



Макросы спустя полгода: не всё так гладко



Дмитрий Головин

Старший разработчик
мобильных приложений



**Дмитрий
Головин**

@looneytsci



iOS-разработчик платформенной команды Ozon Seller



Развиваю inhouse СДК



Студент 3 курса СПбГИ

AGENDA

 Немного о HE макросах



 Немного о макросах

 История внедрения макросов в OzonSeller

 Решение проблем при внедрении макросов

 Performance

 Итоги



Как мы жили без макросов?



Sourcery!

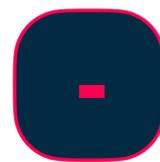


Sourcery



Плюсы

➔ Генерирует ВСЁ (почти)



Минусы

➔ Производительность

➔ Не smoothie

➔ Не натив

➔ Open-source продукт

Генерация с помощью Sourscery

```
configurations:  
- sources:  
  - ${TARGET_DIR}/  
  - ../../../../SellerServices/Sources/  
  - ../../../../SellerUtils/Sources/  
  - ../../../../SellerNetwork/Sources/  
  - ../../../../SellerInterfaces/Sources/  
  - ../../../../SellerCore/Sources/  
  - ../../../../SellerFeatureFlags/Sources/  
  - ../../../../SellerLocalization/Sources/  
  - ../../../../SellerAnalytics/Sources/  
templates:  
  - ${TEMPLATES_DIR}/AutoMockable.stencil  
output: ${TARGET_DIR}/Mocks/Generated/Mocks.generated.swift  
args:  
  autoMockableImports:  
    - SellerUtils  
    - SellerInterfaces  
    - SellerNetwork  
    - Alamofire  
    - SellerInterfaces  
    - SellerCore  
    - AVFoundation  
    - SellerFeatureFlags  
    - SellerLocalization  
    - OzonId  
    - SellerAnalytics  
    - Combine  
  autoMockableTestableImports:  
    - SellerServices
```

config

```
extension IService {}
```

```
class IServiceMock {  
    ...  
}
```

AGENDA

 Немного о HE макросах

 **Немного о макросах** 

 История внедрения макросов в OzonSeller

 Решение проблем при внедрении макросов

 Performance

 Итоги



Swift Macro



Давным-давно...



Тим Кук
Oct 2022

Всем привет, коллеги.
Посмотрите, пожалуйста, mr по
макросам! Выложил на ревью!



Oct 2022

Gentle ping 🙏



Nov 2022

Ping 😓



Dec 2022

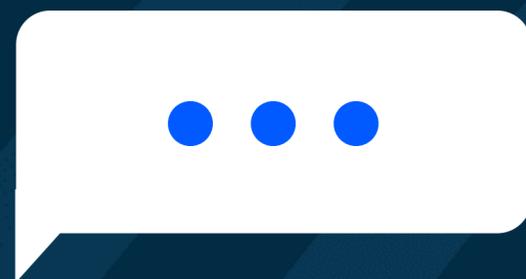
EVIL PING 😡😡😡



One more thing...



Ожидания от макросов



01

Ускорить сборку



02

Генерировать код проще и удобнее



03

Отказаться от сторонних зависимостей



А что это такое?

```
@AutoEquatable
enum Route {
    case main

    var isMain: Bool { self == .main }
}

extension Route: Equatable {
    static func == (lhs: Route, rhs: Route) -> Bool {
        let otherVariablesCheck = lhs.isMain == rhs.isMain
        switch (lhs, rhs) {
        case (.main, .main):
            return otherVariablesCheck
        default:
            return false
        }
    }
}
```



Freestanding macro
(самодостаточный)



Attached macro
(зависимый)



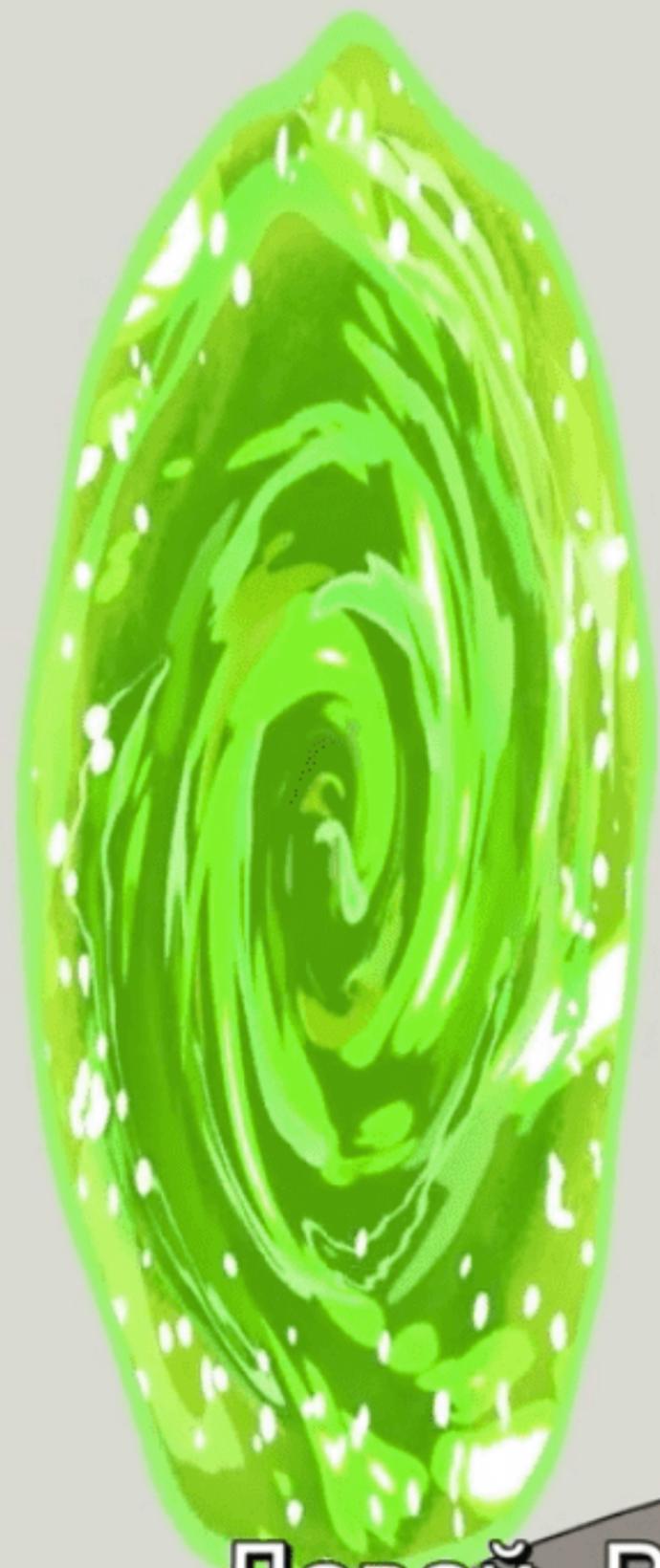
Роли макросов

Freestanding Macro

- Expression
- Declaration

Attached Macro

- Peer
- Accessor
- MemberAttribute
- Member
- Conformance



Давай. Вошли и вышли, приключение на 20 минут.

AGENDA

 Немного о HE макросах

 Немного о макросах

 **История внедрения макросов
в OzonSeller** 

 Решение проблем при внедрении макросов

 Performance

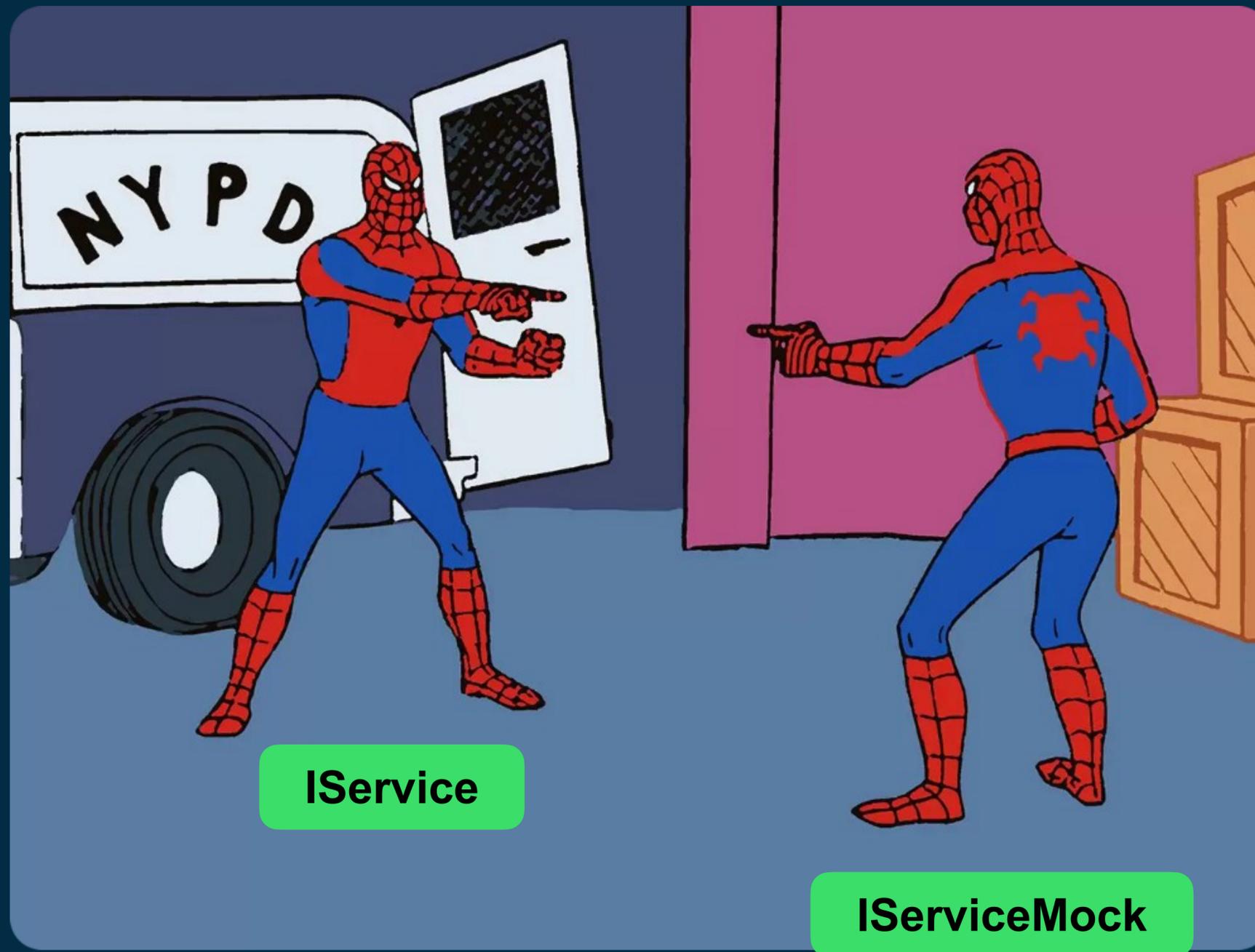
 Итоги



MockMacro



MockMacro



MockMacro

```
@Mock
protocol IService {
    func method()
}
```

```
final class IServiceMock: IService {
    func method() {
        methodCallsCount += 1
        methodClosure?()
    }
    var methodCallsCount = 0
    var methodClosure: (() -> Void)?
}
```

OzonMacro



OzonMacro

01

@Mock

02

@AnyMockable + @MockAccessor

03

```
let executionTime = #performanceMeasure { }
```

04

@AutoEquatable

05

@Arbitrary(.dynamic)

Первые впечатления работы с макросами

Плюсы

-  Макросы простые и удобные
-  Макросы работают быстро
-  Макросы имеют «низкий» порог входа
-  OzonMacro сможет заменить Sourcery

Минусы

-  Макросы — ограниченный инструмент
-  Макросы имеют баги
-  Макросы не умеют «коммуницировать» между собой
-  Нет удобной документации

Отключение генерации кода макросами

```
@Mock
protocol IService: AnyActor {}
actor IServiceMock: IService { @Mock
}
```

Видим лишь то, что дают...

```
@Mock
protocol IService: IParentProtocol {
    var closure: () -> Void { get set }
}
```

Видим лишь то, что дают...

```
(lldb) po decl
ProtocolDeclSyntax
├─attributes: AttributeListSyntax
│   └─[0]: AttributeSyntax
│       ├─atSign: atSign
│       └─attributeName: IdentifierTypeSyntax
│           └─name: identifier("Mock")
├─modifiers: DeclModifierListSyntax
├─protocolKeyword: keyword(SwiftSyntax.Keyword.protocol)
├─name: identifier("IService")
├─inheritanceClause: InheritanceClauseSyntax
│   ├─colon: colon
│   └─inheritedTypes: InheritedTypeListSyntax
│       └─[0]: InheritedTypeSyntax
│           └─type: IdentifierTypeSyntax
│               └─name: identifier("IParentService")
```

Видим лишь то, что дают...

```
@Mock
```

```
protocol IService: IParentProtocol {  
    var closure: () -> Void { get set }  
}
```

```
final class IServiceMock: IService {
```



```
Type 'IServiceMock' does not conform to protocol 'IParentProtocol'
```

Pain moments

```
@Mock
protocol IService {
    var worker: Worker { get set }
}

final class IServiceMock: IService {

    // MARK: - worker

    var worker: Worker {
        get {
            return underlyingWorker
        }
        set(value) {
            underlyingWorker = value
        }
    }

    var underlyingWorker: Worker !
}
```

Class 'IServiceMock' has no initializers

2 Consecutive declarations on a line must be separated by ';'

Pain moments

```
@Mock
protocol IService {
    var worker: Worker { get set }
}
final class IServiceMock: IService {
    // MARK: - worker

    var worker: Worker {
        get {
            return underlyingWorker
        }
        set(value) {
            underlyingWorker = value
        }
    }
    var underlyingWorker: Worker ?
}
```

Class 'IServiceMock' has no initializers

2 Consecutive declarations on a line must be separated by ';'

Pain moments

```
let forceUnwrappedType = ImplicitlyUnwrappedOptionalTypeSyntax(  
    wrappedType: variableType  
)
```

Pain moments

CMD + SHIFT + K, ты должен был бороться со злом.. а не примкнуть к нему...

- ✓  OzonMacroTests 2 issues
 - ✗ Undefined symbols:
 - ✗ Linker command failed with exit code 1 (use -v to see invocation)

Pain moments

CMD + SHIFT + K, ты должен был бороться со злом.. а не примкнуть к нему...

```
let testMacros: [String: Macro.Type] = [  
  "Mock": MockMacro.self,  
  "performanceMeasure": PerformanceMeasureMacro.self,  
  "AutoEquatable": AutoEquatableMacro.self,  
]
```

Отсутствие коммуникации

```
@Mock
protocol IService {
    func makeWork()
}

#Preview {
    ContentView(service: IServiceMock())
}
```

✘ Cannot find 'IServiceMock' in scope

Нельзя генерировать Preview

```
@Previewable
struct SomeView: View {
    var text: String

    var body: some View { Text(text) }
}

#Preview {                                     @Previewable
    SomeView(text: "123")
}
```

**И что с этим
делать?**



AGENDA

 Немного о HE макросах

 Немного о макросах

 История внедрения макросов в OzonSeller

 **Решение проблем
при внедрении макросов** 

 Performance

 Итоги



Отключение генерации кода макросами

```
@Mock
protocol IService { }
#if DEBUG
final class IServiceMock: IService {
}
#endif
```

@Mock

Макрос макросу рознь

```
enum Generated {  
    static let service = IServiceMock()  
}  
  
#Preview {  
    ContentView(service: Generated.service)  
}
```

Проблема отцов и детей

```
protocol ParentProtocol {  
    func makeParentWork()  
}  
  
@Mock  
protocol ChildProtocol: ParentProtocol {  
    func makeChildWork()  
}
```

Проблема отцов и детей

Parent



Child



ParentMock



ChildMock

Проблема отцов и детей

```
@Mock(.open, shouldBeInherited: true)
protocol ParentProtocol {
    func makeParentWork()
}

@Mock
protocol ChildProtocol: ParentProtocol {
    func makeChildWork()
}
```

Проблема отцов и детей

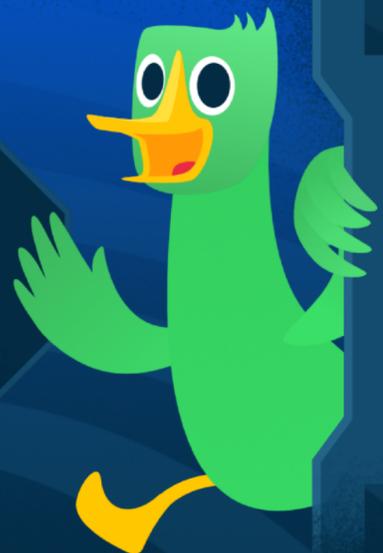


Проблема отцов и детей

ParentOne ParentTwo



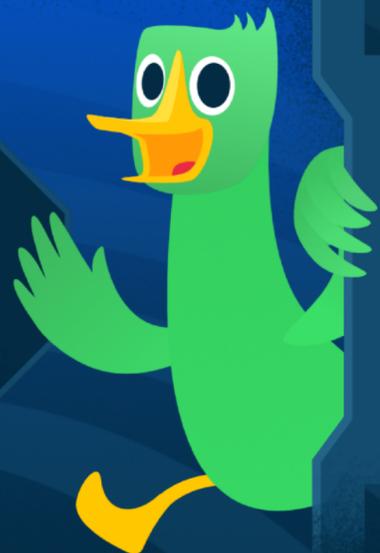
Проблема отцов и детей



**Возможные
решения**

- Искать нужный родительский протокол в проекте
- Украсть бэкап сик soucery
- Спарсить все протоколы проекта в один файл

Проблема отцов и детей



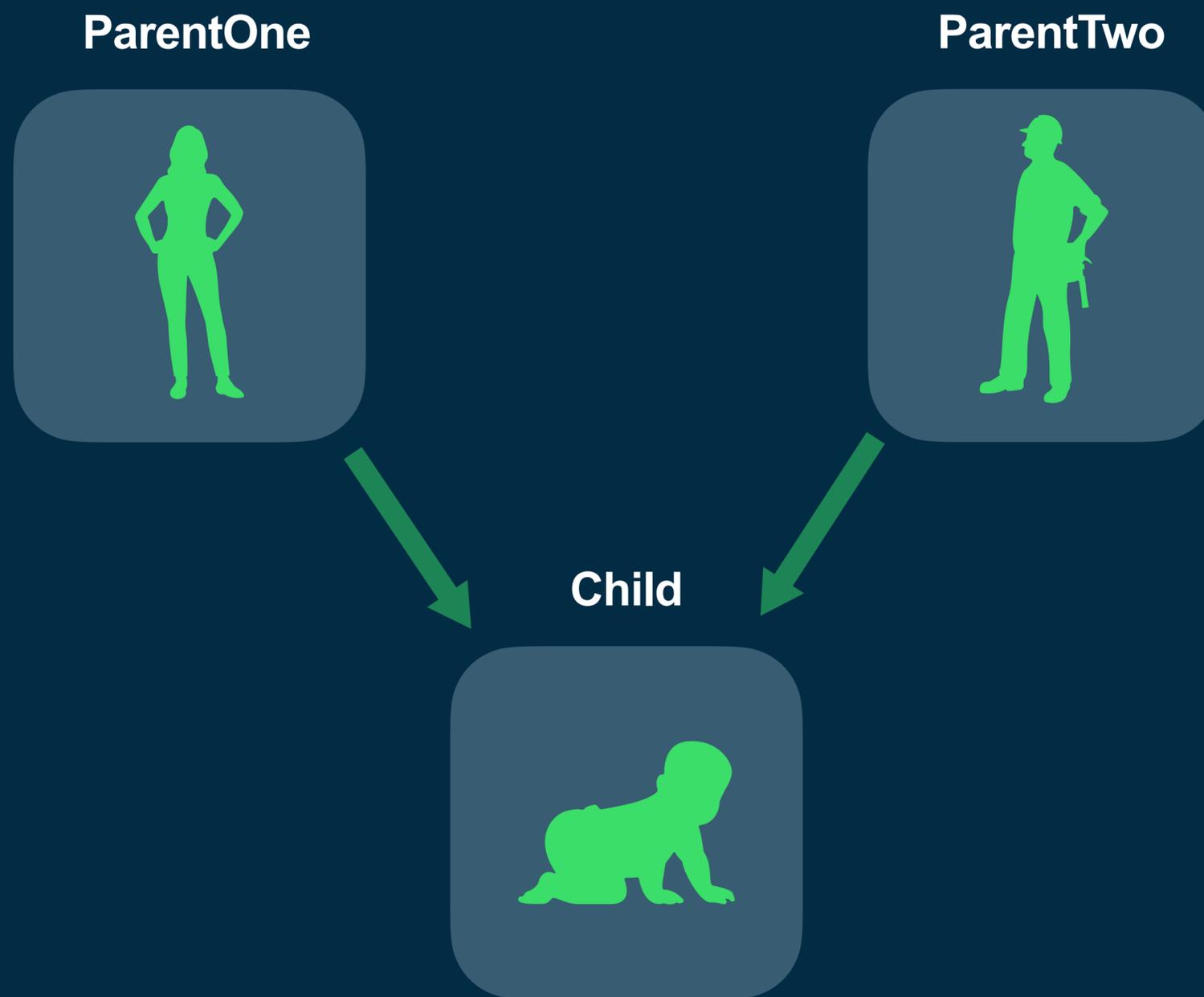
Возможные решения

- **Заставить разработчиков писать моки руками**
- **Построить граф наследования протоколов**
- **Полуавтоматически генерировать моки**

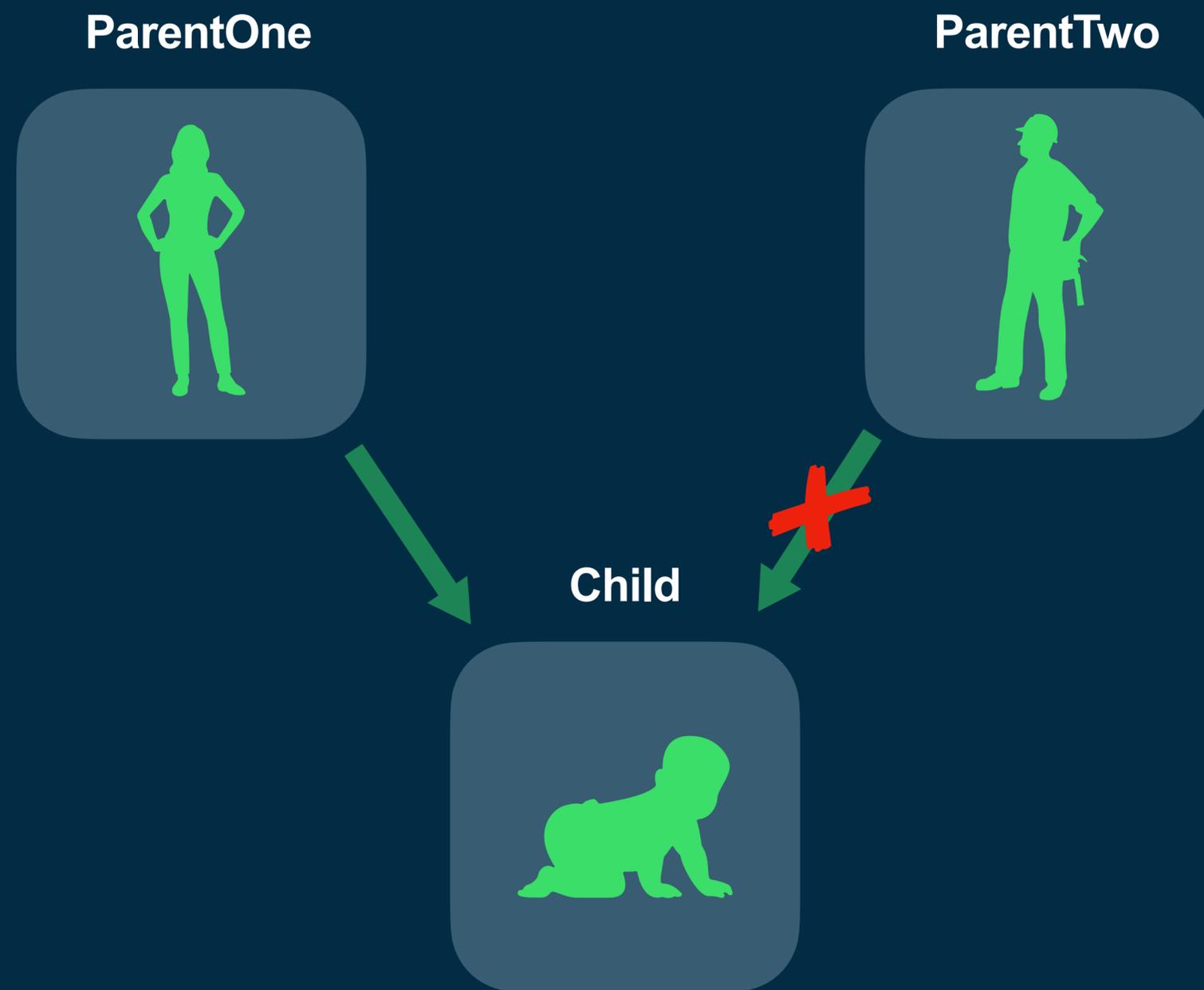
Дружеская беседа с командой



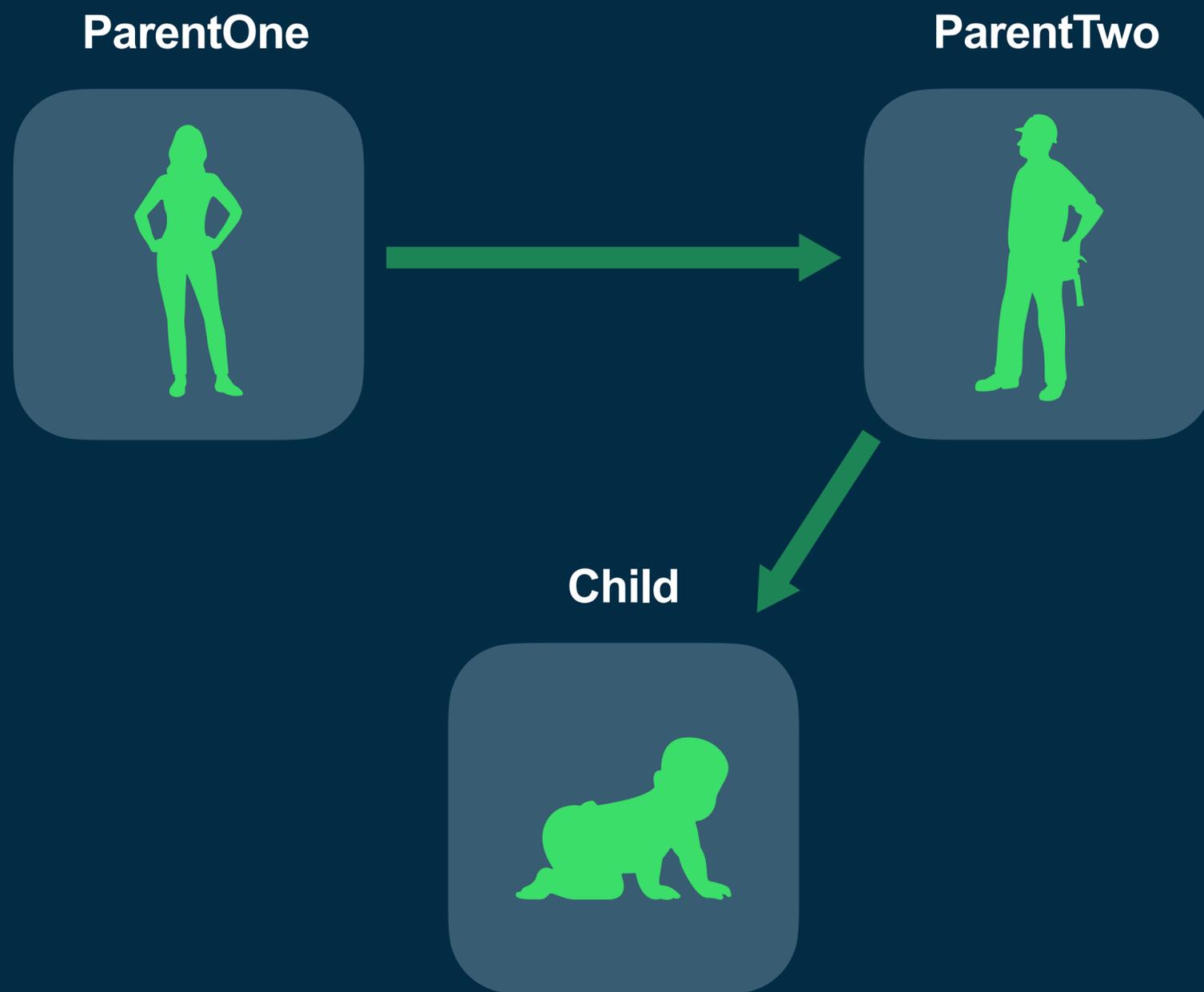
Граф наследования протоколов



Граф наследования протоколов



Граф наследования протоколов



Граф наследования протоколов



```
protocol ParentOne {  
    func eat()  
}  
  
class ParentOneMock: ParentOne {  
    func eat() {}  
}  
  
protocol ParentTwo {  
    func cook()  
}  
  
class ParentTwoMock: ParentOneMock, ParentTwo {  
    func cook() { }  
}  
  
protocol Child: ParentOne, ParentTwo {  
    func walk()  
}  
  
class ChildMock: ParentTwoMock, Child {  
    func walk() { }  
}
```

Полуавтоматическое создание комплексного мока

```
protocol ParentOne {  
    func eat()  
}  
  
protocol ParentTwo {  
    func cook()  
}  
  
protocol Child: ParentOne, ParentTwo {  
    func walk()  
}
```

Где-то в проекте...

```
final class ChildMock: Child {  
    private let mock = Mock()  
  
    func eat() { mock.eat() }  
  
    func walk() { mock.walk() }  
  
    func cook() { mock.cook() }  
  
    private class Mock {  
        var walkCallsCount = 0  
        func walk() {  
            walkCallsCount += 1  
        }  
  
        var cookCallsCount = 0  
        func cook() {  
            cookCallsCount += 1  
        }  
  
        var eatCallsCount = 0  
        func eat() {  
            eatCallsCount += 1  
        }  
    }  
}
```

В тестовом таргете

Что-то надо делать руками?..

```
@AnyMockable
final class ChildMock: Child {
    var property: String

    func methodOne() { mock.methodOne() }
}
```

AnyMockable

```
⌘ let mock = Mock()
2
3 final class Mock {
4     var underlyingProperty: String!
5
6     private let lock = NSLock()
7
8     // MARK: - methodOne
9
10    fileprivate func methodOne() {
11        lock.lock()
12        methodOneCallsCount += 1
13        lock.unlock()
14        methodOneClosure?()
15    }
16    var methodOneCallsCount = 0
17    var methodOneClosure: (() -> Void)?
18 }
```

@AnyMockable

FunctionBodyMacro

```
@AnyMockable
final class ChildMock: Child {
    var property: String

    @MockCall
    func methodOne() { }
}
```

FunctionBodyMacro

```
@AnyMockable  
final class ChildMock: Child {  
    var property: String  
  
    func methodOne() { }  
}
```

MockAccessor

```
@AnyMockable
final class ChildMock: Child {
    @MockAccessor
    var property: String
    {
        get {
            mock.underlyingProperty
        }
        set(newValue) {
            mock.underlyingProperty = newValue
        }
    }
}
```

AGENDA

 Немного о HE макросах

 Немного о макросах

 История внедрения макросов в OzonSeller

 Решение проблем при внедрении макросов

 **Performance**



 Итоги



падскажи макросы скока



Картинки

Видео

Покупки

Новости

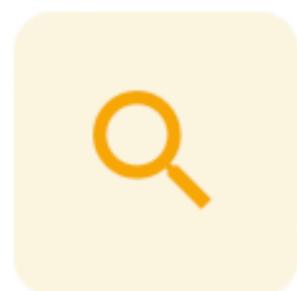
Карты

Книги

Авиабилеты

Результатов: примерно 3 (0,20 сек.)

Возможно, вы имели в виду: **подскажи макросы сколько**

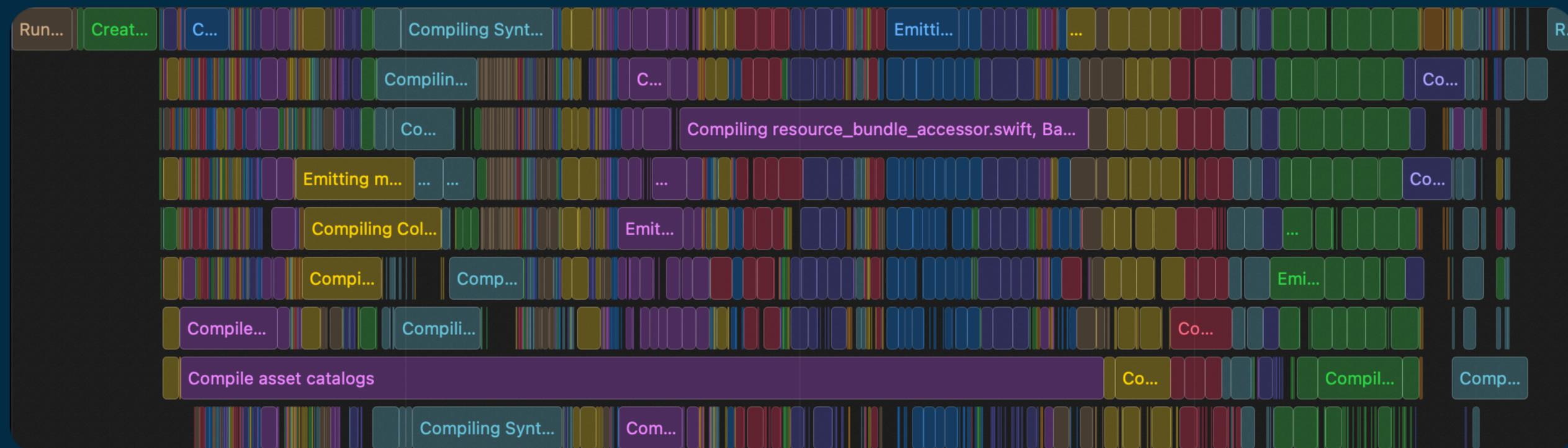


Похоже, по вашему запросу нет полезных результатов.

Используйте слова, которые могут быть на нужной странице. Например, ищите "рецепты пирогов", а не "как испечь пирог".

Нужна помощь? Прочитайте [другие советы](#) по поиску в Google.

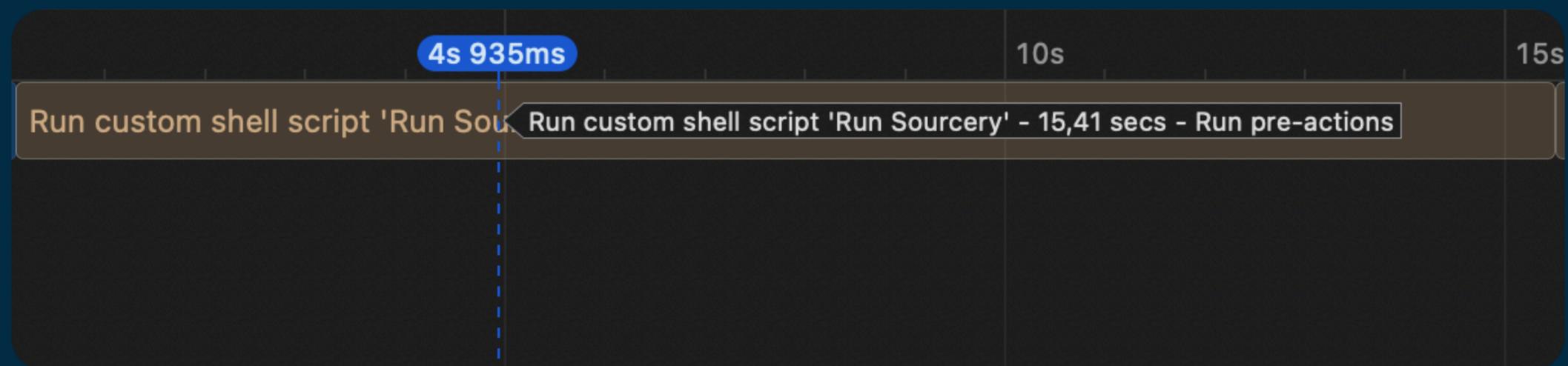
скриншот тетриса





Холодная сборка

Sourcery



Холодная сборка

OzonMacro

~20 с

Swift-syntax

~100 мс

**Генерация мока
для большого
протокола**

~20 мс

**Генерация
AutoEquatable
макроса для
большой
структуры**



Горячая сборка

Sourcery

10 с

**Pre-action-script
sourcery-
константа**

~3-8 с

**Компиляция
.generated файлов**



Горячая сборка

OzonMacro

0 с

Swift-syntax

0 с

Генерация
макросов





OzonMacro

Sourcery

AGENDA

 Немного о HE макросах

 Немного о макросах

 История внедрения макросов в OzonSeller

 Решение проблем при внедрении макросов

 Performance

 **Итоги**



🚀 ⚡ Скорость



🏄‍♂️ Удобство



Как генерировали раньше?



config

```
extension IService {}
```

```
class IServiceMock {  
    ...  
}
```

Как генерируем сейчас?

```
import OzonMacro
```

```
@Mock  
protocol IService {  
    func makeWork()  
}
```

Trade offs



Sandbox



**Ограниченный
скоуп Input'a**



**Отсутствие
связи между
макросами**

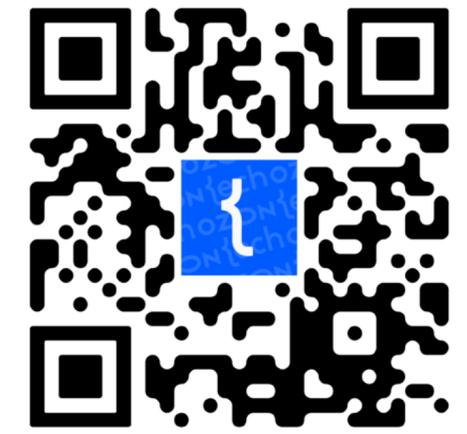


Спасибо
за внимание!

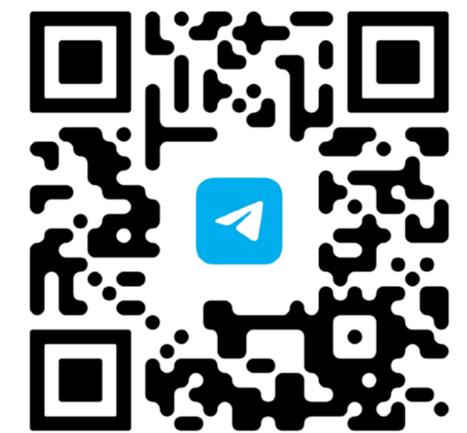


Дмитрий Головин

Старший разработчик
мобильных приложений



Ozon Tech Github



Я в Telegram

