

# Swift Macro 도입하기

Swift Macro로 코드 줄이기

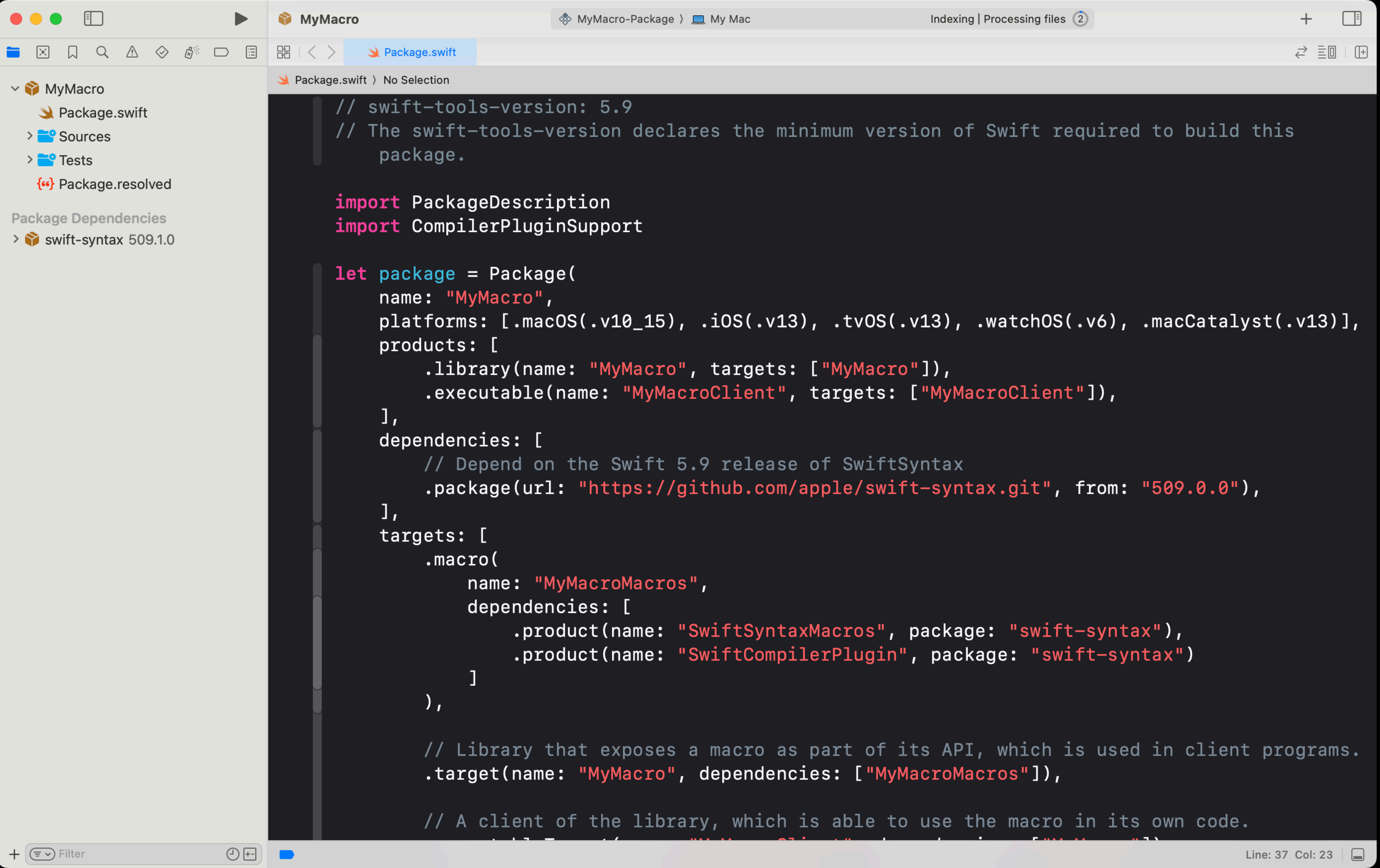
# 목차

- Swift Macro 도입이 쉽지 않은 이유
- Swift Macro 도입 시 얻는 장점과 우려사항
- QnA

# Swift Macro 도입이 쉽지 않은 이유

# Swift Macro 도입이 쉽지 않은 이유

- Swift Syntax 라이브러리의 빌드 시간이 오래 걸림



```

platforms: [.macOS(.v10_15), .iOS(.v13), .tvOS(.v13), .watchOS(.v6), .macCatalyst(.v1)]
products: [
    .library(name: "MyMacro", targets: ["MyMacro"]),
    .executable(name: "MyMacroClient", targets: ["MyMacroClient"]),
],
dependencies: [
    // Depend on the Swift 5.9 release of SwiftSyntax
    .package(url: "https://github.com/apple/swift-syntax.git", from: "509.0.0"),
],
targets: [
    .macro(
        name: "MyMacroMacros",
        dependencies: [
            .product(name: "SwiftSyntaxMacros", package: "swift-syntax"),
            .product(name: "SwiftCompilerPlugin", package: "swift-syntax")
        ]
    ),
    // Library that exposes a macro as part of its API, which is used in client products
    .target(name: "MyMacro", dependencies: ["MyMacroMacros"]),
    // A client of the library, which is able to use the macro in its own code.
    .target(name: "MyMacroClient", dependencies: ["MyMacro"])
]

```



```

platforms: [.macOS(.v10_15), .iOS(.v13), .tvOS(.v13), .watchOS(.v6), .macCatalyst(.v1)]
products: [
    .library(name: "MyMacro", targets: ["MyMacro"]),
    .executable(name: "MyMacroClient", targets: ["MyMacroClient"]),
],
dependencies: [
    // Depend on the Swift 5.9 release of SwiftSyntax
    .package(url: "https://github.com/apple/swift-syntax.git", from: "509.0.0"),
],
targets: [
    .macro(
        name: "MyMacroMacros",
        dependencies: [
            .product(name: "SwiftSyntaxMacros", package: "swift-syntax"),
            .product(name: "SwiftCompilerPlugin", package: "swift-syntax")
        ]
    ),
    // Library that exposes a macro as part of its API, which is used in client products
    .target(name: "MyMacro", dependencies: ["MyMacroMacros"]),
    // A client of the library, which is able to use the macro in its own code.
    .target(name: "MyMacroClient", dependencies: ["MyMacro"])
]

```

**MyMacro**



**MyMacroMacro**



**Swift Syntax**



MyMacro

0.x초



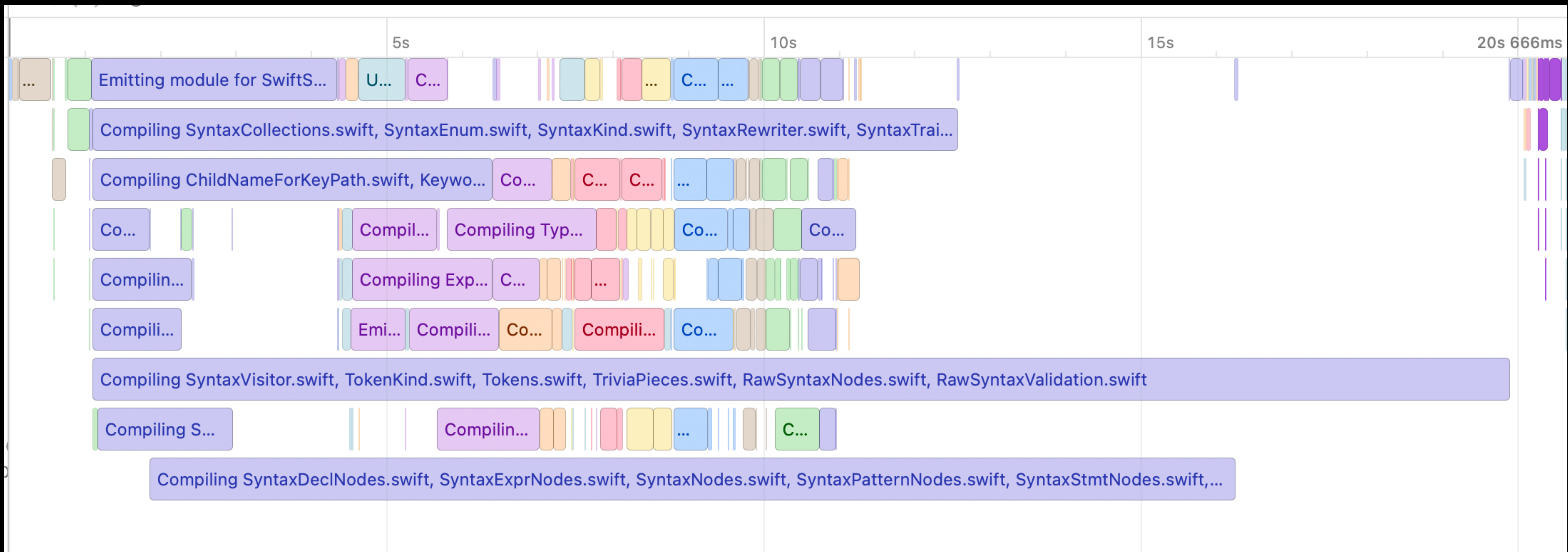
MyMacroMacro

0.x초

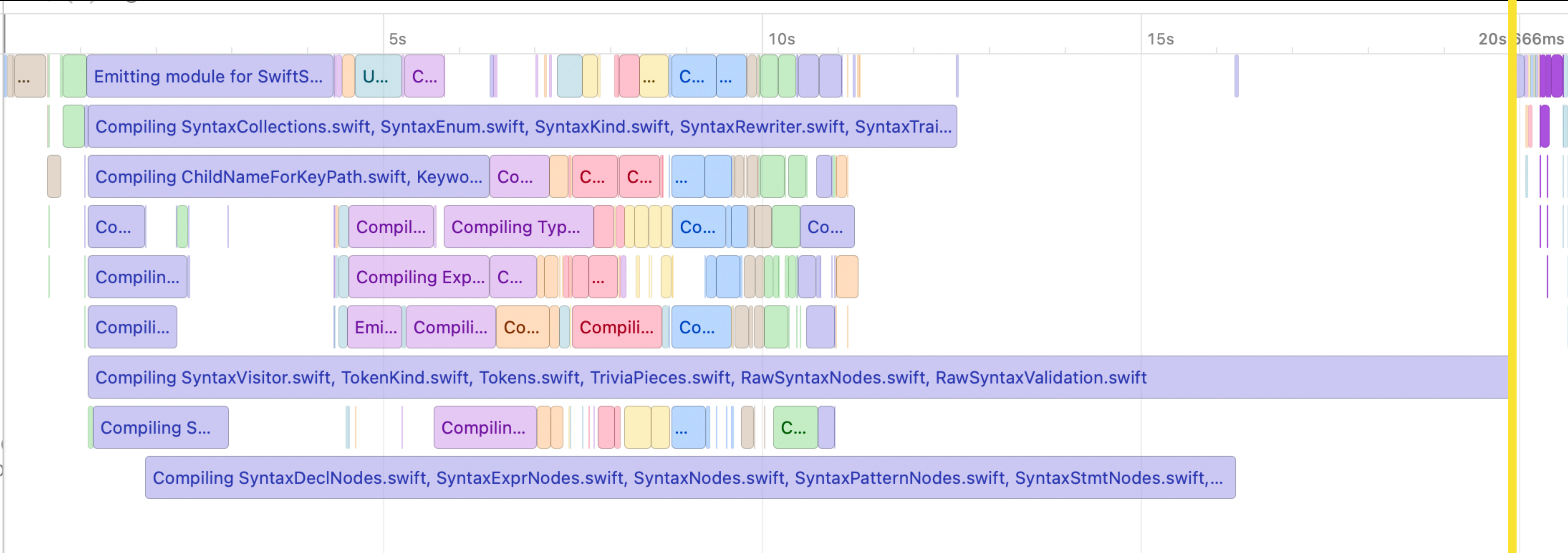


Swift Syntax

20초



# Swift Syntax Build Timeline



# Swift Macro 도입이 쉽지 않은 이유

- Swift Syntax 라이브러리의 빌드 시간이 오래 걸림
  - 빌드 시간 늘어남으로 인해 전체적인 개발 시간 증가 발생(로컬 환경, CI 등)
  - UI의 Preview가 실패할 수 있음

