终于来15咯

UI Testing in Xcode

Session 406

在Xcode进行UI测试

Wil Turner Developer Tools
Brooke Callahan Developer Tools

© 2015 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.

Overview

UI testing UI测试

- Find and interact with UI elements 找到交互的UI元素
- Validate UI properties and state 验证UI的属性和状态

UI recording UI记录，这个是重点

Test reports 测试报告

Core Technologies

XCTest + Accessibility

Testing Matrix

NEW

	Unit	UI
Correctness	✓	✓
Performance	✓	✓

Accessibility

Rich semantic data about UI

UIKit and AppKit integration

APIs for fine tuning

UI tests interact with the app the way a user does

Getting Started

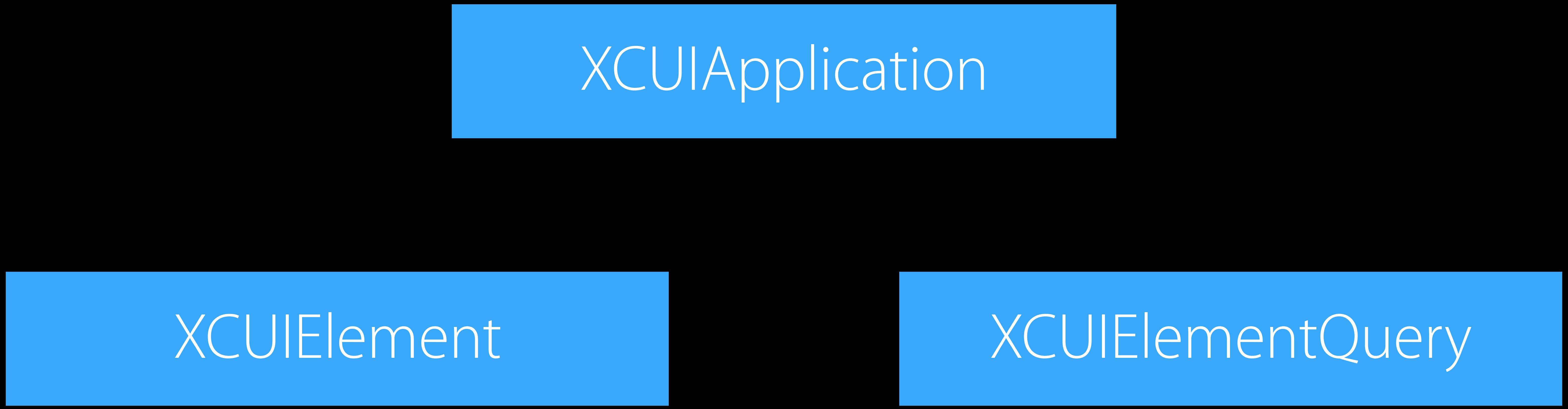
APIs

Three new classes

- XCUIApplication
- XCUIElement
- XCUIElementQuery

UI Testing API

UI Testing API



Example

Testing the Add button

代理App启动

```
// application:
let app = XCUIApplication()
app.launch()
```

```
// element and query:
let addButton = app.buttons[“Add”]
addButton.tap()
```

App.buttons 获取到 XCUIElementQuery 对象之后调用

```
- (XCUIElement *)objectForKeyedSubscript:(NSString *)key;

subscript(key: String) -> XCUIElement { get }
```

```
// assertion:
XCTAssertEqual(app.tables.cells.count, 1)
```

断言进行判断

Example

Testing the Add button

```
// application:
let app = XCUIApplication()
app.launch()

// element and query:
let addButton = app.buttons[“Add”]
addButton.tap()

// assertion:
XCTAssertEqual(app.tables.cells.count, 1)
```

Example

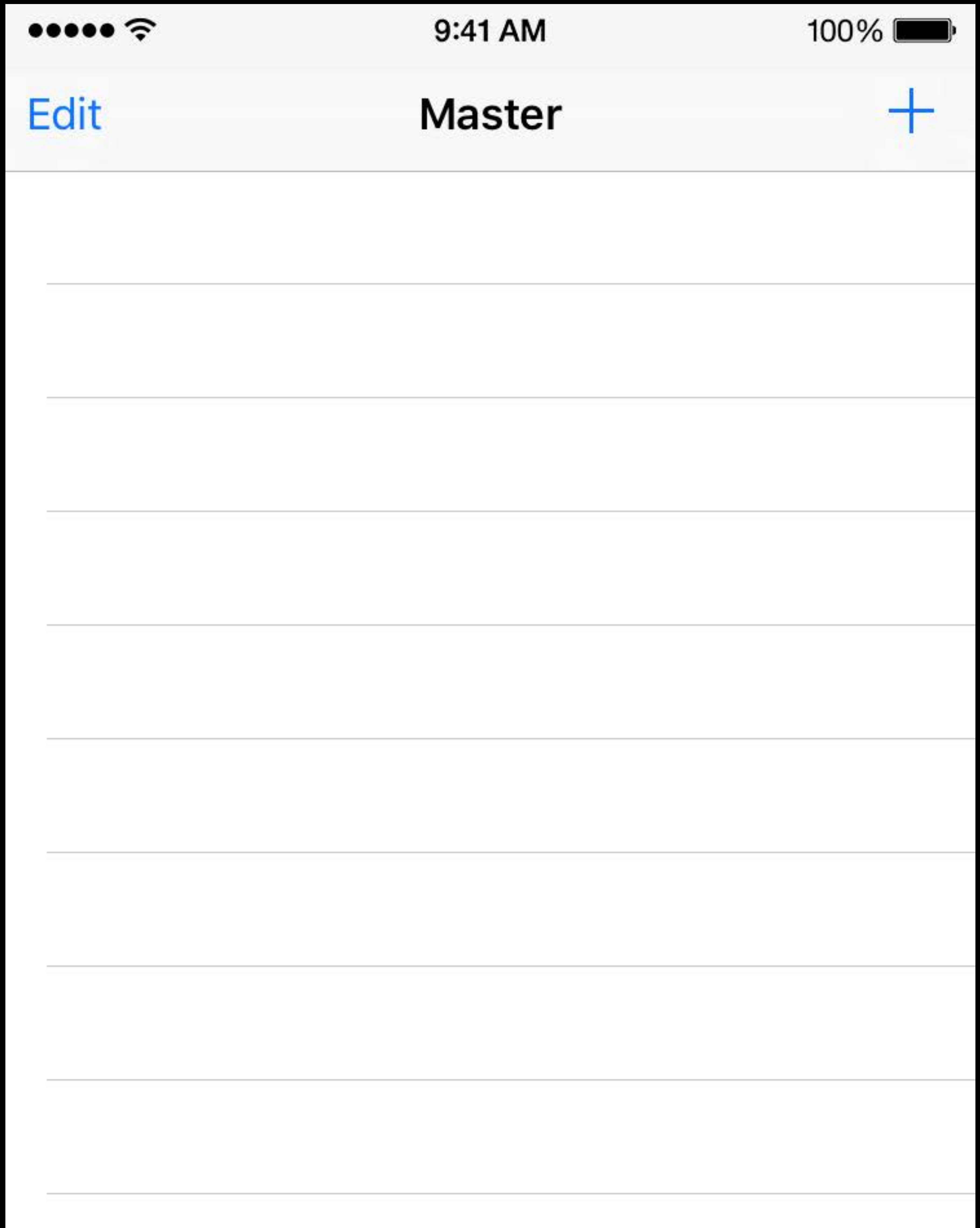
Testing the Add button

```
// application:
let app = XCUIApplication()
app.launch()

// element and query:
let addButton = app.buttons[“Add”]
addButton.tap()

// assertion:
XCTAssertEqual(app.tables.cells.count, 1)
```

Example

Testing the Add button

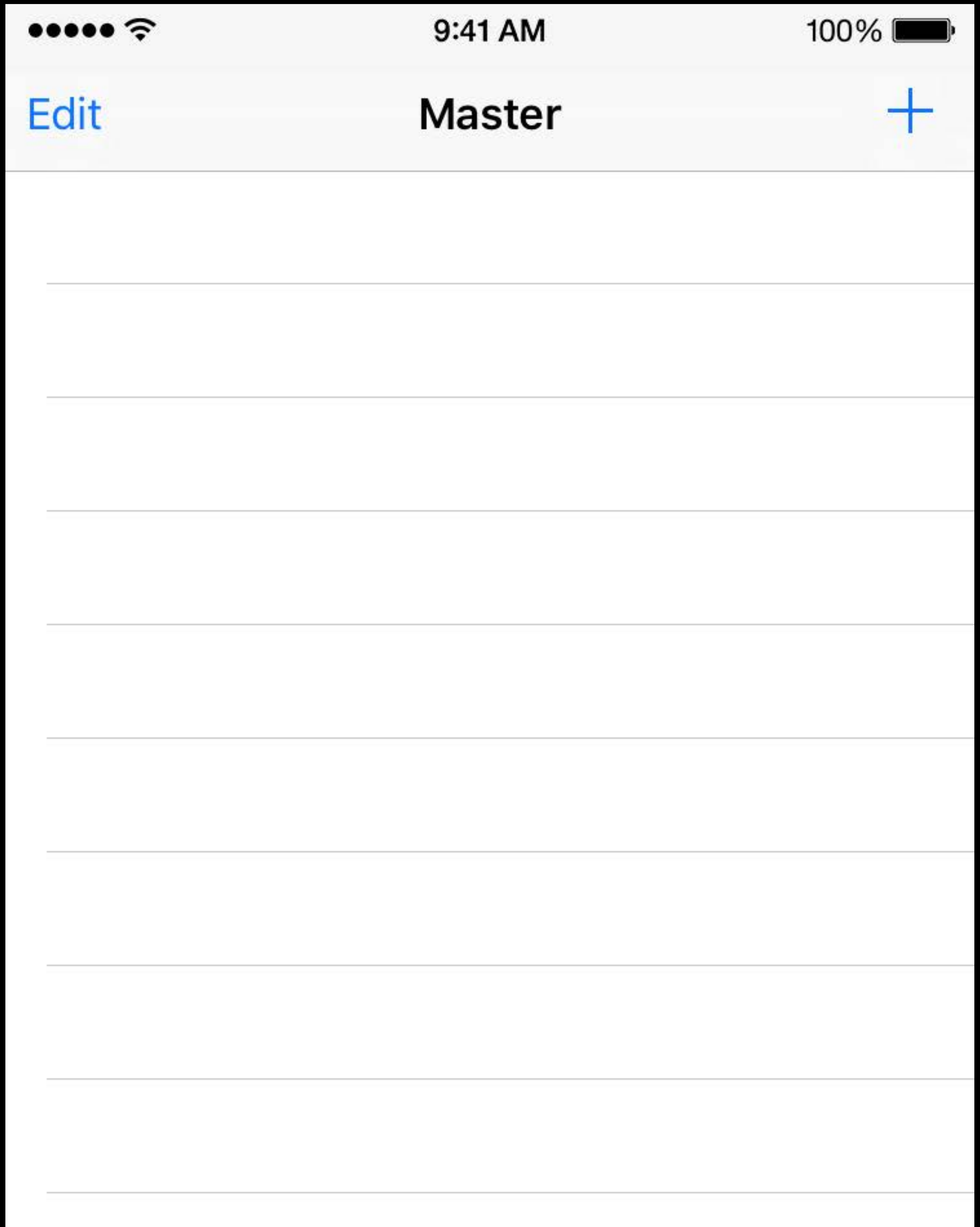
```
// application:
let app = XCUIApplication()
app.launch()

// element and query:
let addButton = app.buttons[“Add”]
addButton.tap()

// assertion:
XCTAssertEqual(app.tables.cells.count, 1)
```



Example

Testing the Add button

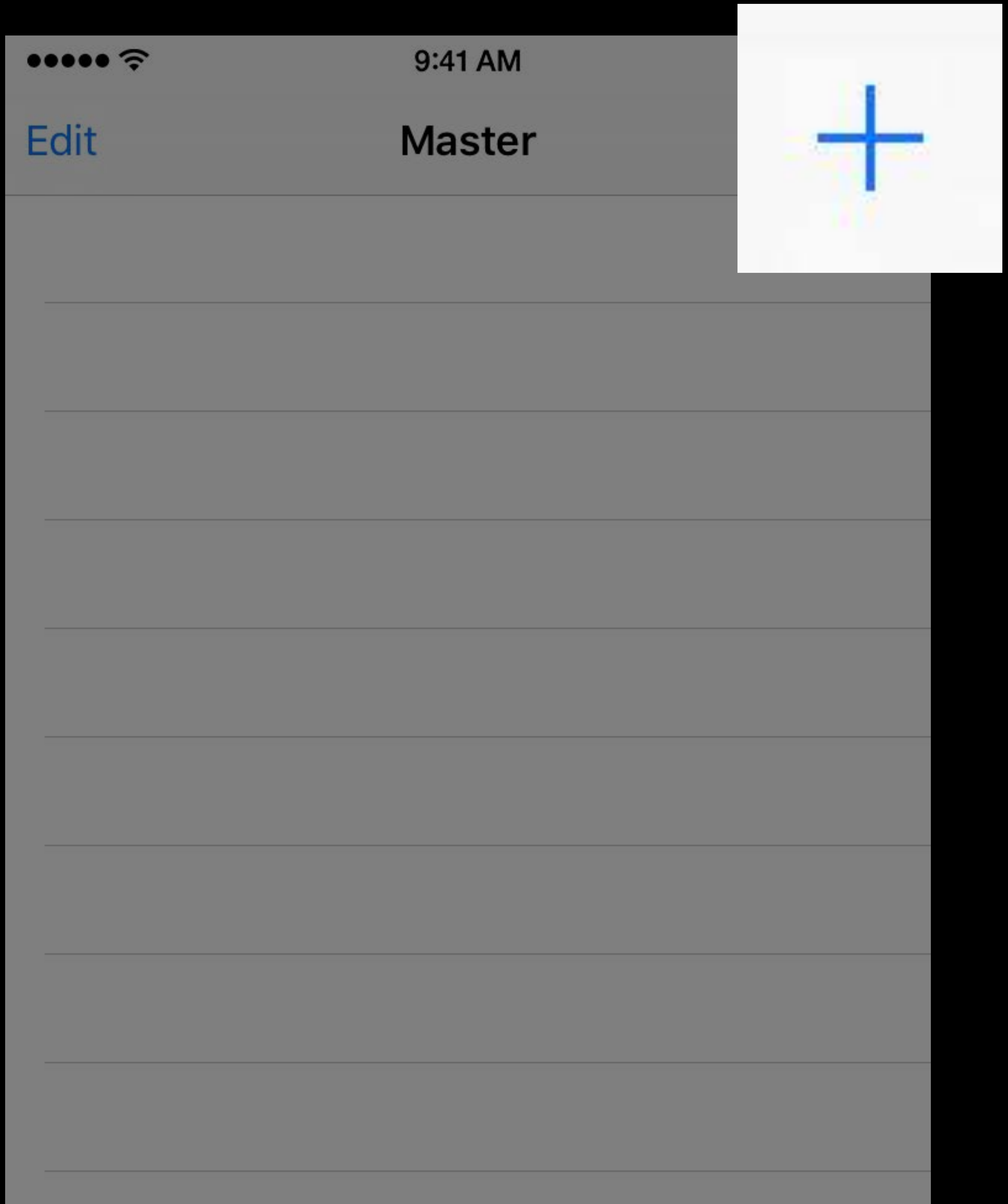
```
// application:
let app = XCUIApplication()
app.launch()

// element and query:
let addButton = app.buttons[“Add”]
addButton.tap()

// assertion:
XCTAssertEqual(app.tables.cells.count, 1)
```



Example

Testing the Add button

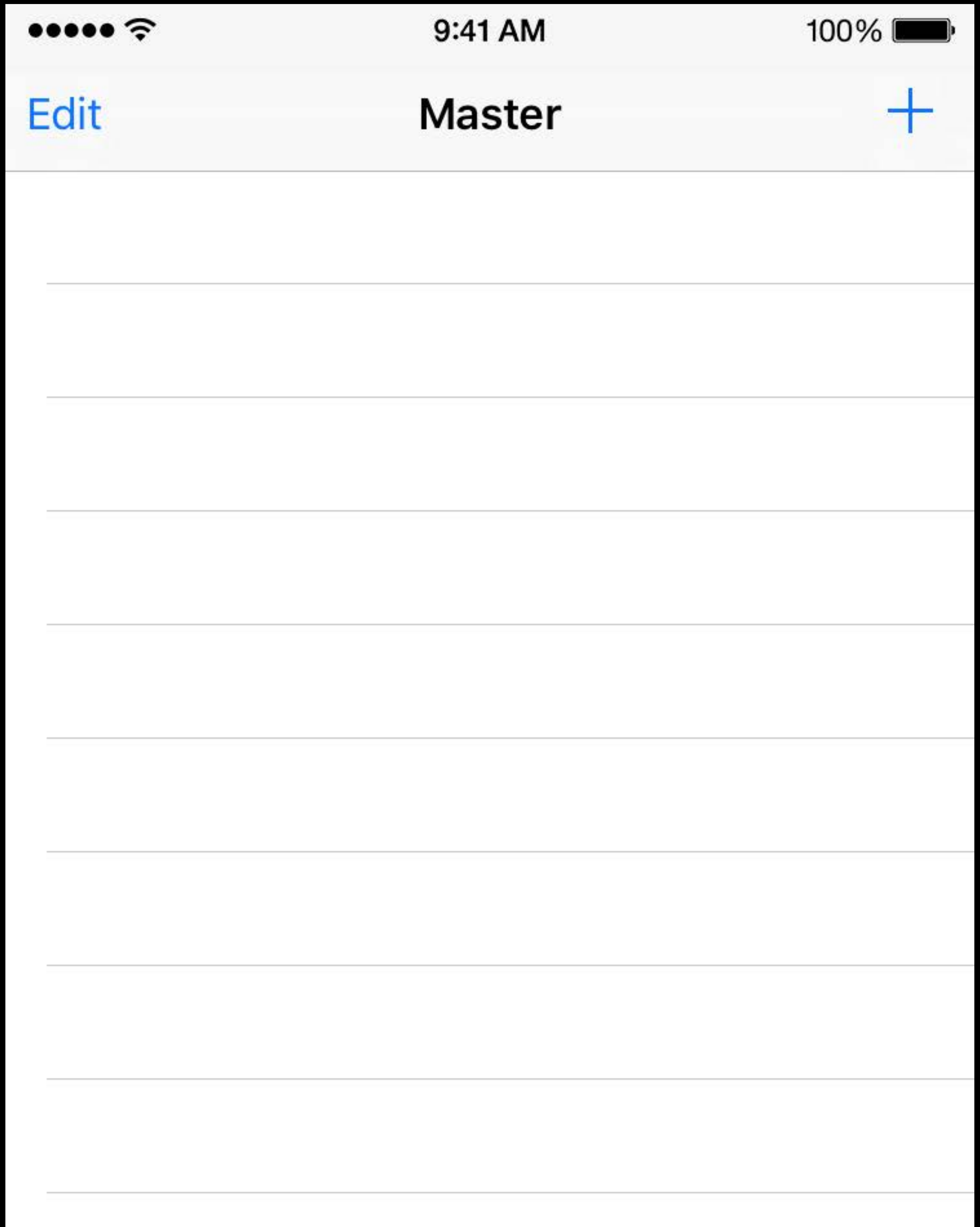
```
// application:
let app = XCUIApplication()
app.launch()

// element and query:
let addButton = app.buttons[“Add”]
addButton.tap()

// assertion:
XCTAssertEqual(app.tables.cells.count, 1)
```



Example

Testing the Add button

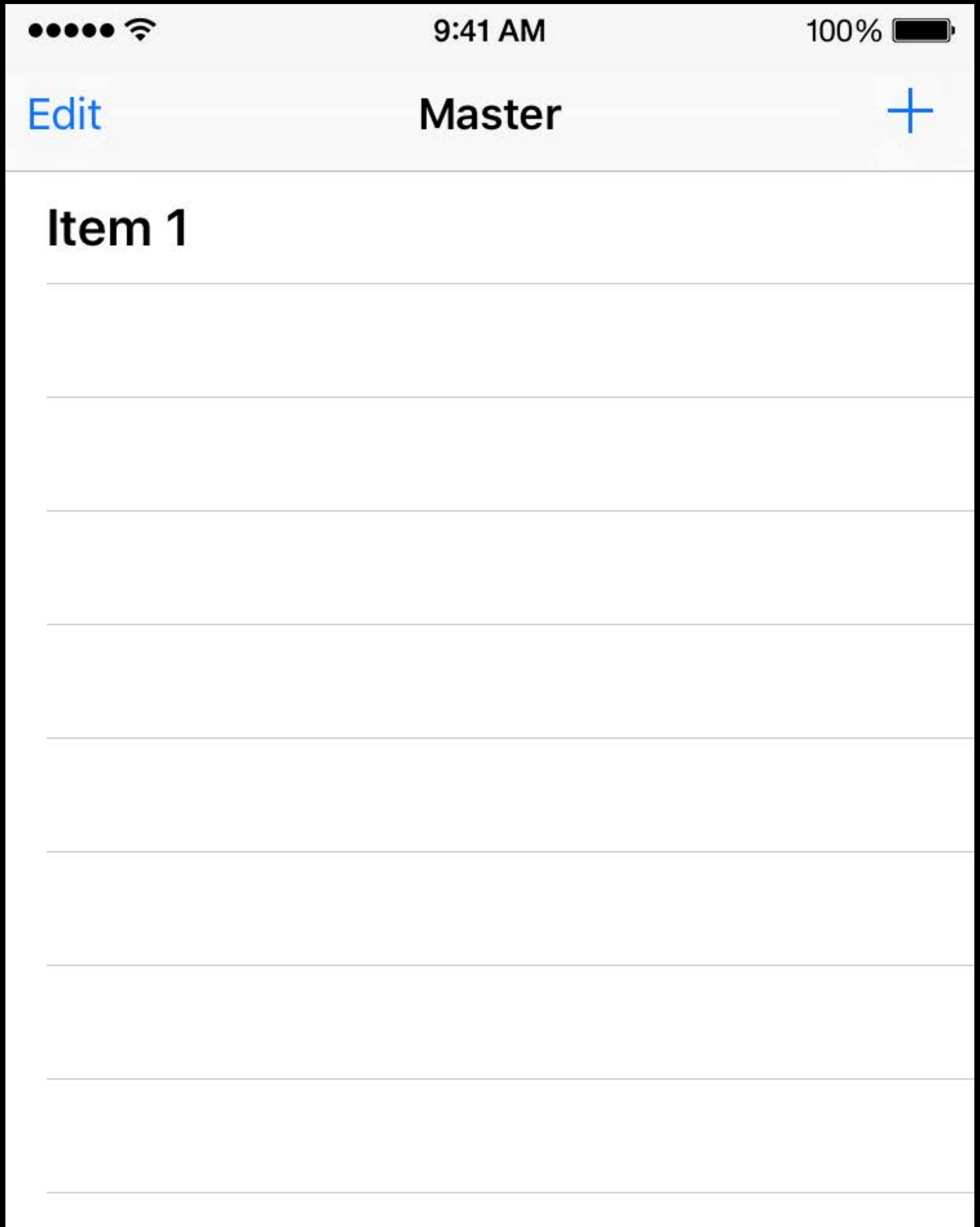
```
// application:
let app = XCUIApplication()
app.launch()

// element and query:
let addButton = app.buttons[“Add”]
addButton.tap()

// assertion:
XCTAssertEqual(app.tables.cells.count, 1)
```



Example

Testing the Add button

```
// application:
let app = XCUIApplication()
app.launch()

// element and query:
let addButton = app.buttons[“Add”]
addButton.tap()

// assertion:
XCTAssertEqual(app.tables.cells.count, 1)
```

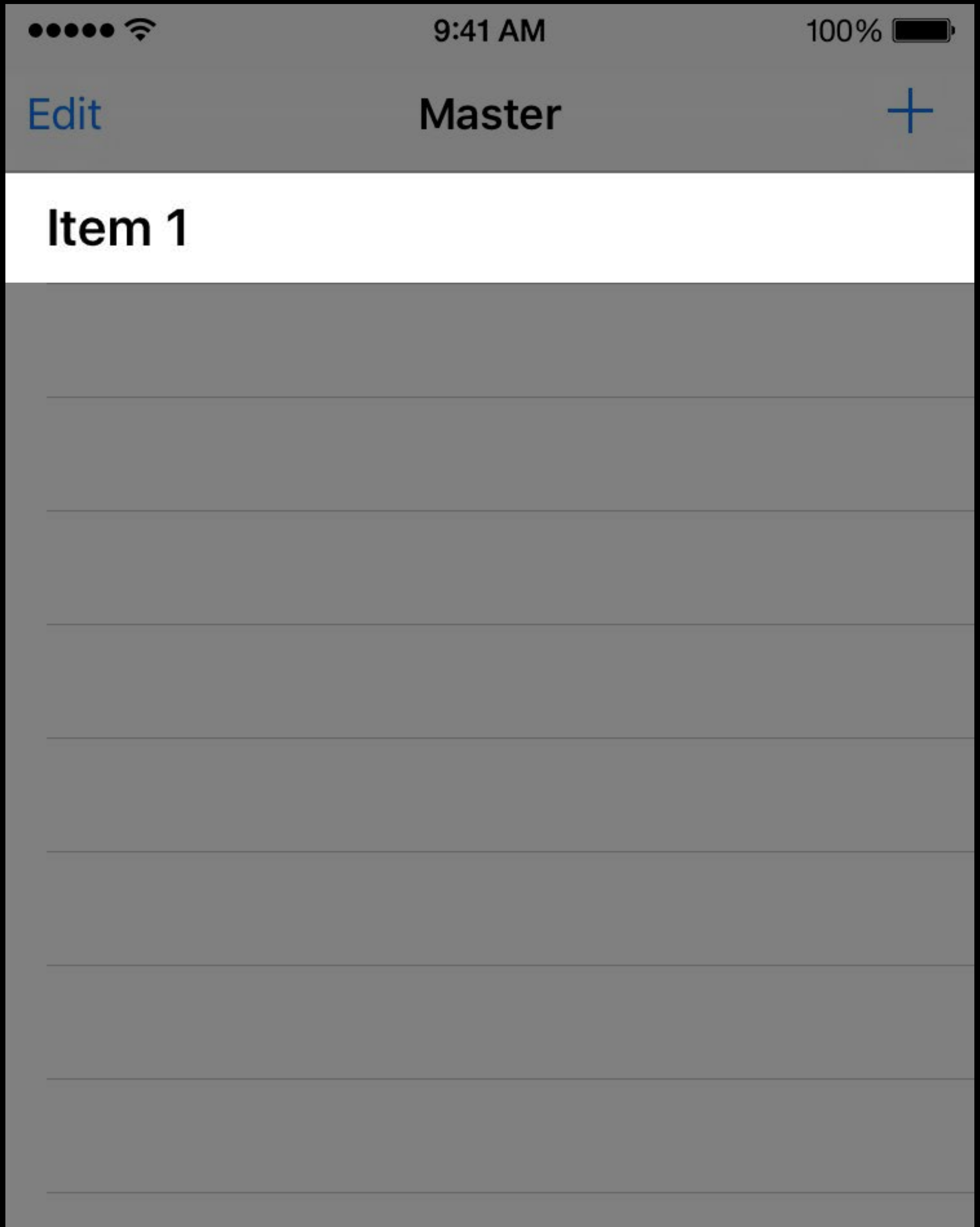


Example

Testing the Add button

```
// application:
let app = XCUIApplication()
app.launch()

// element and query:
let addButton = app.buttons[“Add”]
addButton.tap()

// assertion:
XCTAssertEqual(app.tables.cells.count, 1)
```

✓



XCUIApplication

XCUIApplication

Proxy for the tested application 测试应用的代理

- Tests run in a separate process 在隔离的进程中测试

Launch

- Always spawns a new process 总是产生一个新的进程
- Implicitly terminates any preexisting instance 绝对终止任何之前的实例

Starting point for finding elements 开始寻找元素

XCUIElement

XCUIElement

Proxy for elements in application 为应用中的元素进行代理

Types

- Button, Cell, Window, etc. 按钮、Cell、窗口

Identifiers 身份

- Accessibility identifier, label, title, etc. 身份、标签、按钮

XCUIElement

Proxy for elements in application

Types

- