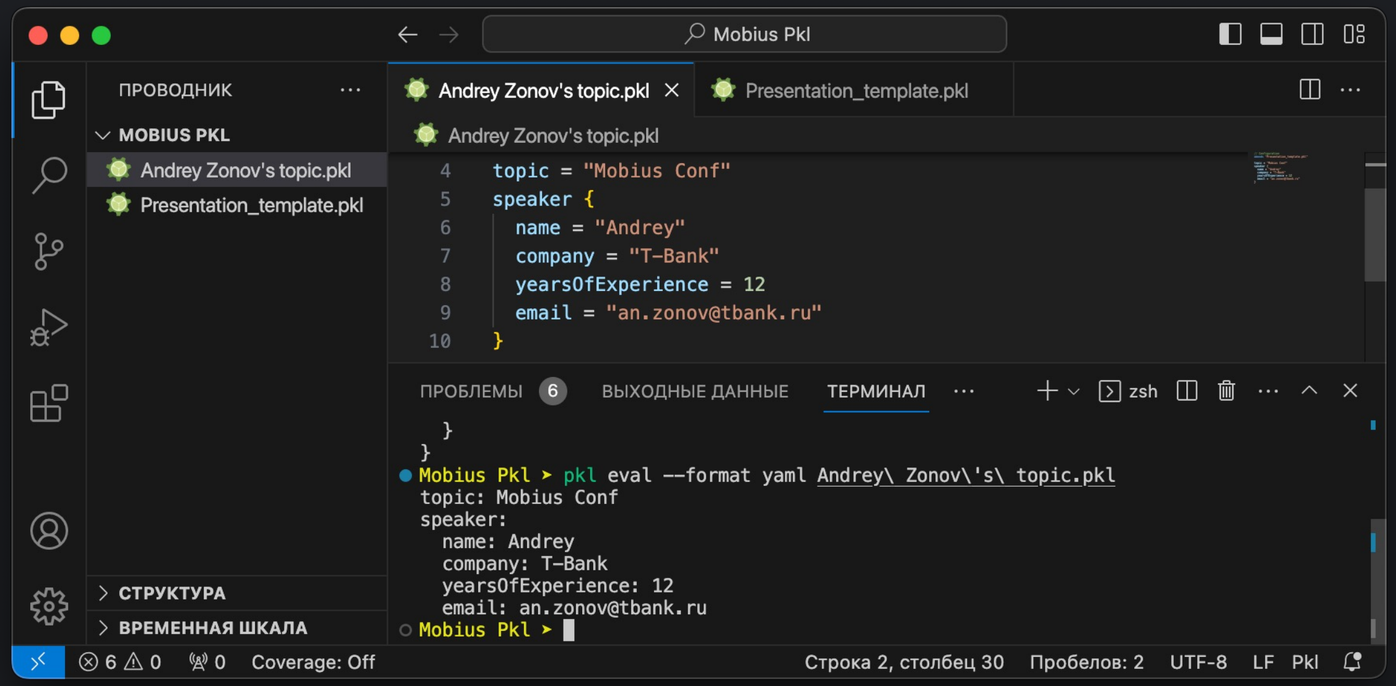
- Определяем типизированный формат конфигурации

Генерация статической конфигурации



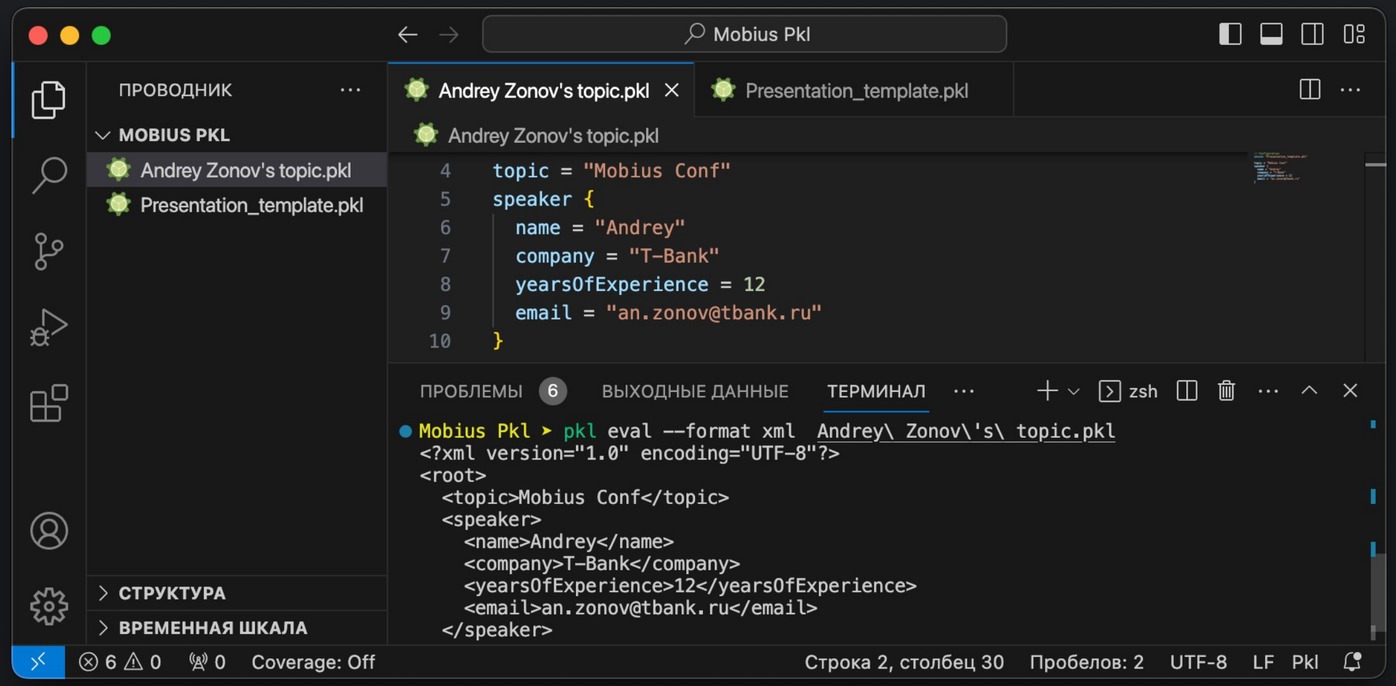
- Определяем типизированный формат конфигурации
- Генерируем данные для транспортного уровня в любом удобном формате

Генерация статической конфигурации



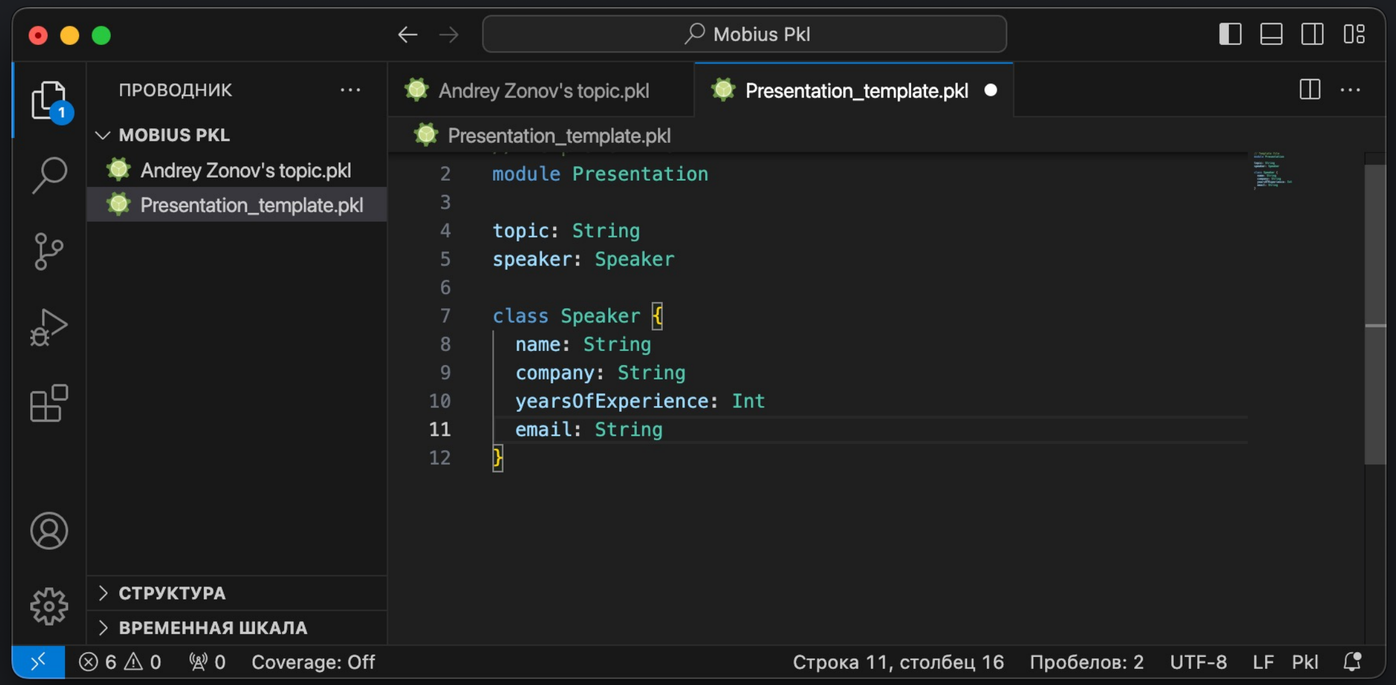
- Определяем типизированный формат конфигурации
- Генерируем данные для транспортного уровня в любом удобном формате

Генерация статической конфигурации



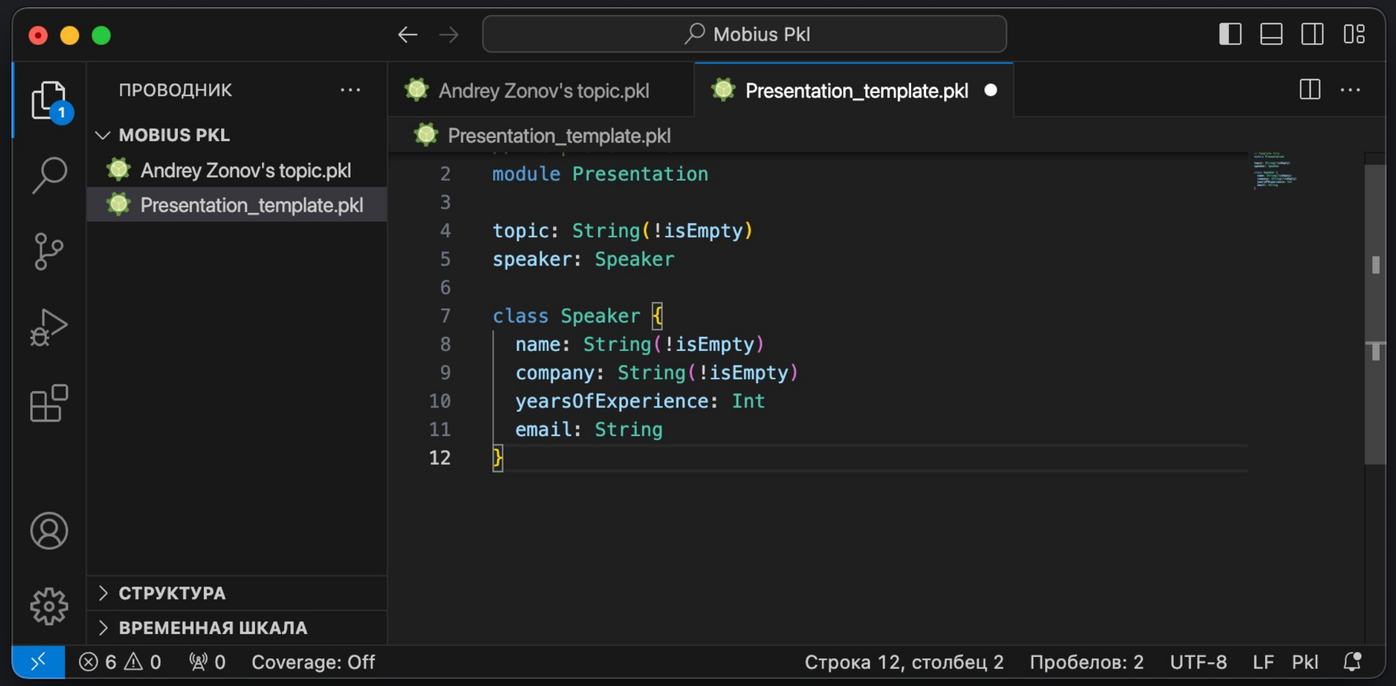
- Определяем типизированный формат конфигурации
- Генерируем данные для транспортного уровня в любом удобном формате

Генерация статической конфигурации



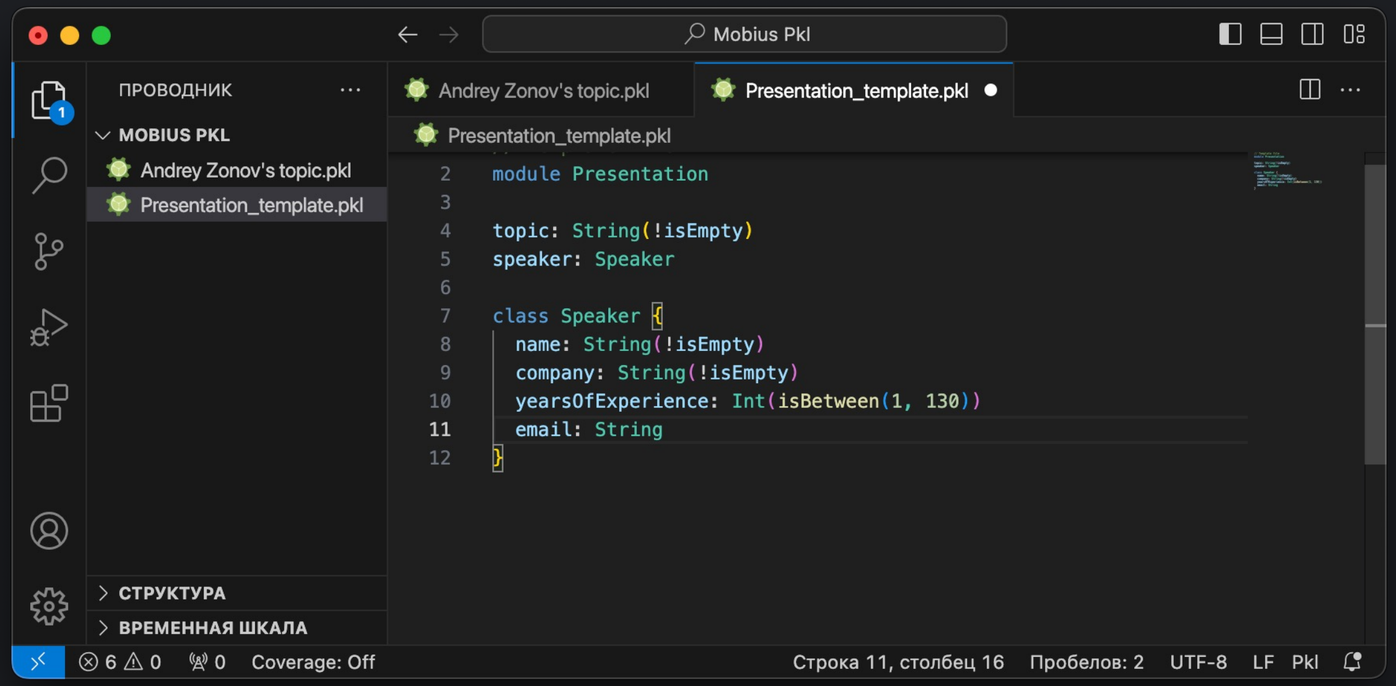
- Определяем типизированный формат конфигурации
- Генерируем данные для транспортного уровня в любом удобном формате
- Определяем темплейт для данных

Генерация статической конфигурации



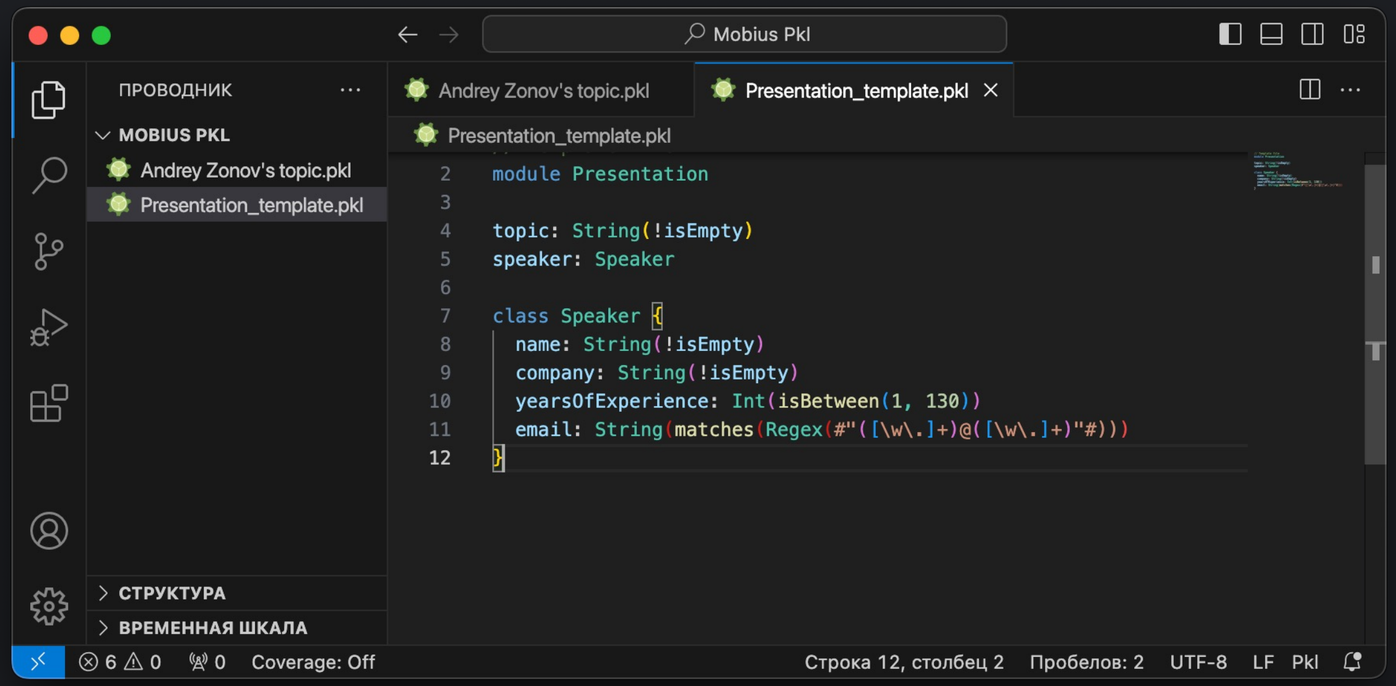
- Определяем типизированный формат конфигурации
- Генерируем данные для транспортного уровня в любом удобном формате
- Определяем темплейт для данных
- Задаем ограничения

Генерация статической конфигурации



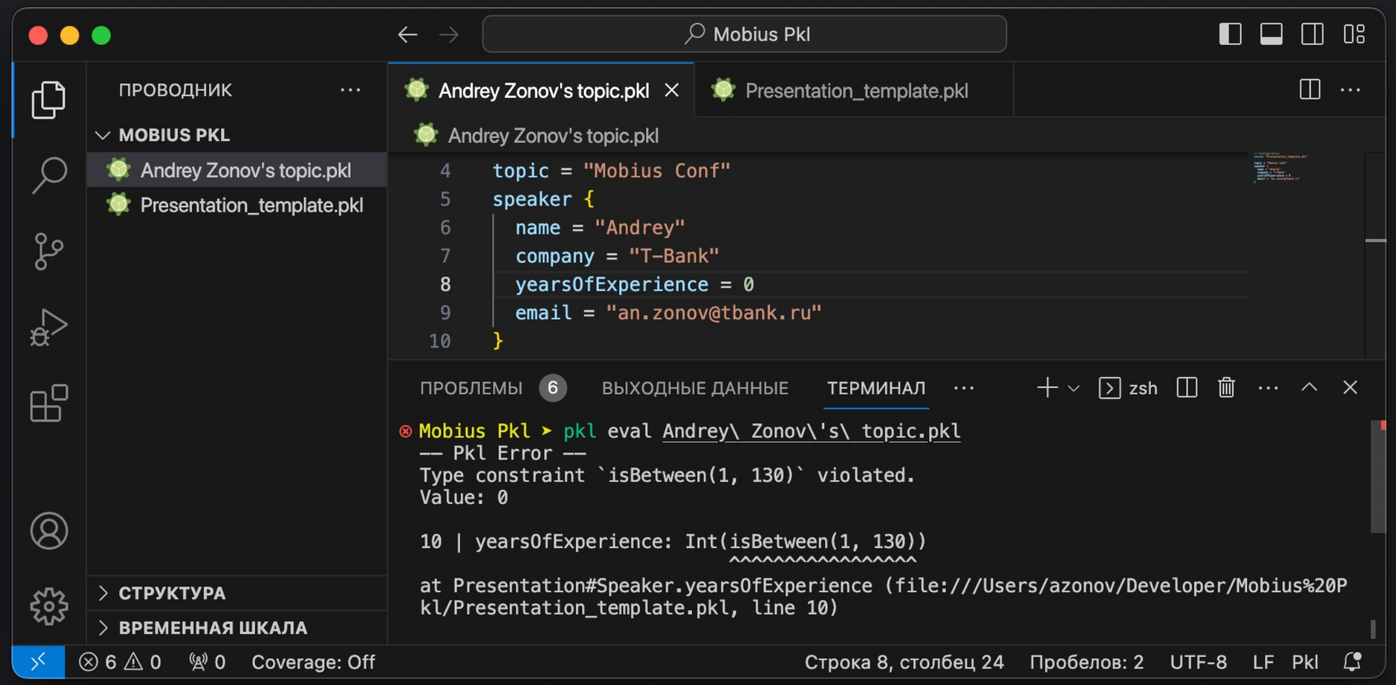
- Определяем типизированный формат конфигурации
- Генерируем данные для транспортного уровня в любом удобном формате
- Определяем темплейт для данных
- Задаем ограничения

Генерация статической конфигурации



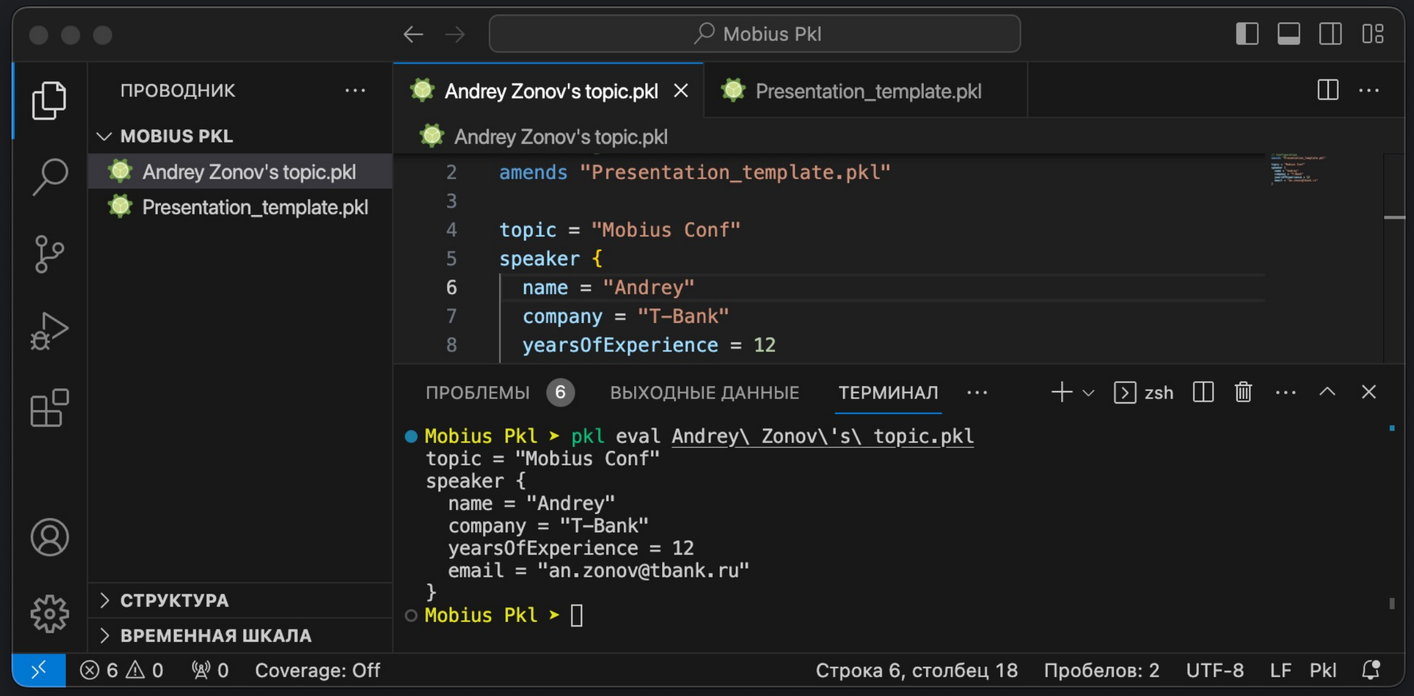
- Определяем типизированный формат конфигурации
- Генерируем данные для транспортного уровня в любом удобном формате
- Определяем темплейт для данных
- Задаем ограничения

Генерация статической конфигурации



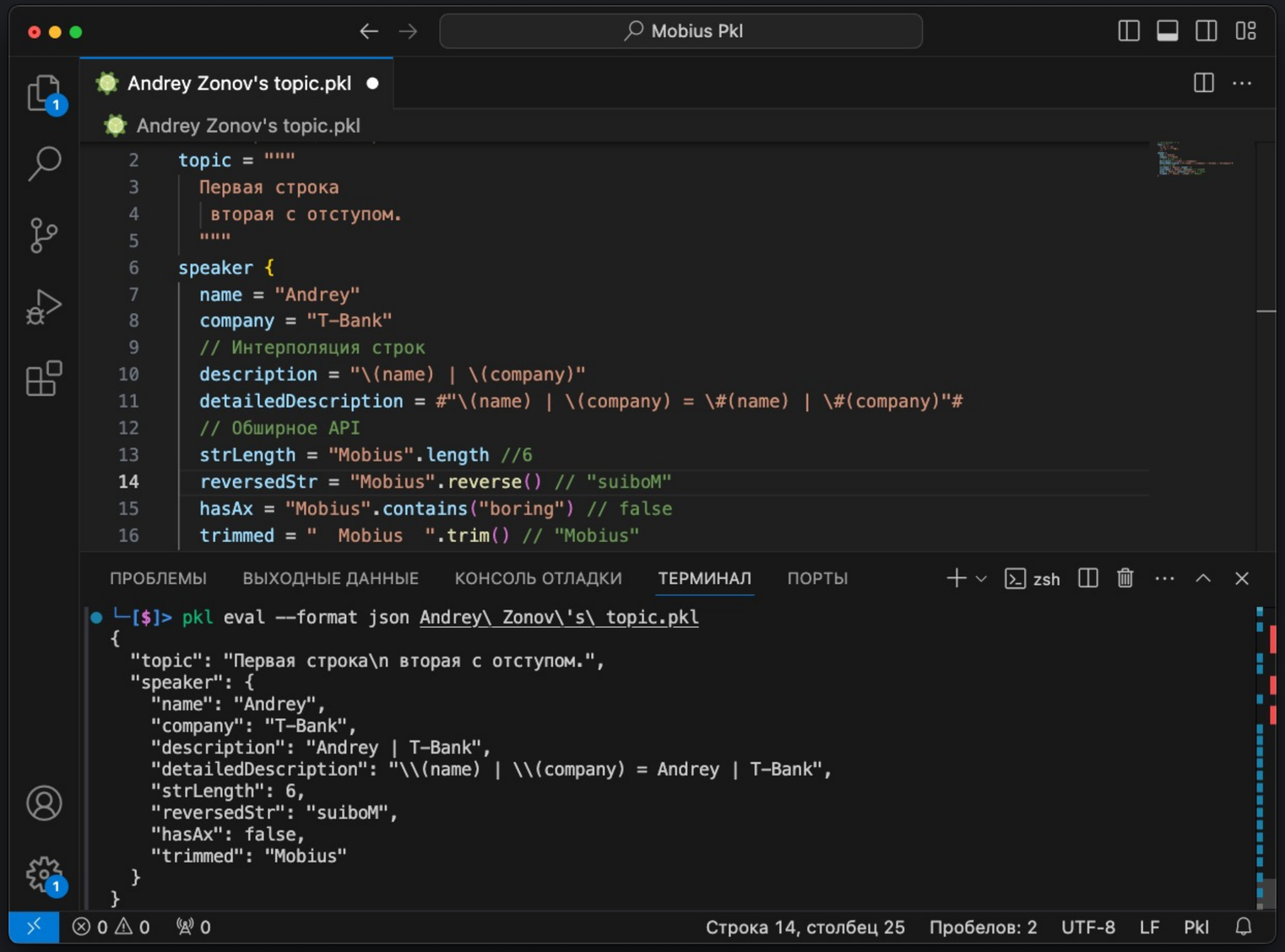
- Определяем типизированный формат конфигурации
- Генерируем данные для транспортного уровня в любом удобном формате
- Определяем темплейт для данных
- Задаем ограничения
- Валидируем

Генерация статической конфигурации



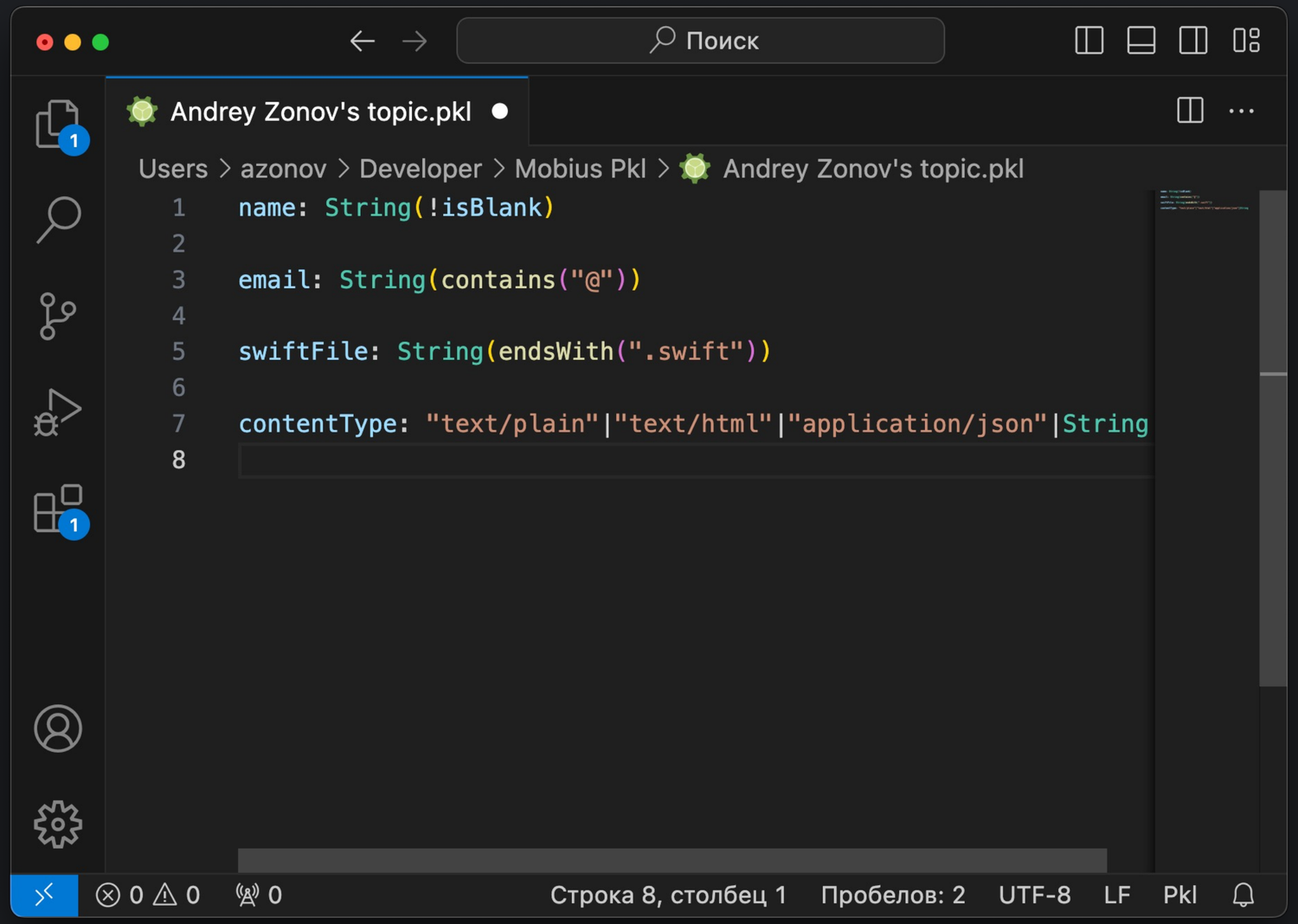
- Определяем типизированный формат конфигурации
- Генерируем данные для транспортного уровня в любом удобном формате
- Определяем темплейт для данных
- Задаем ограничения
- Валидируем

Возможности языка



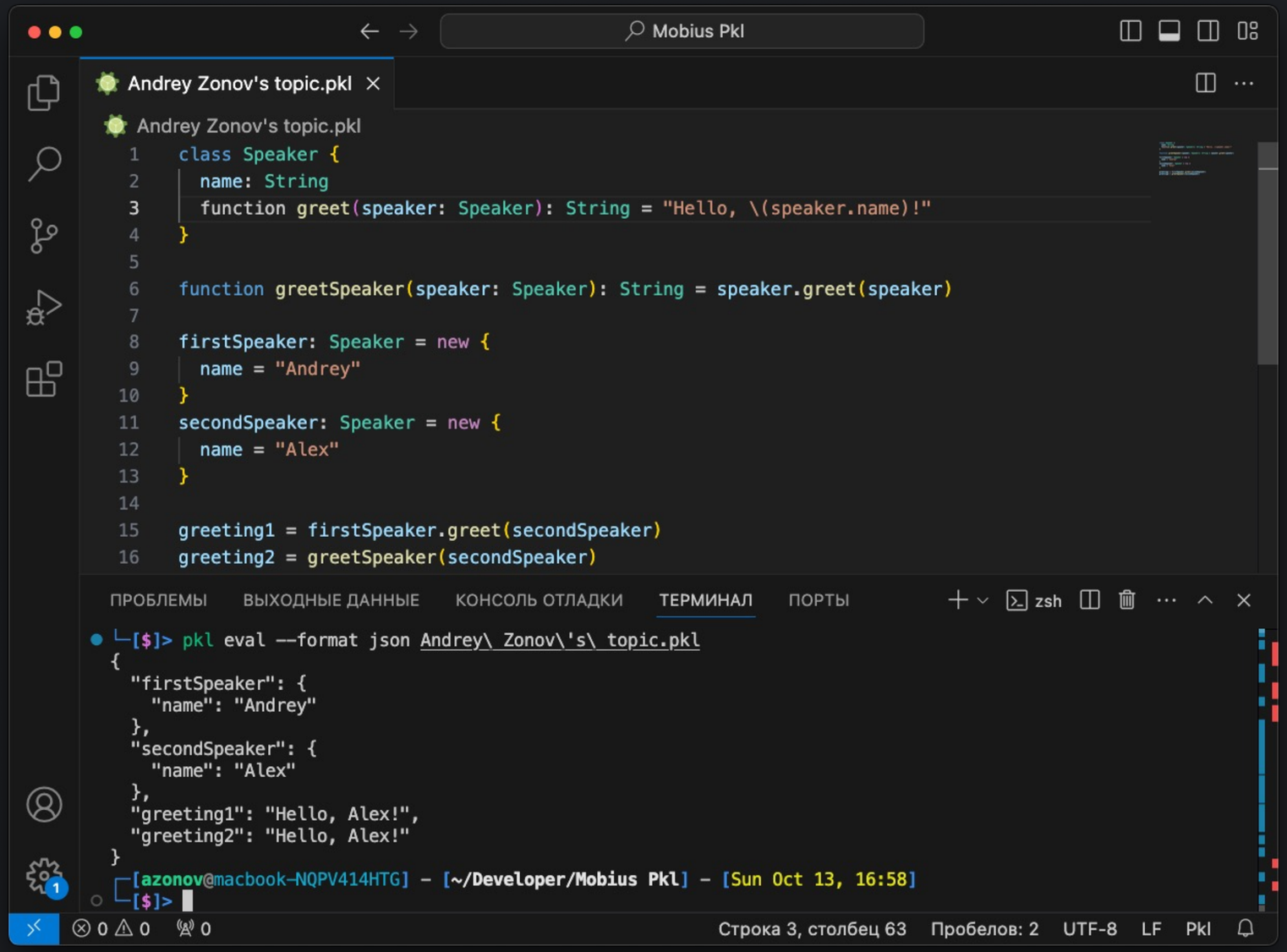
- Удобная работа со строками

Возможности языка



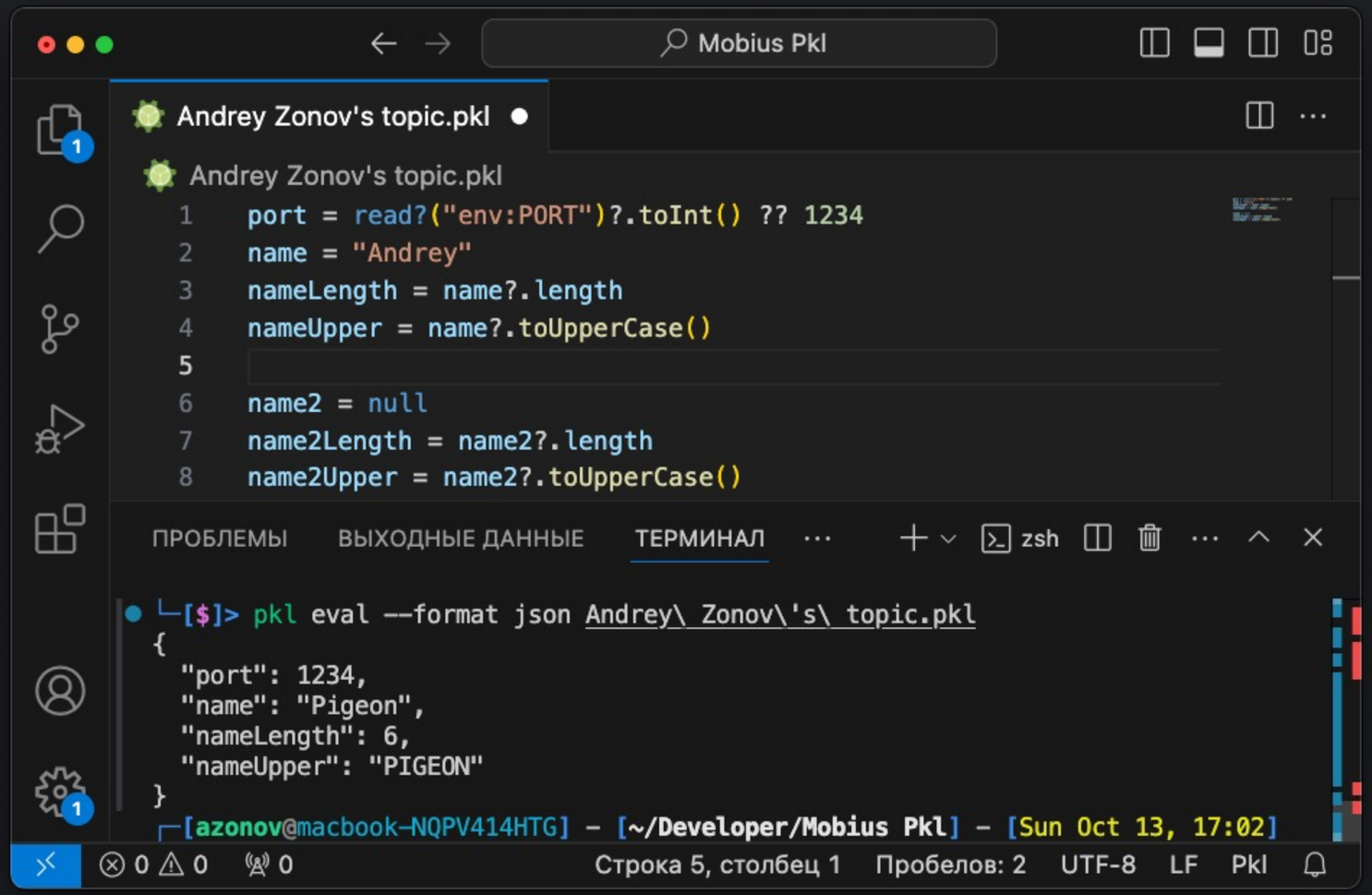
- Удобная работа со строками

Возможности языка



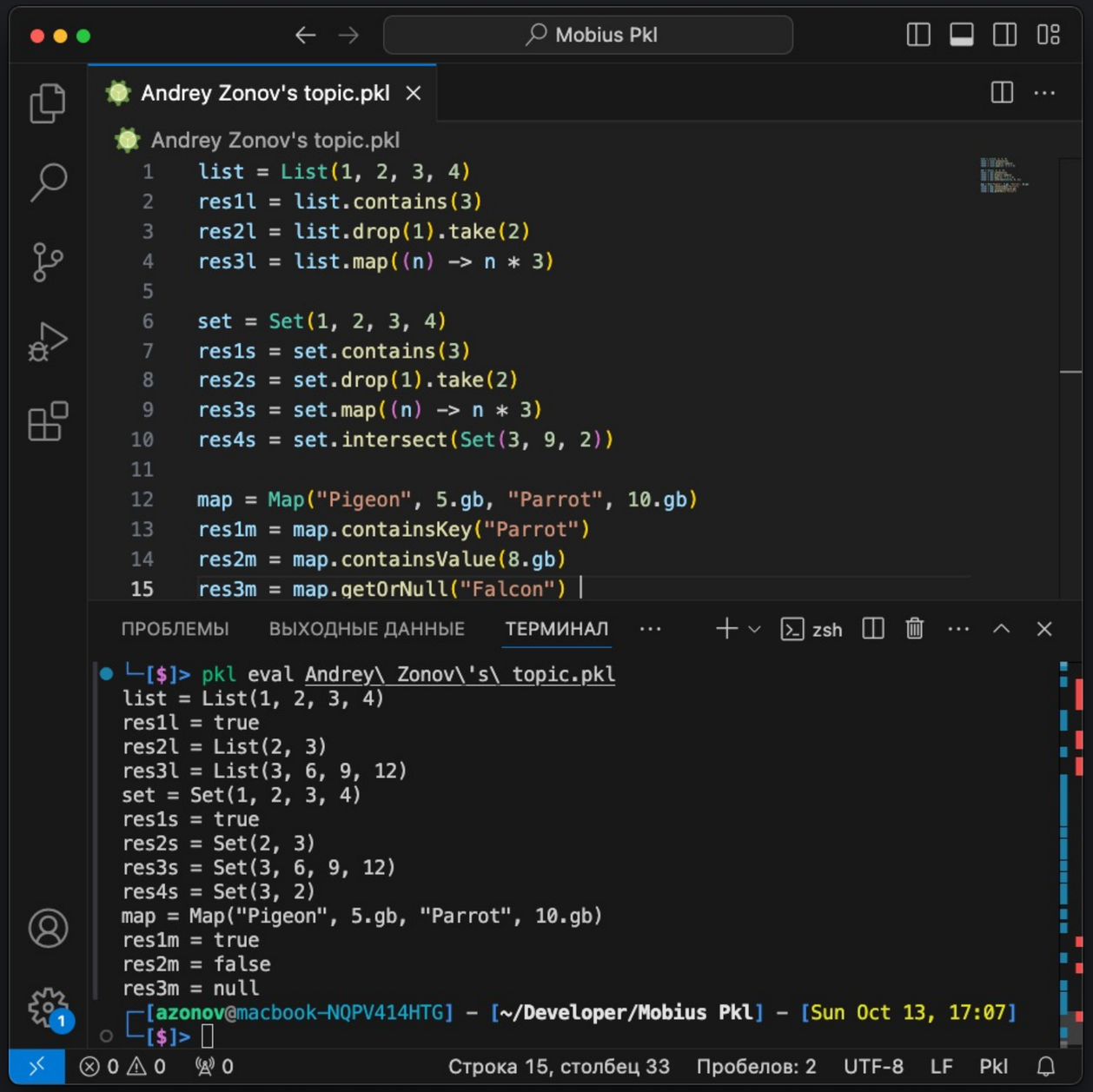
- Удобная работа со строками
- ООП

Возможности языка



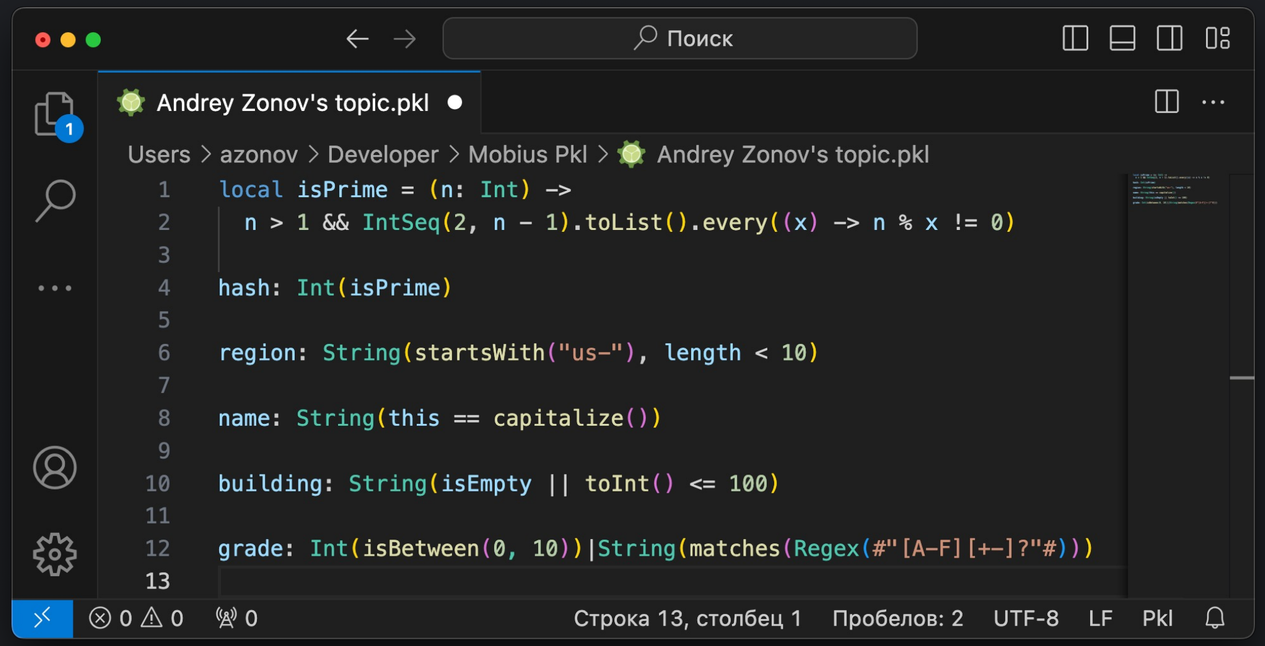
- Удобная работа со строками
- ООП
- Nullable

Возможности языка



- Удобная работа со строками
- ООП
- Nullable
- Коллекции

Возможности языка



- Удобная работа со строками
- ООП
- Nullable
- Коллекции
- Ограничения

Кодогенерация для популярных ЯП

Pkl | Java | Kotlin | Swift | Go

```
module example.MyAppConfig

/// The hostname for the application
host: String

/// The port to listen on
port: UInt16(this > 1000)
```

- Java
- Kotlin
- Swift
- Go

Pkl | Java | Kotlin

```
package example;

public final class MyAppConfig {
  /**
   * The hostname for the applicatio
   */
  public final @NonNull String host;

  /**
   * The port to listen on
   */
```

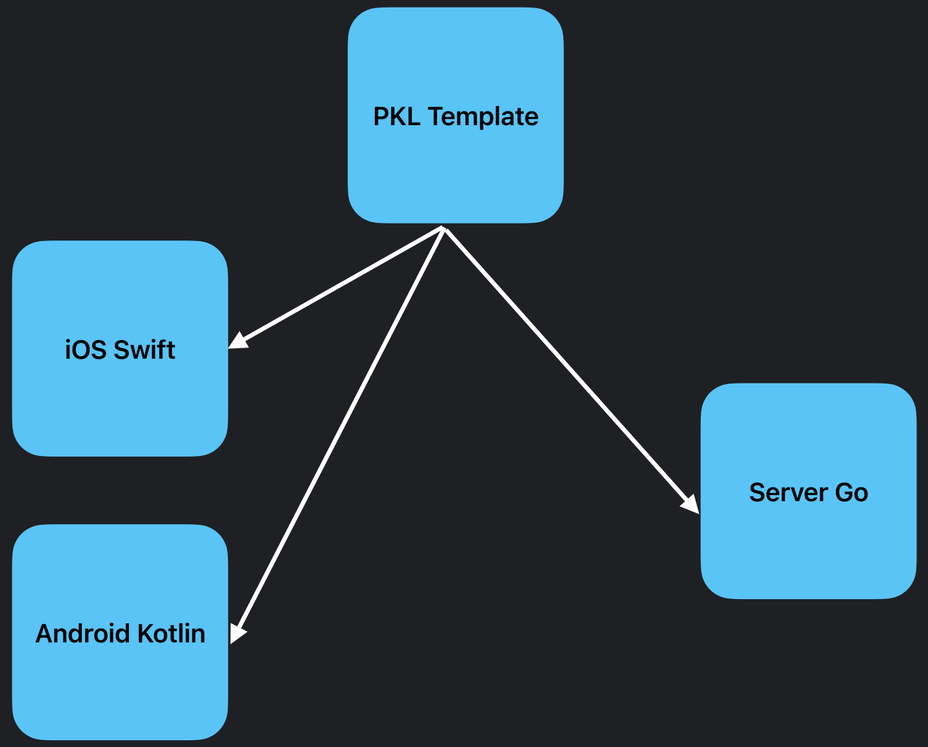
Pkl | Java | Kotlin

```
package example

data class MyAppConfig(
  /**
   * The hostname for the applica
   */
  val host: String,
  /**
   * The port to listen on
   */
  val port: Int
)
```

Pkl | Java | Kotlin | Swift | Go

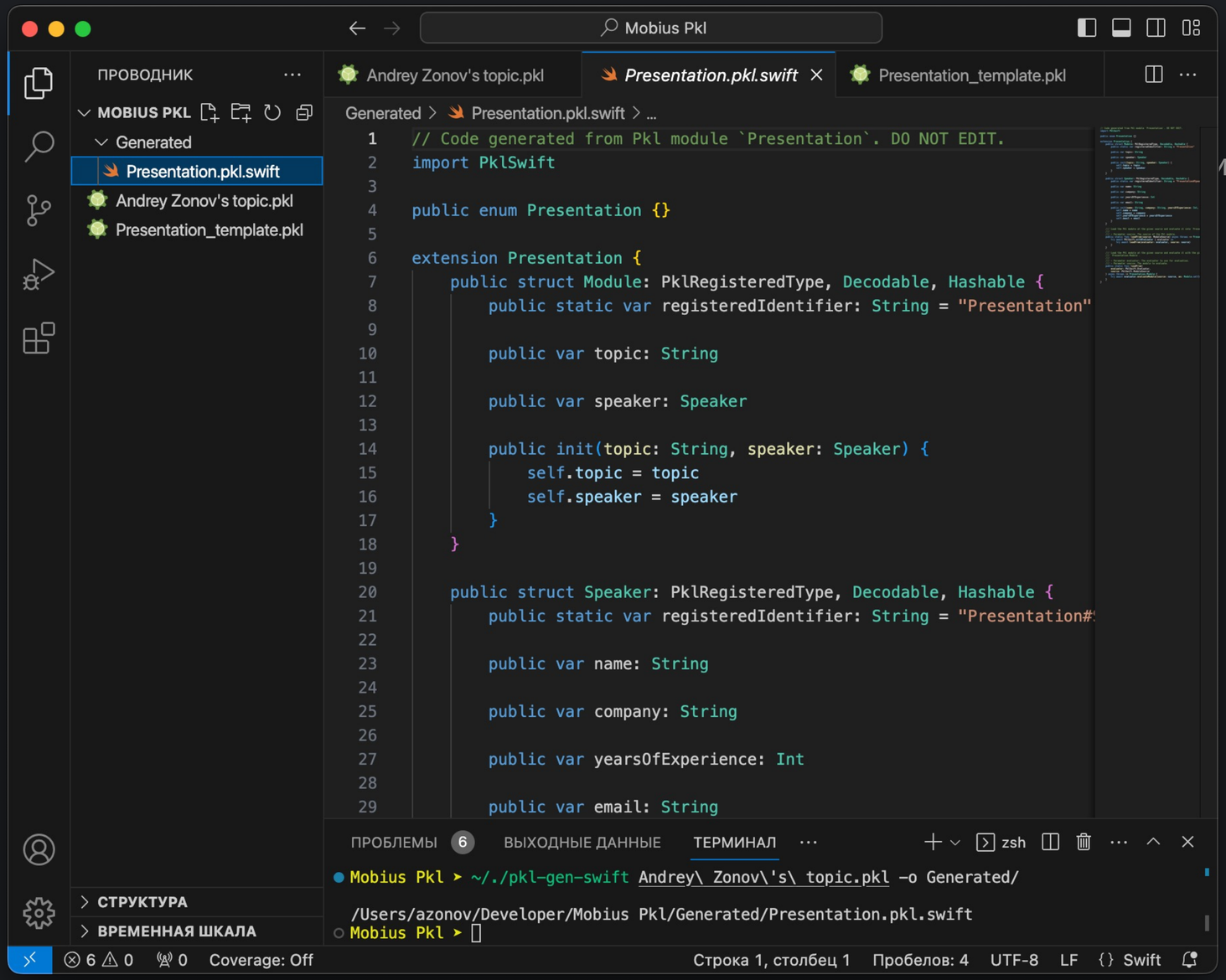
```
enum MyAppConfig {}

extension MyAppConfig {
    struct Module {
        /// The hostname for the application
        let host: String

        /// The port to listen on
        let port: UInt16
    }
}
```

Кодогенерация для популярных ЯП

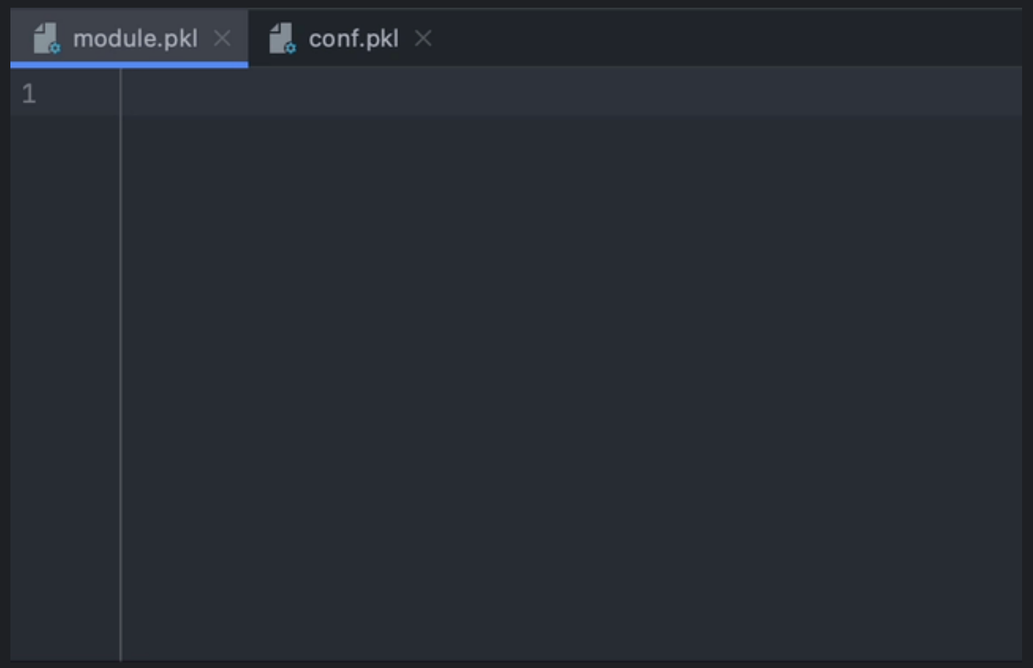


- Java
- Kotlin
- Swift
- Go

Кодогенерация



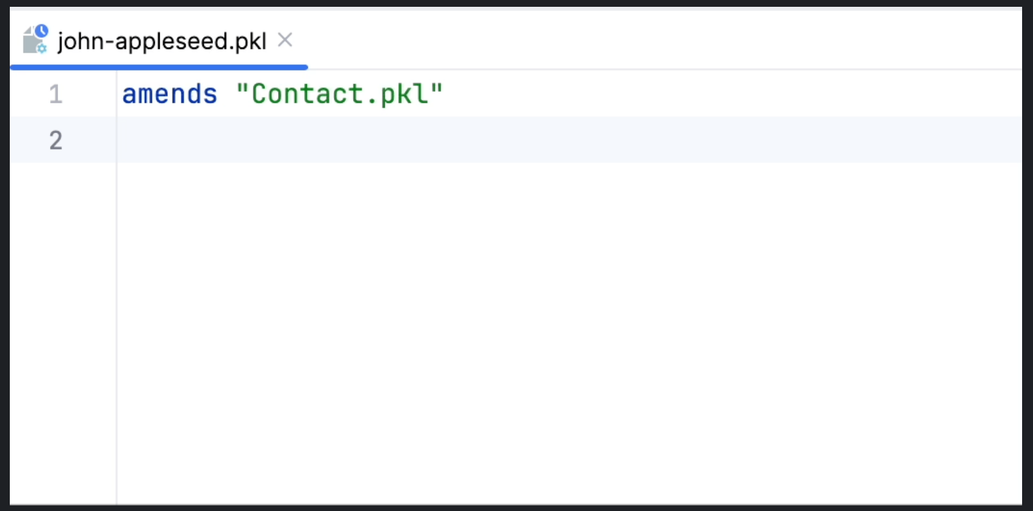
Интеграция с IDE через LSP



- IntelliJ
- VS Code
- Neovim
- Xcode

Валидация

- IntelliJ
- VS Code
- Neovim
- Xcode

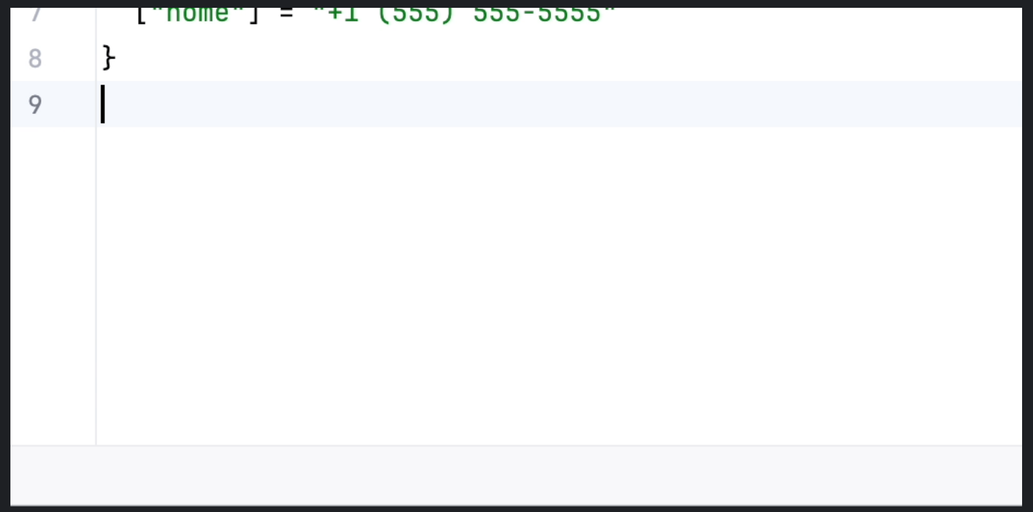


Навигация

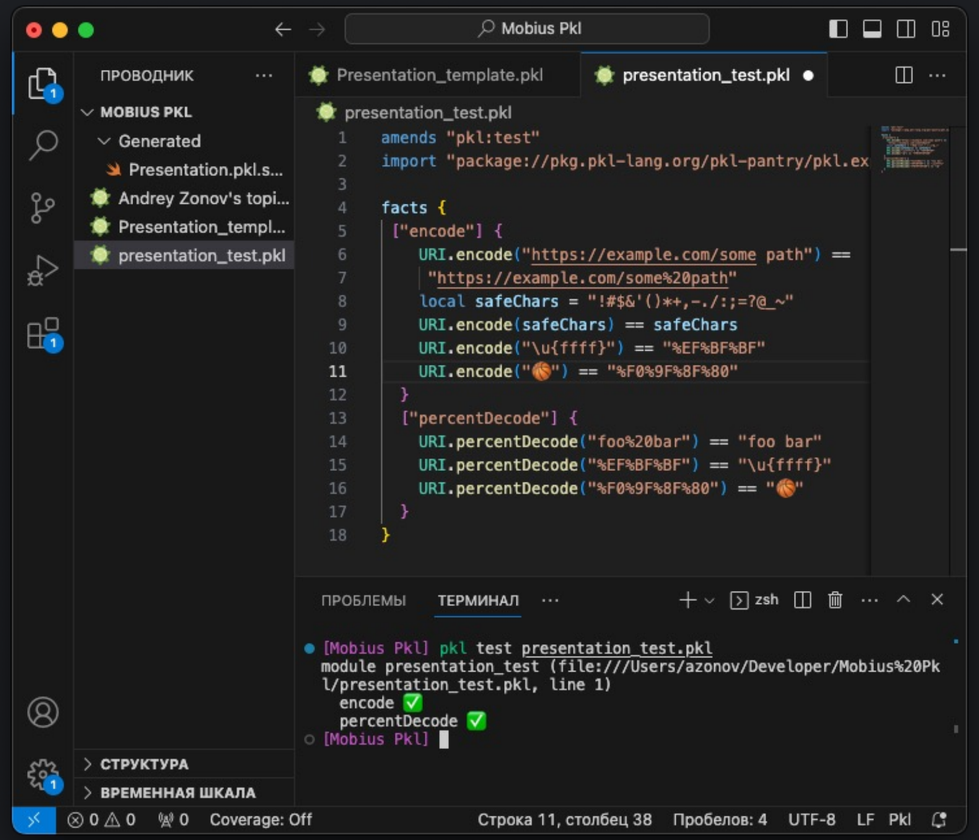
- IntelliJ
- VS Code
- Neovim
- Xcode

Автокомплит

- IntelliJ
- VS Code
- Neovim
- Xcode

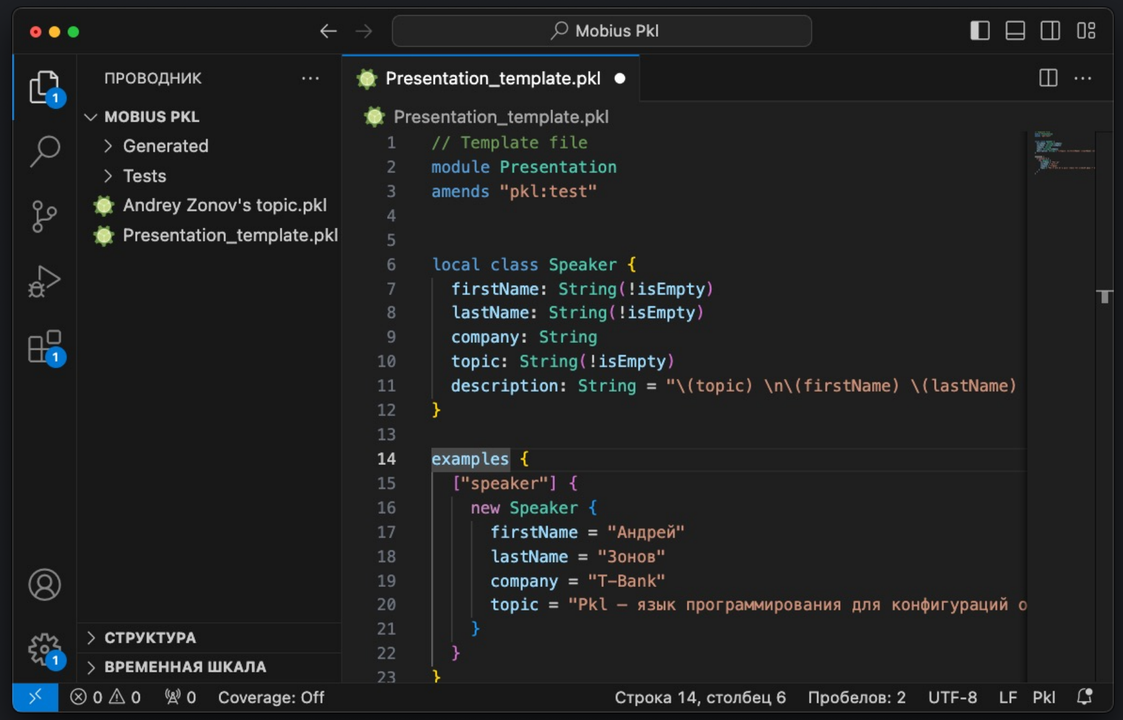


Тестирование



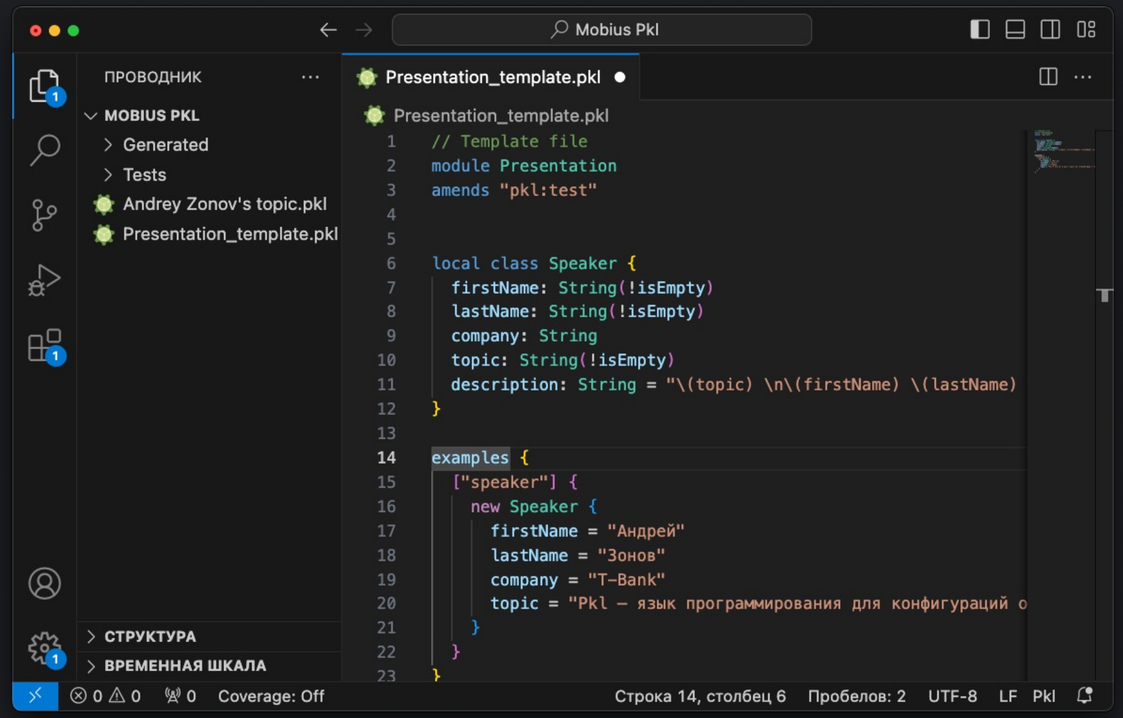
- Тесты поддерживаются из коробки
- Fact – обычный логический ассерт, подходит для юнит тестов

Тестирование

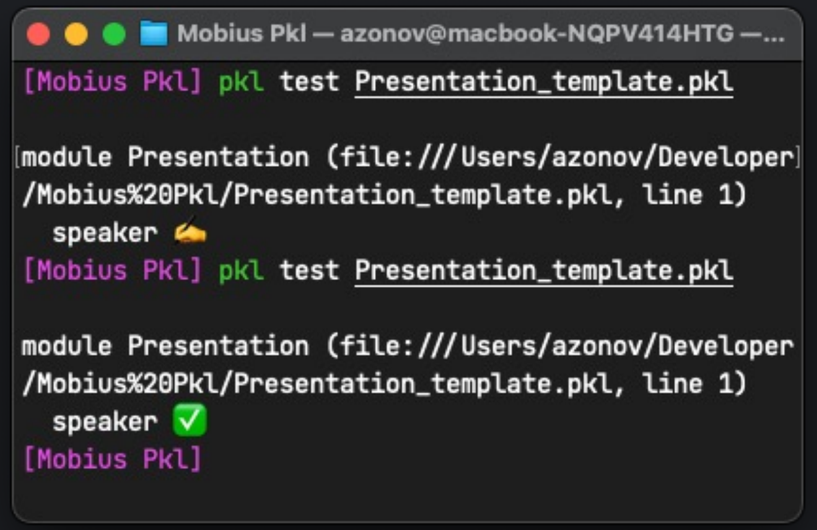


- Тесты поддерживаются из коробки
- Fact – обычный логический ассерт, подходит для юнит тестов
- Example – аналог снапшот тестирования для конфига

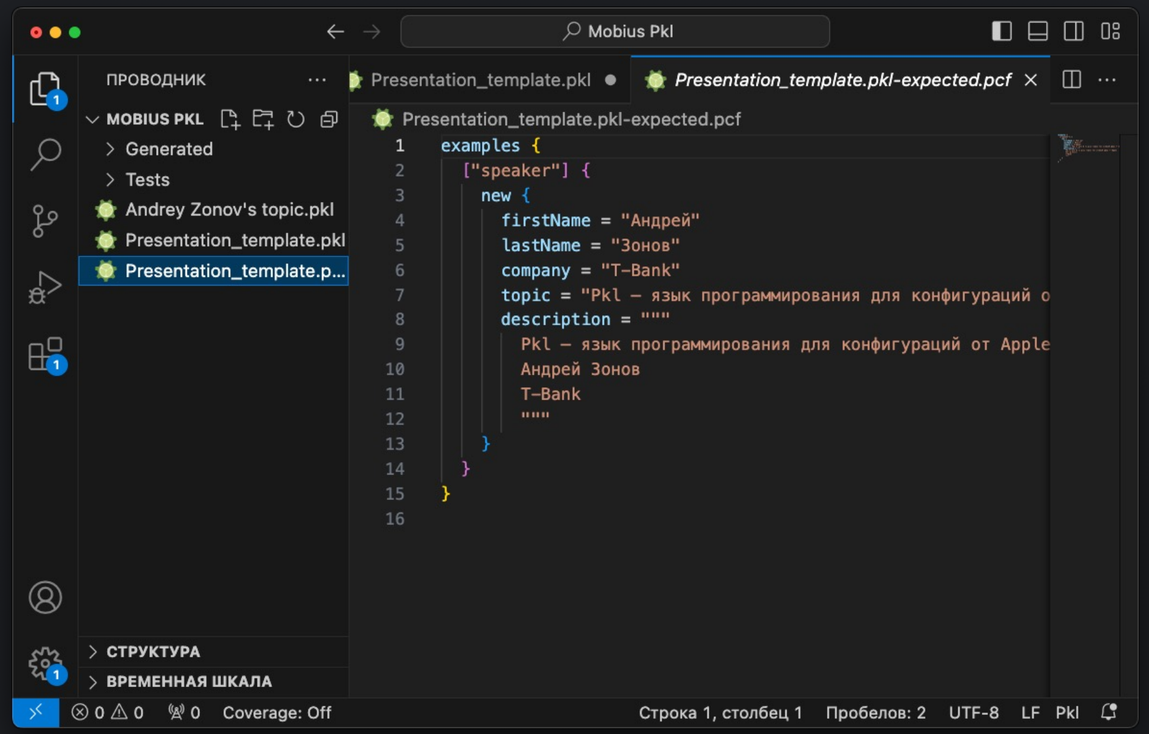
Тестирование



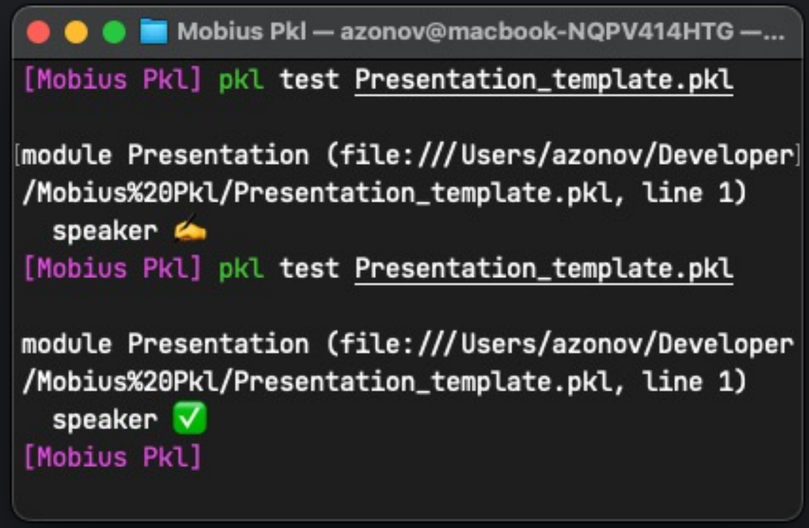
- Тесты поддерживаются из коробки
- Fact – обычный логический ассерт, подходит для юнит тестов
- Example – аналог снапшот тестирования для конфига



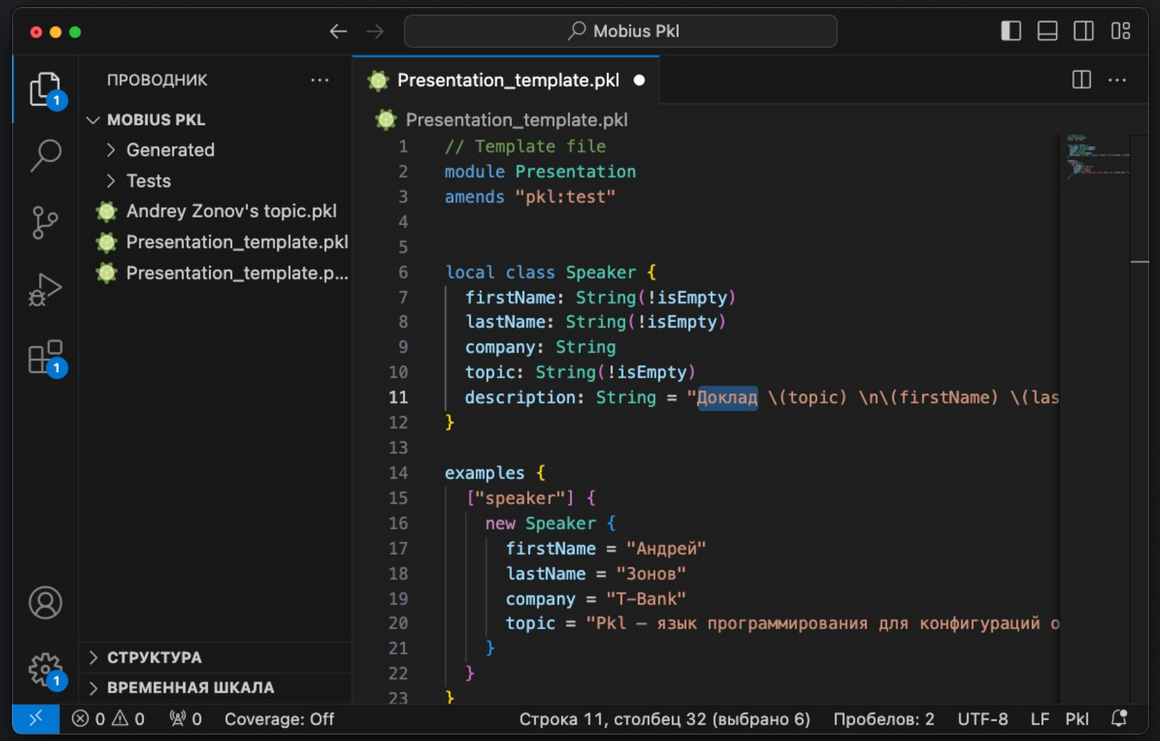
Тестирование



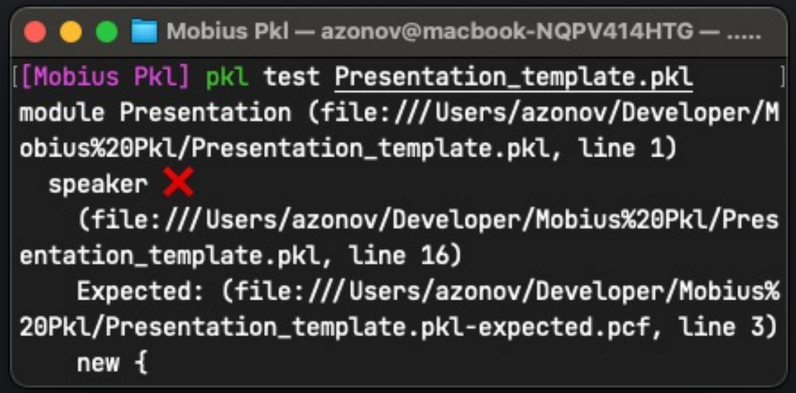
- Тесты поддерживаются из коробки
- Fact – обычный логический ассерт, подходит для юнит тестов
- Example – аналог снапшот тестирования для конфига



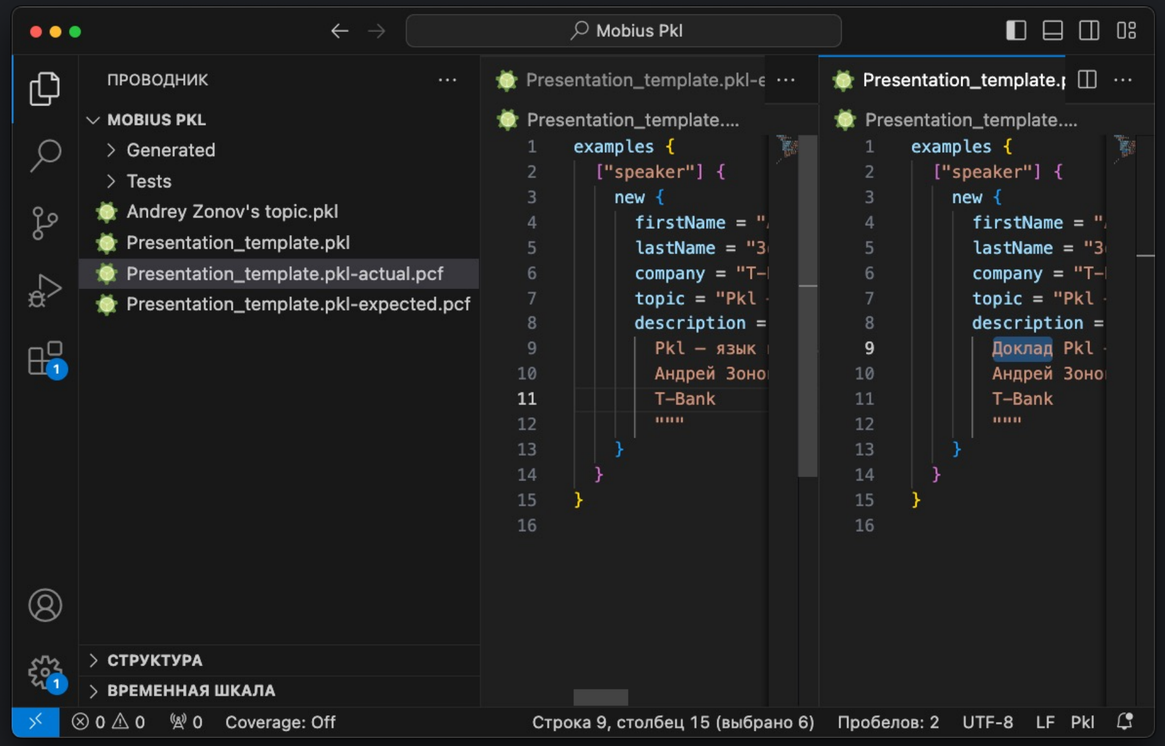
Тестирование



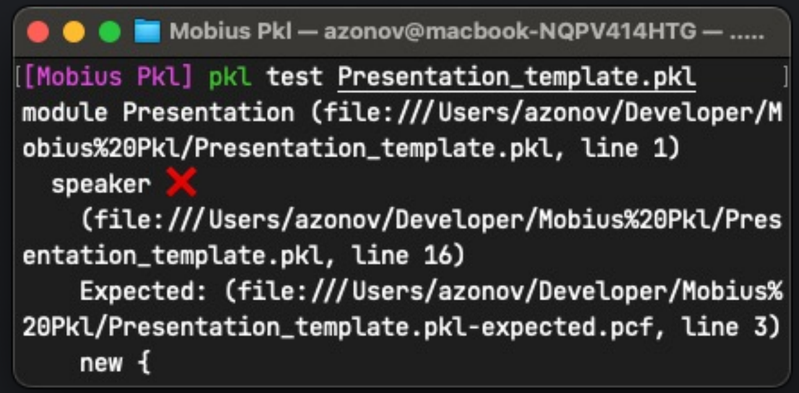
- Тесты поддерживаются из коробки
- Fact – обычный логический ассерт, подходит для юнит тестов
- Example – аналог снапшот тестирования для конфига



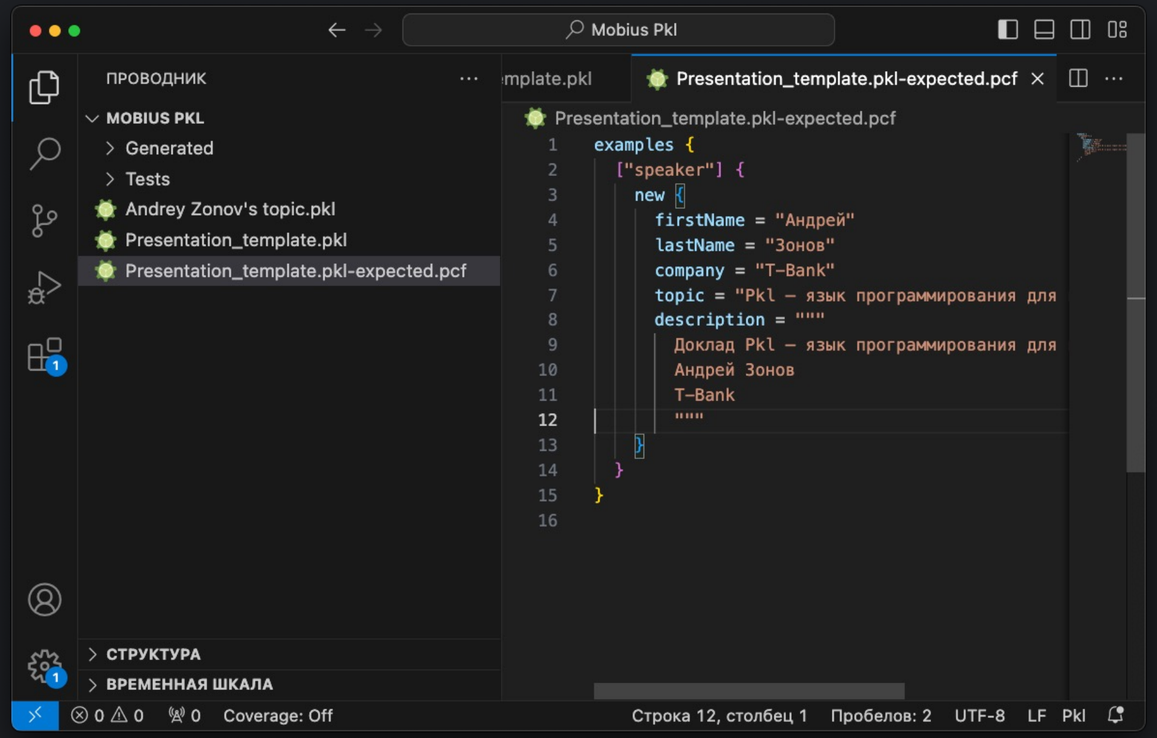
Тестирование



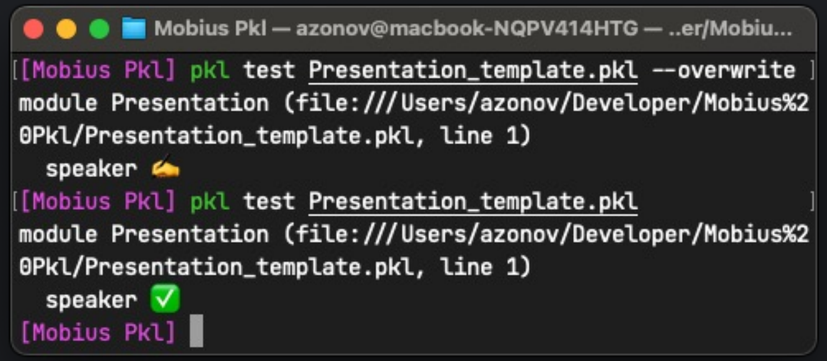
- Тесты поддерживаются из коробки
- Fact – обычный логический ассерт, подходит для юнит тестов
- Example – аналог снапшот тестирования для конфига

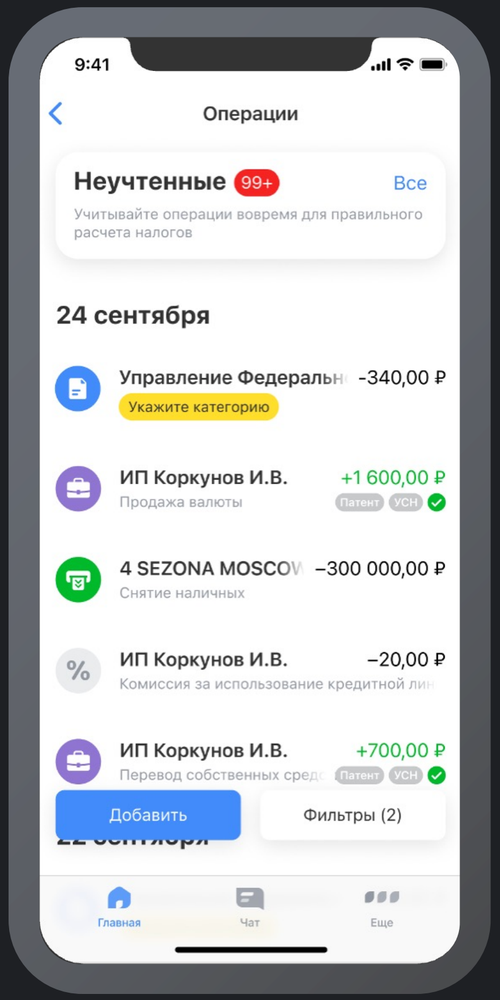


Тестирование



- Тесты поддерживаются из коробки
- Fact – обычный логический ассерт, подходит для юнит тестов
- Example – аналог снапшот тестирования для конфига

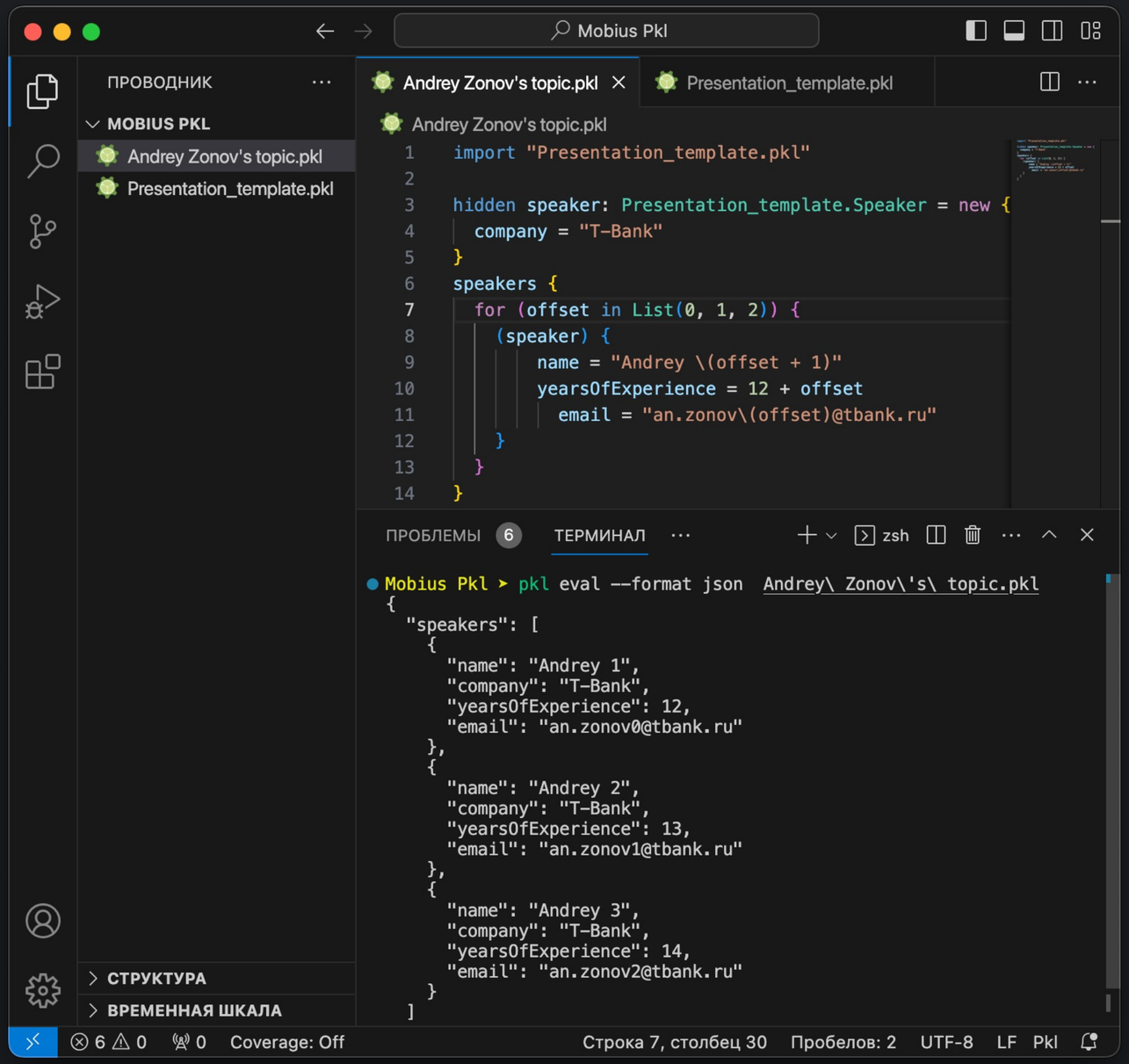




Мокирование

- Пагинация
- Ограничения разных полей
- Уникальность

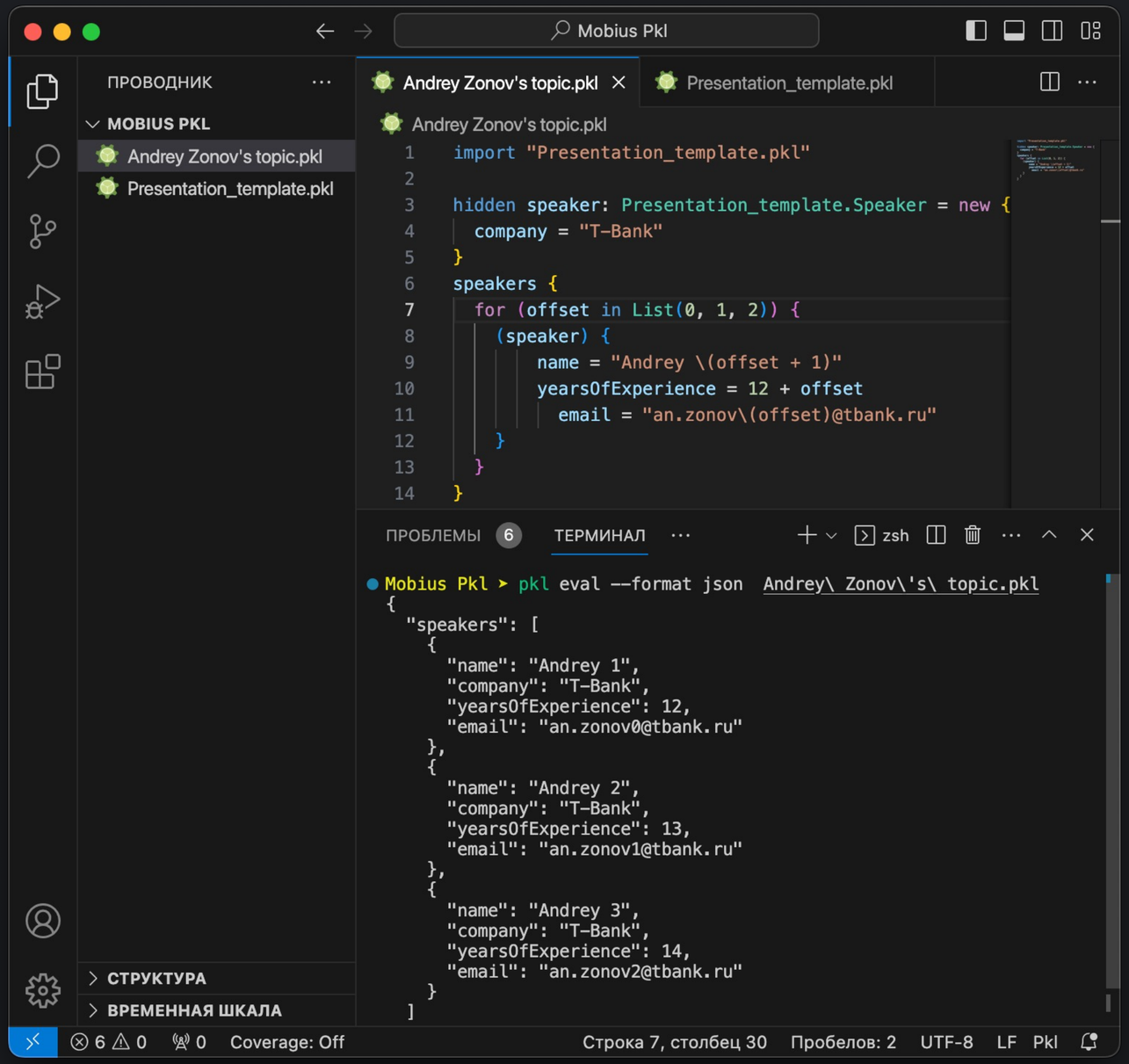
Мокирование



- Задача написать тест на пагинацию
- В зависимости от параметров, количество отличается
- Часть полей должна быть уникальна

1

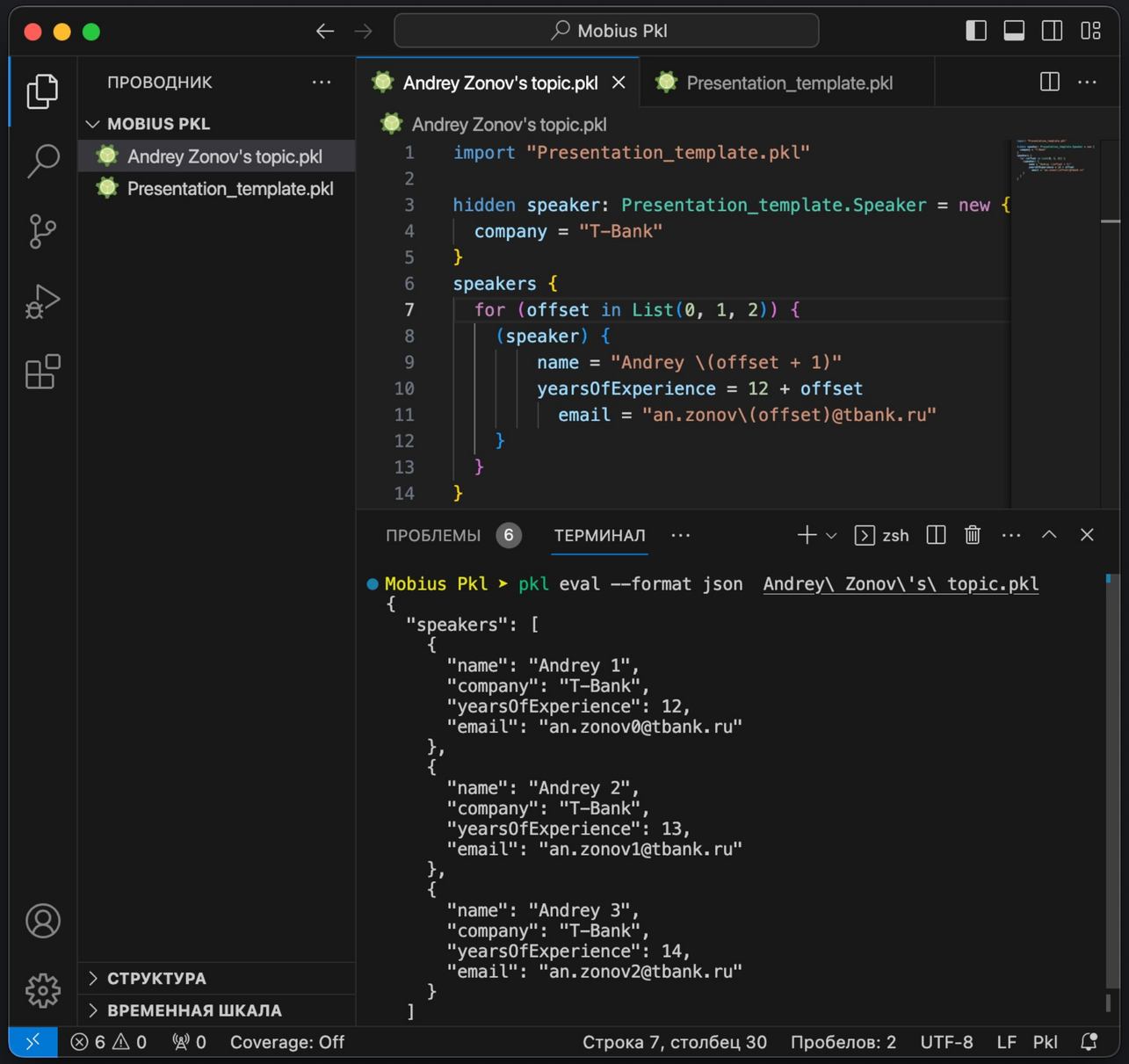
Мокирование



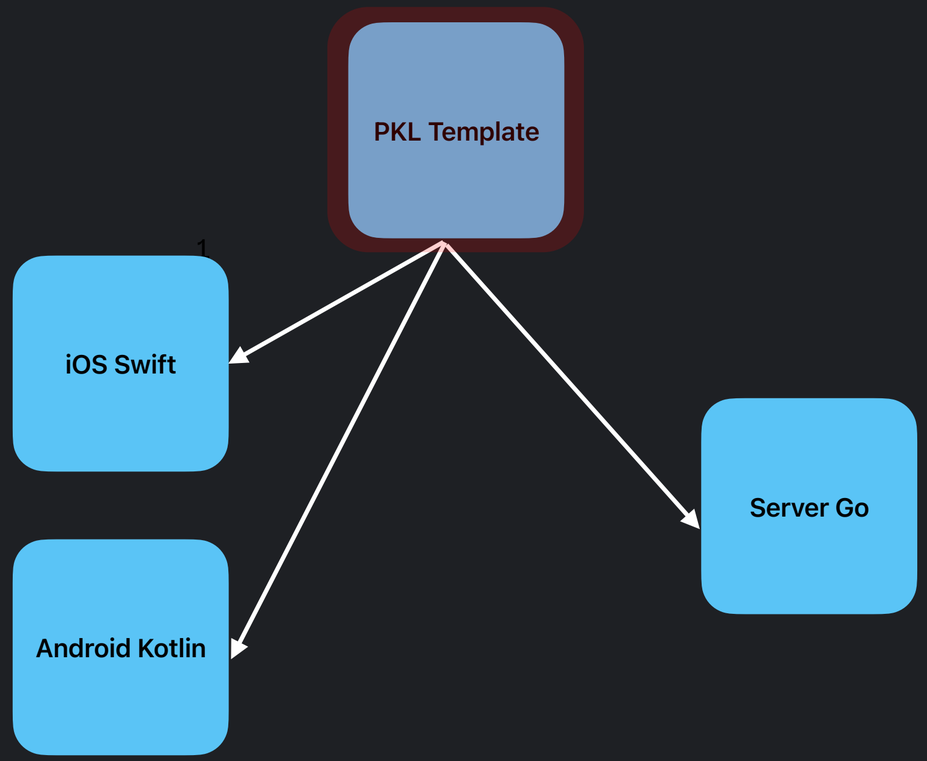
- Возможность динамически генерировать поддерживаемые моки
- Единая логика для клиентов и для сервера

1

Мокирование



- Возможность динамически генерировать поддерживаемые моки
- Единая логика для клиентов и для сервера



```
/// When the above code is evaluated for the first time, the fo
/// ```
/// examples {
///   ["create Pigeons"] {
///     new {
///       name = "Pigeon"
///       age = 21
///     }
///     new {
///       name = "Pigeon"
///       age = 42
///     }
///   }
/// }
/// ```
///
/// To update expected results after making changes,
/// delete _$filename-expected.pcf_ and evaluate the test modul
examples: Mapping<String, Listing<unknown>>?

  /// The results of running a module's benchmarks.
  class BenchmarkReport {
    /// The results of the microbenchmarks that were run.
    ///
    /// Mapping keys are the benchmarks' descriptive names.
    microbenchmarks: Mapping<String, BenchmarkResult>?

    /// The results of the output benchmarks that were run.
    ///
    /// Mapping keys are the benchmarks' descriptive names.
    outputBenchmarks: Mapping<String, BenchmarkResult>?

    /// The results of the parser benchmarks that were run.
    ///
    /// Mapping keys are the benchmarks' descriptive names.
    parserBenchmarks: Mapping<String, BenchmarkResult>?

    /// The platform that benchmarks were run on.
    platform: _platform.Platform
  }
```

Pkl

Standard Library

```
/// A normal version number MUST take the form X.Y.Z
function isNormal(): Boolean = preRelease == null &&

/// Tells if this version has a non-zero [major] and
function isStable(): Boolean = major > 0 && preReleas

/// Strips [preRelease] and [build] from this versio
function toNormal(): Version = (this) { preRelease =

function toString() = "\(major).\(minor).\(patch)\(i

local function isPreReleaseLessThan(other: Version):
  if (preRelease == null) false
  else if (other.preRelease == null) true
  else if (preRelease == other.preRelease) false
  else
```

- stdlib
  - Benchmark.pkl
  - DocPackageInfo.pkl
  - DocsiteInfo.pkl
  - EvaluatorSettings.pkl
  - Project.pkl
  - base.pkl
  - doc-package-info.pkl
  - gradle.lockfile
  - json.pkl
  - jsonnet.pkl
  - math.pkl
  - platform.pkl
  - protobuf.pkl
  - reflect.pkl
  - release.pkl
  - semver.pkl
  - settings.pkl
  - shell.pkl
  - stdlib.gradle.kts
  - test.pkl
  - xml.pkl
  - yaml.pkl

```
/// When the above code is evaluated for the first time, the fo
/// ```
/// examples {
///   ["create Pigeons"] {
///     new {
///       name = "Pigeon"
///       age = 21
///     }
///     new {
///       name = "Pigeon"
///       age = 42
///     }
///   }
/// }
/// ```
///
/// To update expected results after making changes,
/// delete _$filename-expected.pcf_ and evaluate the test modul
examples: Mapping<String, Listing<unknown>>?

  /// The results of running a module's benchmarks.
  class BenchmarkReport {
    /// The results of the microbenchmarks that were run.
    ///
    /// Mapping keys are the benchmarks' descriptive names.
    microbenchmarks: Mapping<String, BenchmarkResult>?

    /// The results of the output benchmarks that were run.
    ///
    /// Mapping keys are the benchmarks' descriptive names.
    outputBenchmarks: Mapping<String, BenchmarkResult>?

    /// The results of the parser benchmarks that were run.
    ///
    /// Mapping keys are the benchmarks' descriptive names.
    parserBenchmarks: Mapping<String, BenchmarkResult>?

    /// The platform that benchmarks were run on.
    platform: _platform.Platform
  }
```

Pkl

Standard Library

```
/// A normal version number MUST take the form X.Y.Z
function isNormal(): Boolean = preRelease == null &&

/// Tells if this version has a non-zero [major] and
function isStable(): Boolean = major > 0 && preReleas

/// Strips [preRelease] and [build] from this versio
function toNormal(): Version = (this) { preRelease =

function toString() = "\(major).\(minor).\(patch)\(i

local function isPreReleaseLessThan(other: Version):
  if (preRelease == null) false
  else if (other.preRelease == null) true
  else if (preRelease == other.preRelease) false
  else
```

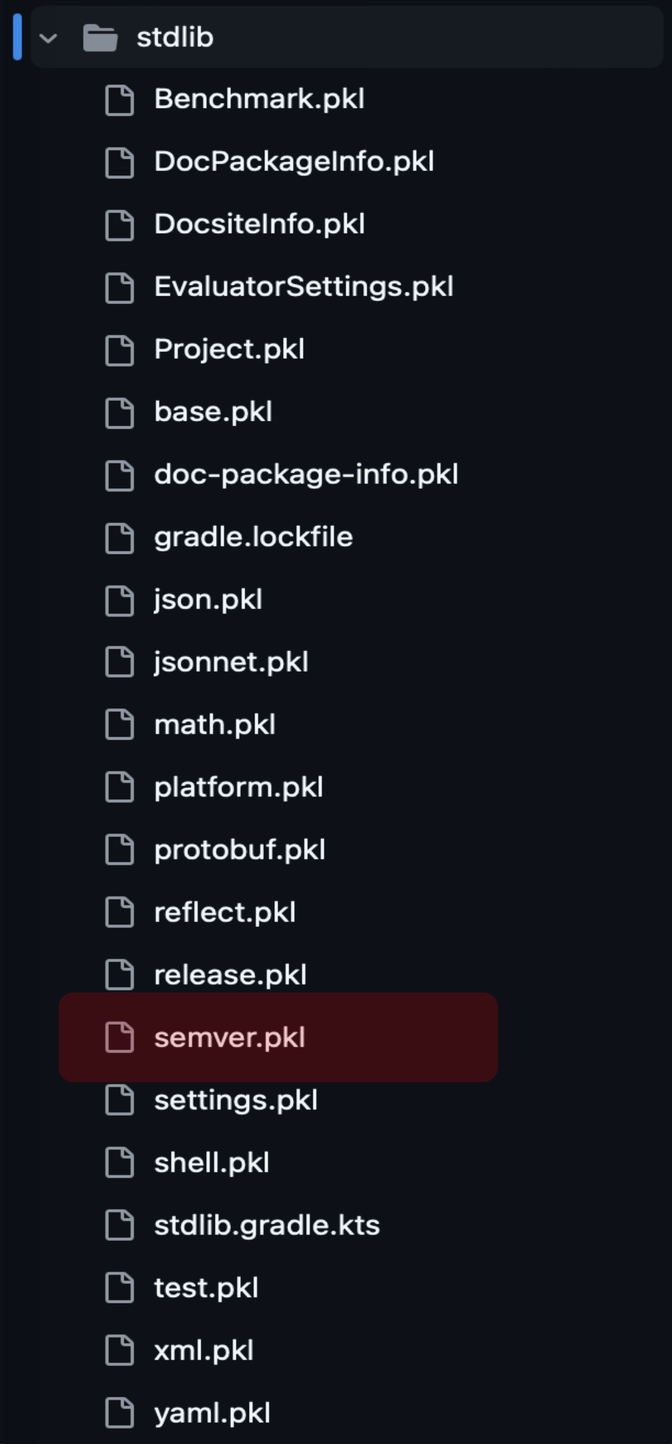
- stdlib
  - Benchmark.pkl
  - DocPackageInfo.pkl
  - DocsiteInfo.pkl
  - EvaluatorSettings.pkl
  - Project.pkl
  - base.pkl
  - doc-package-info.pkl
  - gradle.lockfile
  - json.pkl
  - jsonnet.pkl
  - math.pkl
  - platform.pkl
  - protobuf.pkl
  - reflect.pkl
  - release.pkl
  - semver.pkl
  - settings.pkl
  - shell.pkl
  - stdlib.gradle.kts
  - test.pkl
  - xml.pkl
  - yaml.pkl

```
/// When the above code is evaluated for the first time, the fo
/// ```
/// examples {
///   ["create Pigeons"] {
///     new {
///       name = "Pigeon"
///       age = 21
///     }
///     new {
///       name = "Pigeon"
///       age = 42
///     }
///   }
/// }
/// ```
///
/// To update expected results after making changes,
/// delete _$filename-expected.pcf_ and evaluate the test modul
examples: Mapping<String, Listing<unknown>>?

  /// The results of running a module's benchmarks.
  class BenchmarkReport {
    /// The results of the microbenchmarks that were run.
    ///
    /// Mapping keys are the benchmarks' descriptive names.
    microbenchmarks: Mapping<String, BenchmarkResult>?

    /// The results of the output benchmarks that were run.
    ///
    /// Mapping keys are the benchmarks' descriptive names.
    outputBenchmarks: Mapping<String, BenchmarkResult>?

    /// The results of the parser benchmarks that were run.
    ///
    /// Mapping keys are the benchmarks' descriptive names.
    parserBenchmarks: Mapping<String, BenchmarkResult>?

    /// The platform that benchmarks were run on.
    platform: _platform.Platform
  }
```

Pkl

Standard Library

```
/// A normal version number MUST take the form X.Y.Z
function isNormal(): Boolean = preRelease == null &&

/// Tells if this version has a non-zero [major] and
function isStable(): Boolean = major > 0 && preReleas

/// Strips [preRelease] and [build] from this versio
function toNormal(): Version = (this) { preRelease =

function toString() = "\(major).\(minor).\(patch)\(i

local function isPreReleaseLessThan(other: Version):
  if (preRelease == null) false
  else if (other.preRelease == null) true
  else if (preRelease == other.preRelease) false
  else
```



Basic Pkl/K8s Example

This example demonstrates how to

- convert YAML manifests to Pkl
- evaluate Pkl manifests to YAML
- catch manifest errors during evaluation
- apply Pkl manifests to Kubernetes.

To convert the YAML manifests located in the *yaml* directory to Pkl, run io.k8s.convert for each

```
$ pkl eval -p input=../../yaml/frontend.yaml \
    -o frontend.pkl \
    package://pkg.pkl-lang.org/pkl-pantry/k8s.contrib@1.0.1#/convert.pkl

$ pkl eval -p input=../../yaml/redis-primary.yaml \
    -o redis-primary.pkl \
    package://pkg.pkl-lang.org/pkl-pantry/k8s.contrib@1.0.1#/convert.pkl

$ pkl eval -p input=../../yaml/redis-secondary.yaml \
    -o redis-secondary.pkl \
```

Pkl

Пакеты

Для любой задачи

4. Evaluate Pkl configuration data in Swift

With the above scaffolding set up, evaluate Pkl configuration data in Swift.

First, define some configuration that uses the Pkl schema.

In our example, we create a file at path `pkl/local/appConfig.pkl`. We imagine that we are defi
local environment, and would likewise place other environments in sibling directorys; e.g. `pkl/i`
`pkl/prod/appConfig.pkl`.

pkl/local/appConfig.pkl

```
amends "../AppConfig.pkl"

host = "localhost"

port = 5939
```

Once defined, evaluate the Pkl module into Swift data.

Sources/MyApplication/Main.swift

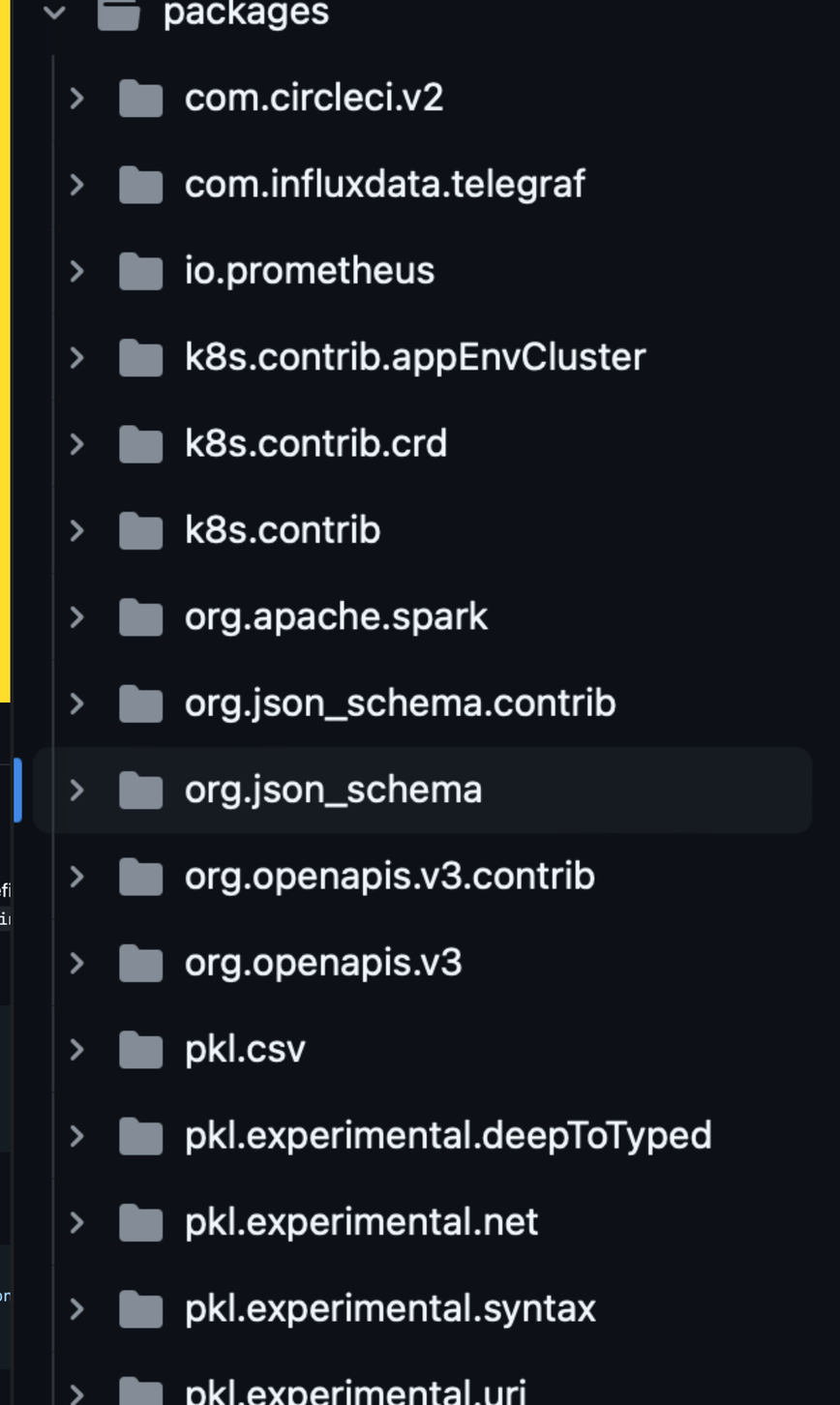
```
func main() async throws {
        let config = try await AppConfig.loadFrom(source: .path("pkl/local/appCor
        print("I'm running on host \(config.host)\n")
}
```

Integration with Go

Pkl provides a rich integration with Go. Our integration allows you to e
generation from Pkl source code.

To get started, reference the Quickstart guide. Alternatively, use our e

```
package main

import (
        "context"
        "fmt"

        "github.com/myteam/myapp/myconfig"
)

func main() {
        cfg, err := myconfig.LoadFromPath(context.Backgrou
        if err != nil {
                panic(err)
```



Basic Pkl/K8s Example

This example demonstrates how to

- convert YAML manifests to Pkl
- evaluate Pkl manifests to YAML
- catch manifest errors during evaluation
- apply Pkl manifests to Kubernetes.

To convert the YAML manifests located in the *yaml* directory to Pkl, run io.k8s.convert for each

```
$ pkl eval -p input=../../yaml/frontend.yaml \
    -o frontend.pkl \
    package://pkg.pkl-lang.org/pkl-pantry/k8s.contrib@1.0.1#/convert.pkl

$ pkl eval -p input=../../yaml/redis-primary.yaml \
    -o redis-primary.pkl \
    package://pkg.pkl-lang.org/pkl-pantry/k8s.contrib@1.0.1#/convert.pkl

$ pkl eval -p input=../../yaml/redis-secondary.yaml \
    -o redis-secondary.pkl \
```

Integration with Go

Pkl provides a rich integration with Go. Our integration allows you to e
generation from Pkl source code.

To get started, reference the Quickstart guide. Alternatively, use our e

```go
package main

import (
        "context"
        "fmt"

        "github.com/myteam/myapp/myconfig"
)

func main() {
        cfg, err := myconfig.LoadFromPath(context.Backgrou
        if err != nil {
                panic(err)
```

Pkl

Пакеты

Для любой задачи

4. Evaluate Pkl configuration data in Swift

With the above scaffolding set up, evaluate Pkl configuration data in Swift.

First, define some configuration that uses the Pkl schema.

In our example, we create a file at path `pkl/local/appConfig.pkl`. We imagine that we are defi
local environment, and would likewise place other environments in sibling directorys; e.g. `pkl/i`
`pkl/prod/appConfig.pkl`.

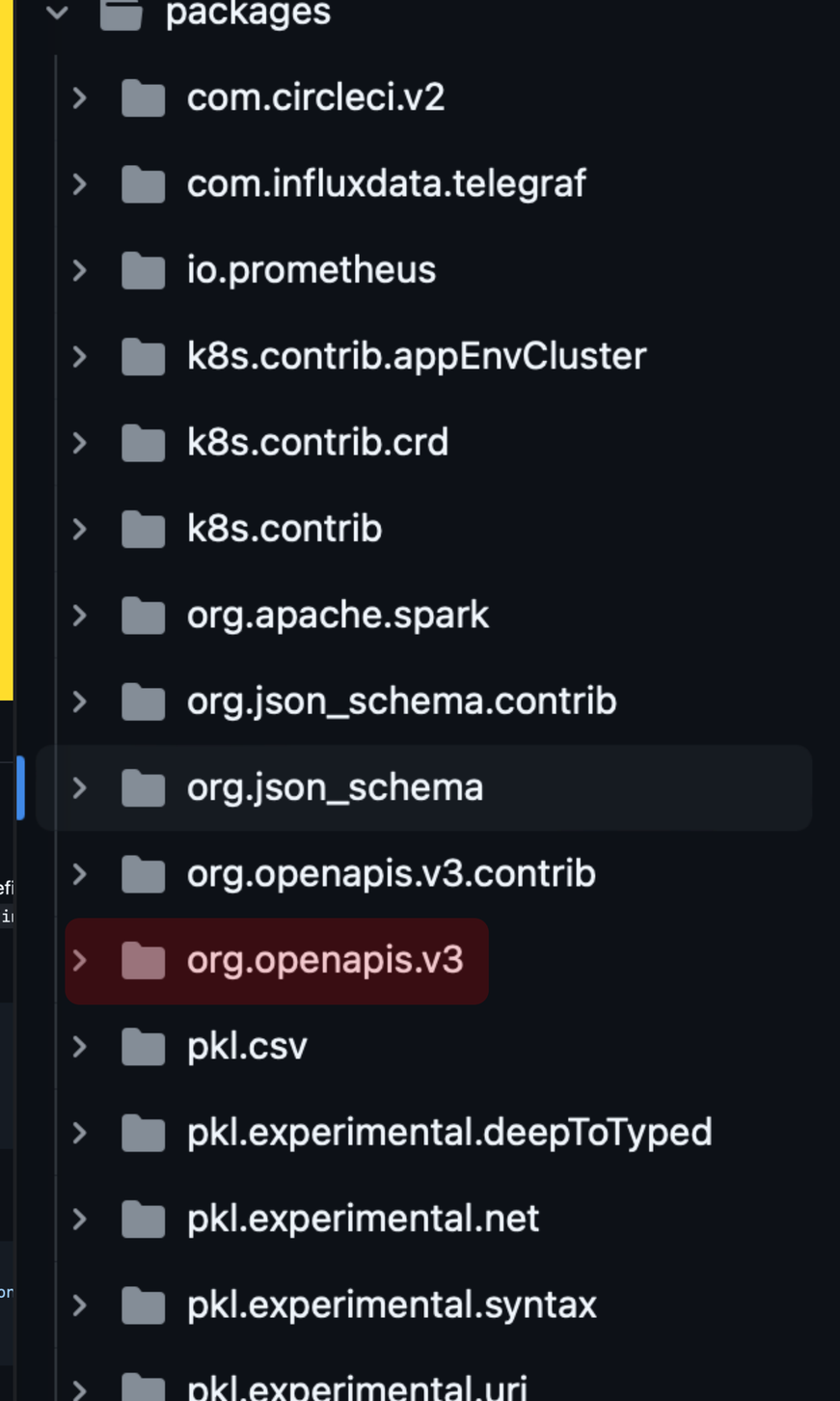
pkl/local/appConfig.pkl

```
amends "../AppConfig.pkl"

host = "localhost"

port = 5939
```

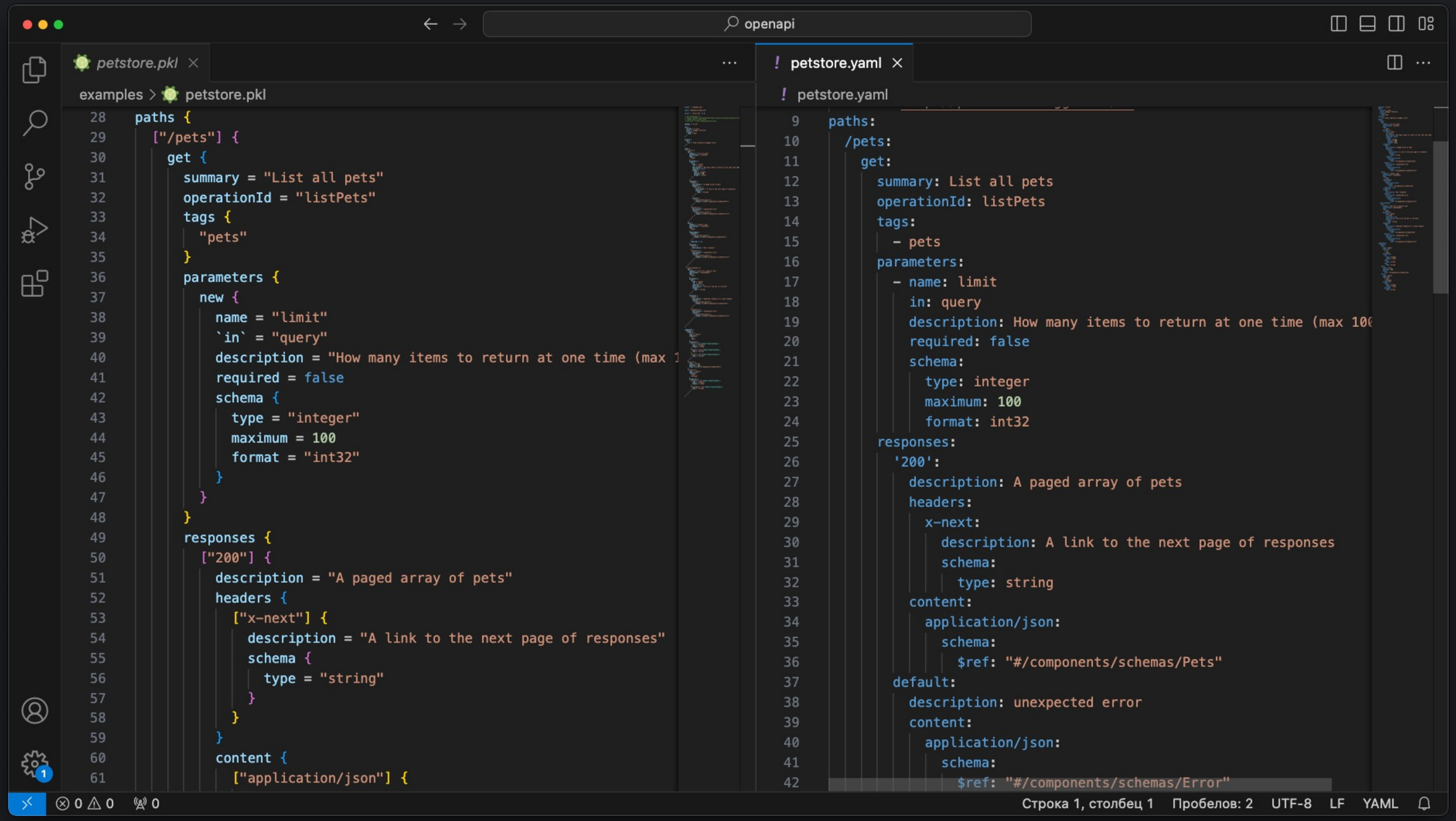
Once defined, evaluate the Pkl module into Swift data.

Sources/MyApplication/Main.swift

```swift
func main() async throws {
        let config = try await AppConfig.loadFrom(source: .path("pkl/local/appCor
        print("I'm running on host \(config.host)\n")
}
```



Package OpenAPI



Выводы

Язык для работы с конфигурациями

JSON/YAML контракты

Не является заменой OpenAPI

Ограничивается моделями

Open source

Позволяет интегрировать без «вендерлока»

Можно начать использовать

Подходит для зрелого проекта

БАНК

Андрей Зонов

Stuff iOS | Руководитель группы

linkedin.com/in/avzonov

t.me/avzonov

@IOS_BROADCAST

Т БАНК

Спасибо!