만약 **Swift Syntax**를

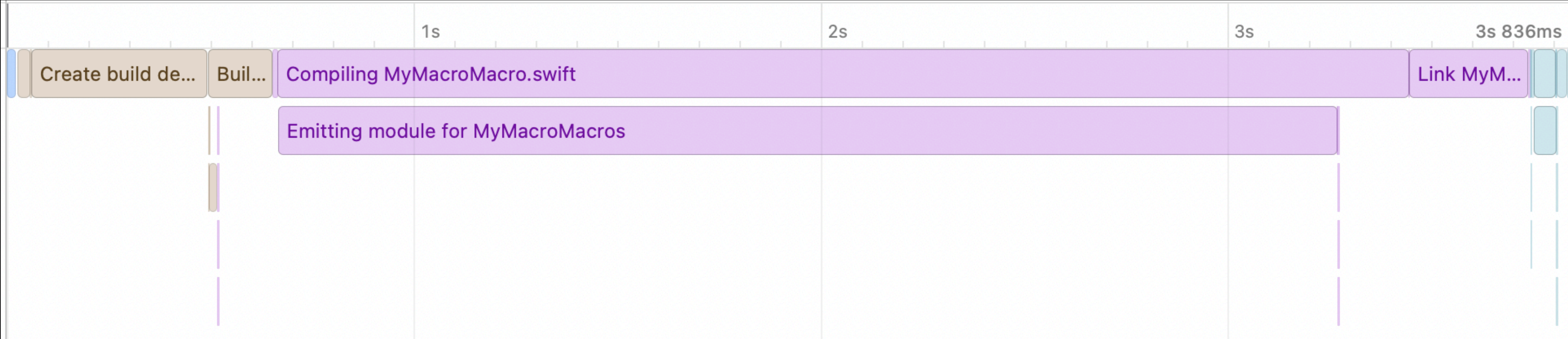
미리 빌드하여 사용할 수 있다면?



✓  XCFramework

 SwiftSyntaxWrapper.xcframework

빌드 시간 : 3.836s



# Swift Macro 도입 시 얻는 장점

- 채널 API의 Response 코드를 다음과 같이 줄일 수 있음

# Swift Macro 도입 시 얻는 장점

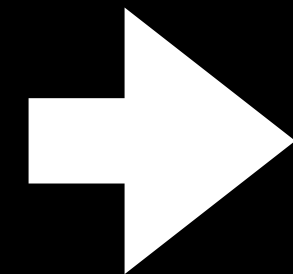
- 채널 API의 Response 코드를 다음과 같이 줄일 수 있음

```
@ResponseInit
public struct SomeResponse {
    @ResponseJSON
    public var title: String
    @ResponseJSON
    public var msg: String
    @ResponseJSON(key: "year_mm")
    public var yearMonth: String
}
```

# Swift Macro 도입 시 얻는 장점

- Public init 코드를 다음과 같이 줄일 수 있음

```
public struct S {  
    let v: String  
    public init(v: String) {  
        self.v = v  
    }  
}
```



```
@MemberwiseInit(.public)  
public struct S {  
    let v: String  
}
```



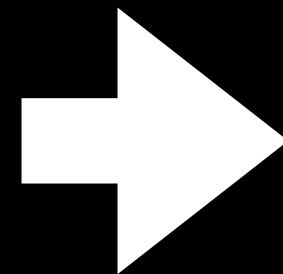
# Swift Macro 도입 시 얻는 장점

- enum case 코드를 다음과 같이 줄일 수 있음

```
enum Animal {  
    case dog  
    case cat(curious: Bool)
```

```
    var isDog: Bool {  
        if case .dog = self {  
            return true  
        }  
        return false  
    }
```

```
    var isCat: Bool {  
        if case .cat = self {  
            return true  
        }  
        return false  
    }  
}
```



```
@CaseDetection  
enum Animal {  
    case dog  
    case cat(curious: Bool)  
}
```

# Swift Macro 도입 시 우려되는 점

- Macro가 무분별하게 추가되지 않을까?
- Macro를 추가한다면, 어떤 기준에서 추가해야할까?

# Swift Macro 도입 시 우려되는 점

- Macro가 무분별하게 추가되지 않을까?
  - Macro가 쉽게 추가되지는 못할 것임. 이는 GitHub에 만들어진 Swift Macro 관련 코드가 별로 늘어나지 않고 있기 때문임.
  - 활용처를 찾기가 쉽지 않아, Macro의 무분별한 추가는 되지 않을 것으로 생각됨
- Macro를 추가한다면, 어떤 기준에서 추가해야할까?
  - 자주 사용되는 상용구 코드의 간소화 하는 방향으로의 Macro 활용

QnA