

Welcome at all about apps

A

Key Facts

250+

Customers

400+

Apps

30+

Employees

8

iOS Devs





code.allaboutapps.at

code.allaboutapps.at

allaboutapps.at/karriere

App Architecture @ all about apps

- 30+ Apps / Year (+ updates, bugfixes, ...)
- 8 iOS Devs
- Changing responsibilities

App Architecture @ all about apps

- Project Structure
- Coordinator
- DataSource

Xcode Project Setup

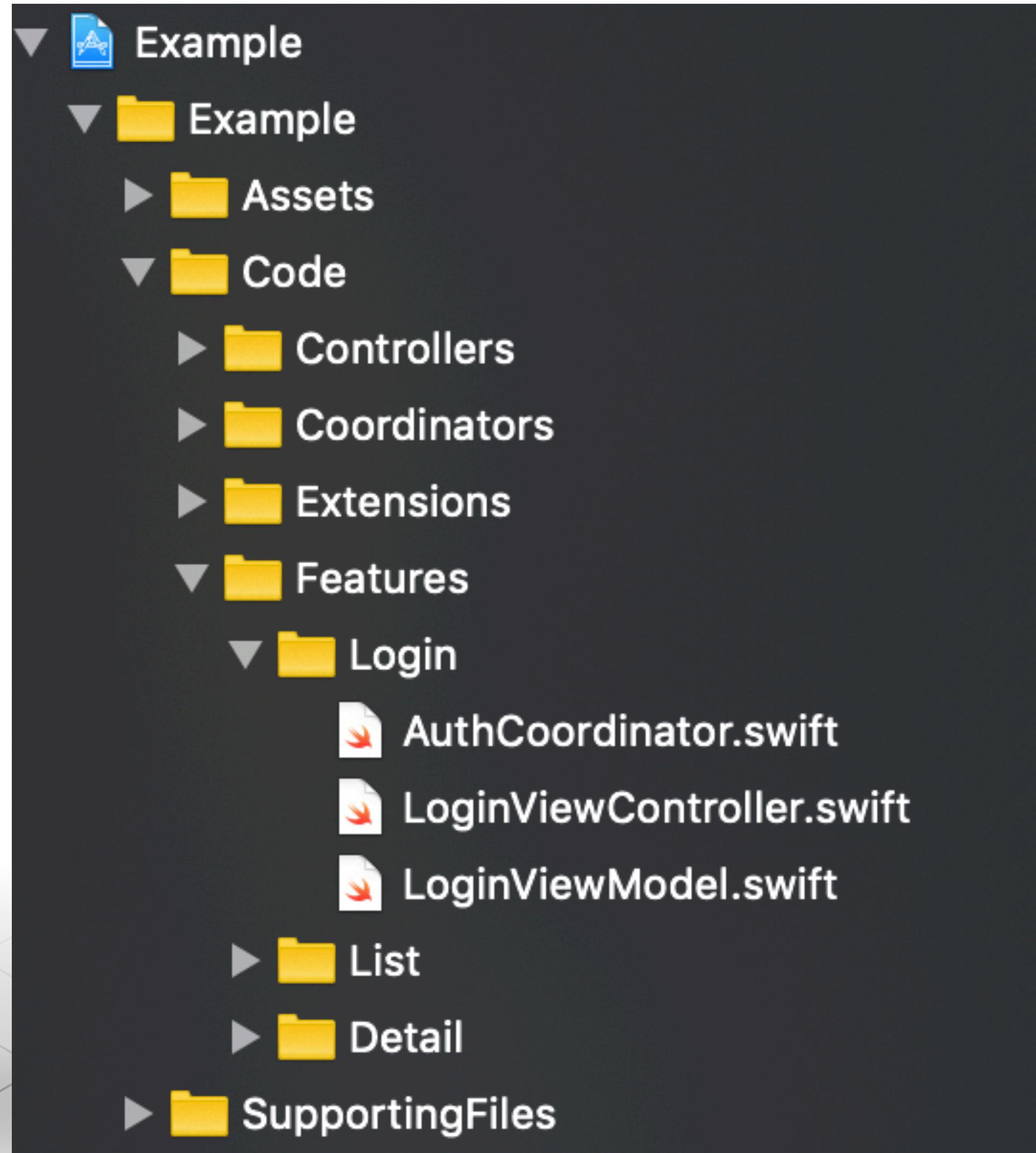
- Feature Structure
- Kit Framework
- Project Template



github.com/allaboutapps/ios-project-template



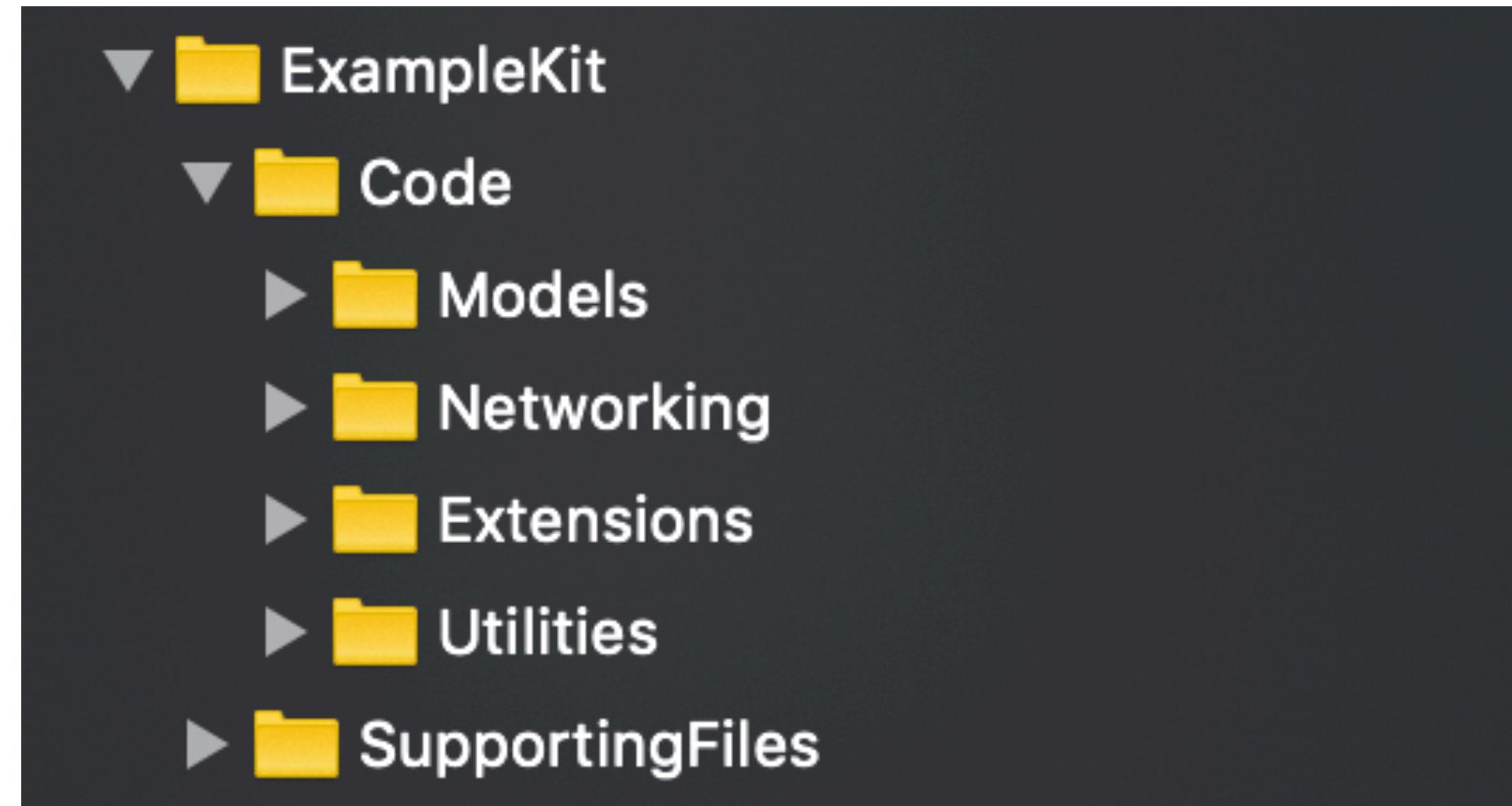
Feature Structure



- Separation
- Reuse
- Collaboration



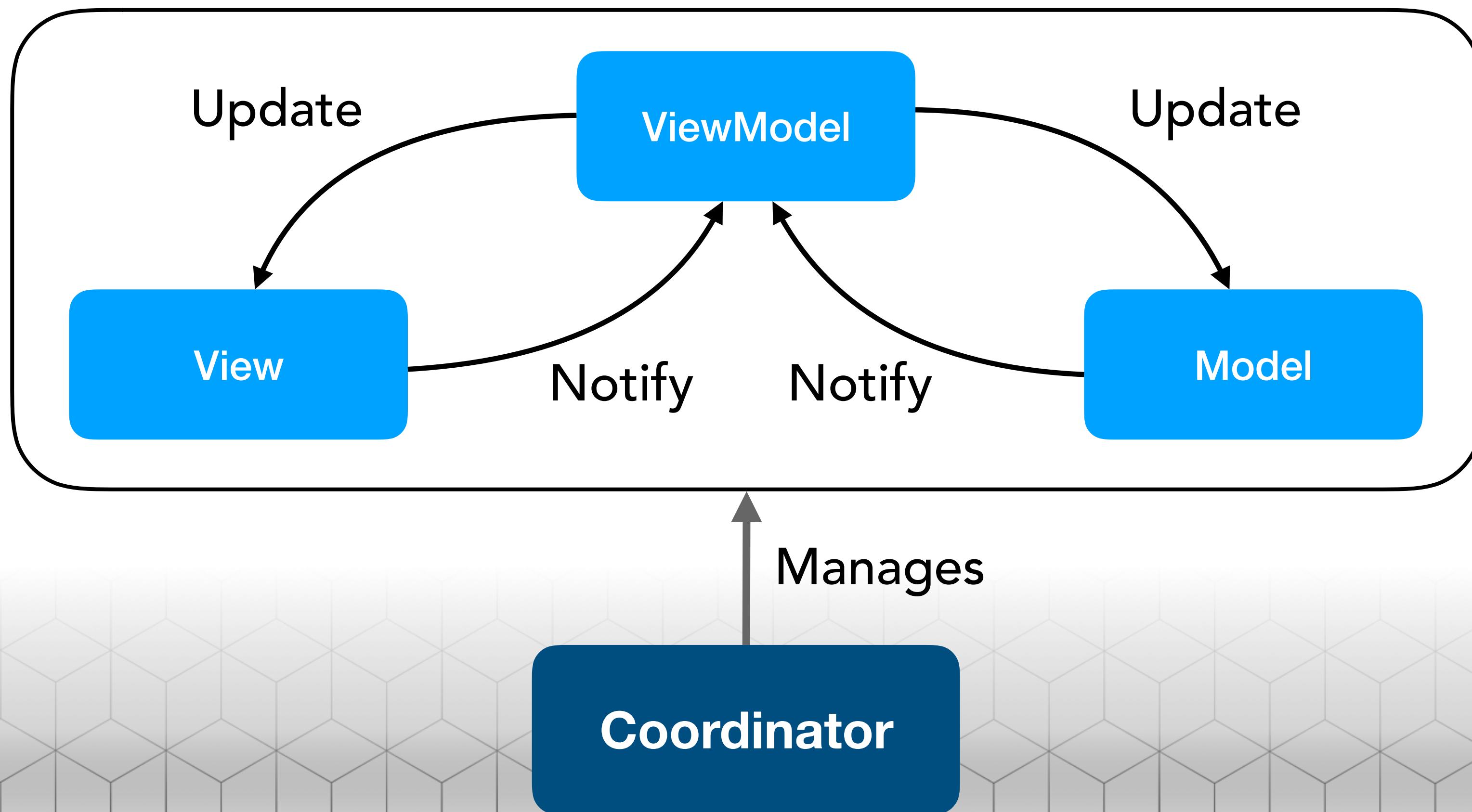
Kit Framework



- Embedded framework
- Encapsulation
- Modularity
- Reuse

MVVM + C

Model-View-ViewModel + Coordinator



Register Example

19:05

Cancel

Email address

Tell us your email address

Email address
michael.heinzl@allaboutapps.at

Or register using Facebook

 Register with facebook

CONTINUE



19:05

< Email address

Password

Enter your password

Password
••••••••

CONTINUE



19:06

< Password

First name

Tell us your first name

First name
Michael

CONTINUE



19:06

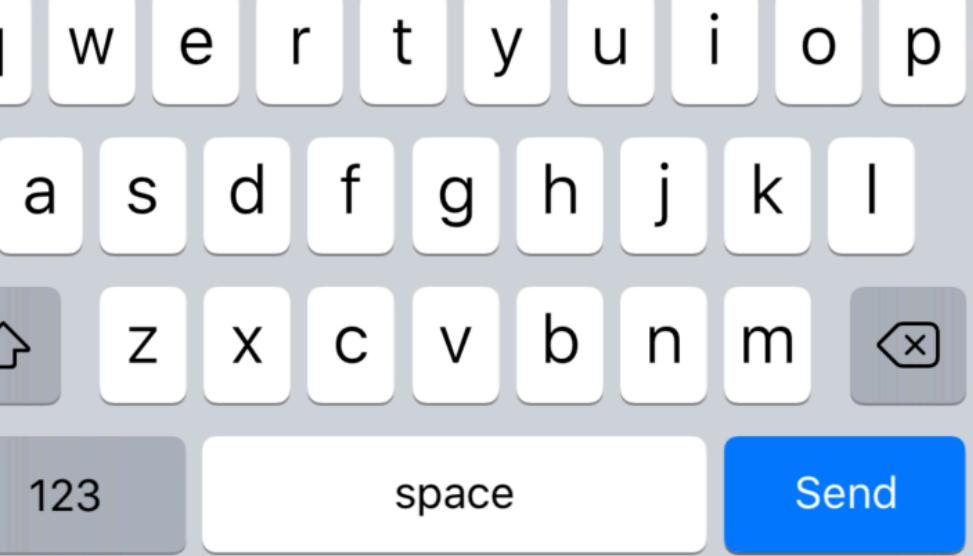
< First name

Last name

Tell us your last name

Last name
Heinzl

CONTINUE





Register Example

```
class RegisterCoordinator: NavigationCoordinator {

    enum Step {
        case email
        case password
        case firstName
        case lastName
    }

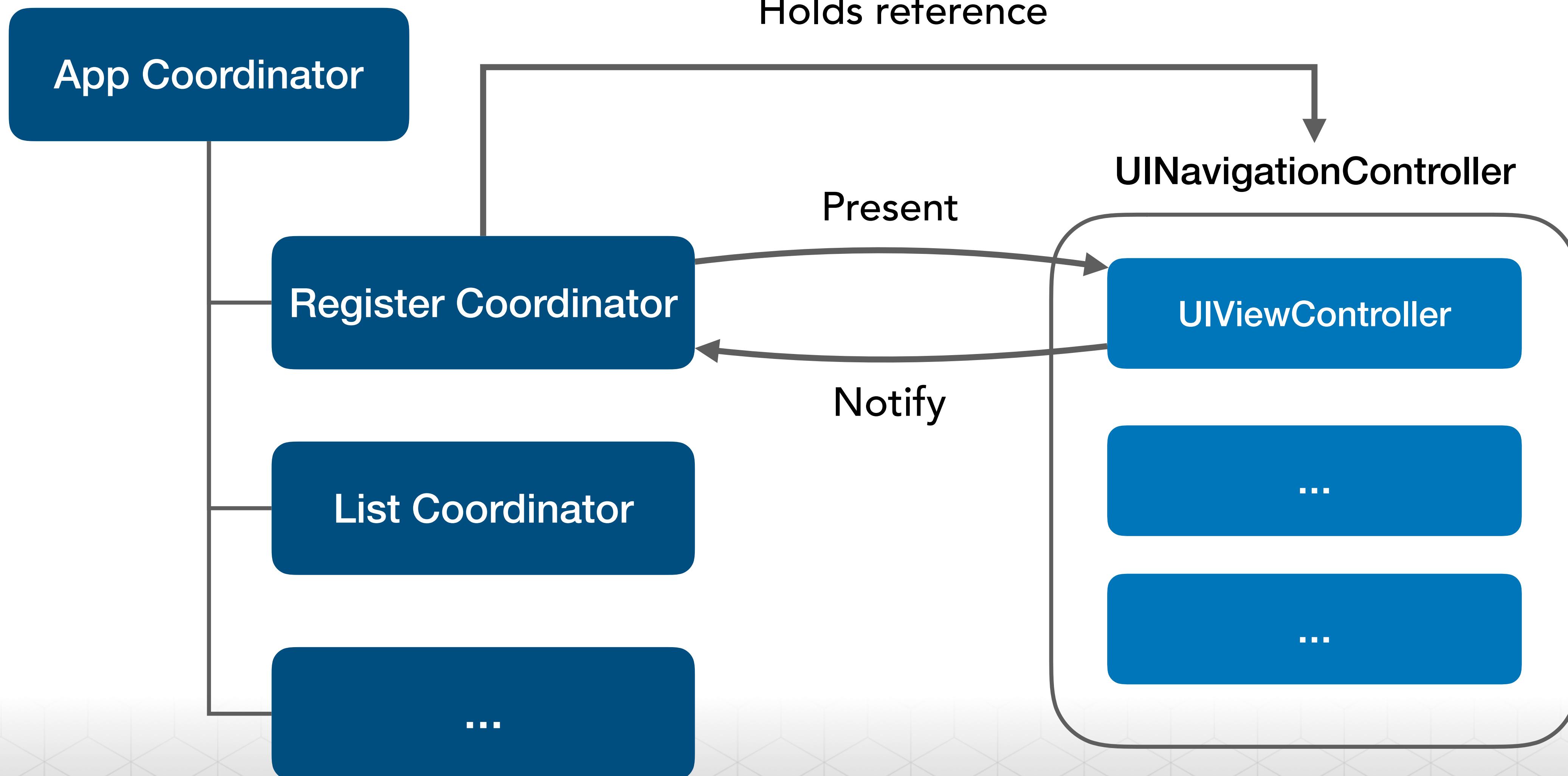
    var didFinish: ((User) -> Void)?
    var didCancel: (() -> Void)?

    let steps: [Step] = [.email, .password, .firstName, .lastName]

    func start() {
        showNext(step: .email)
    }

    func showNext(step: Step) {
        let viewController = createViewController(with: step)
        push(viewController, animated: true)
    }

    // ...
}
```



Reusable View Controllers

```
class TextInputViewController: UIViewController {

    // MARK: - Output

    var didEnterText: ((String) -> Void)?

    // MARK: - Input

    var viewModel: TextInputViewModel!

    class func create(viewModel: TextInputViewModel) -> Self {
        let viewController = UIStoryboard(.onboarding).instantiateViewController(self)
        viewController.viewModel = viewModel
        return viewController
    }
}
```

Why Coordinators?

- Reusability
- Maintainability
- Less responsibility
for view controllers
- Clean storyboards
- (Almost) no segues
- Better collaboration

DataSource

- Type-safe setup and configuration of table views
- UITableViewDataSource
- (optional) UITableViewDelegate
- github.com/allaboutapps/DataSource

Cell Descriptors

```
CellDescriptor<Example, TitleCell>()
    .configure { (example, cell, indexPath) in
        cell.textLabel?.text = example.title
        cell.accessoryType = .disclosureIndicator
    }
    .didSelect { (example, indexPath) in
        self.performSegue(withIdentifier: example.segue, sender: nil)
        return .deselect
    }
}
```

- Maps data (usually ViewModels) → UITableViewCells
- Provides all UITableViewDataSource + Delegate methods in a type-safe way
- Extend descriptors using a fluent interface

Data Setup

```
dataSource.sections = [
    Section(items: [
        Example(title: "Random Persons", segue: "showRandomPersons"),
        Example(title: "Form", segue: "showForm"),
        Example(title: "Lazy Rows", segue: "showLazyRows")
    ])
]
```

9:40



Examples

- Random Persons >
- Form >
- Lazy Rows >
- Diff & Update >
- Swipe To Delete >
- Custom separators >

Animated Diffs

1. Implement Diffable protocol

✓ if you already implement Hashable & Equatable

2. 🎉

Examples	Persons	
A - L		
	Megan Adams	
	Bill Casey	
	Leticia Casey	
	Yvonne Casey	
	William Cole	
	Yvonne Francis	
	Nancy Gross	
	Bill Lane	
M - Z		
	Geoffrey Patrick	
	Jim Patrick	

Row Visibility

[Examples](#)[Form](#)

First Name

Show additional options

```
TextFieldCell.descriptor  
    .isHidden { (field, indexPath) in  
        if field.id == self.lastNameField.id {  
            return self.firstNameField.text?.isEmpty ?? true  
        } else {  
            return false  
        }  
    }  
}
```



code.allaboutapps.at

Q & A

Matthias Buchetics

matthias.buchetics@allaboutapps.at

Michael Heinzl

michael.heinzl@allaboutapps.at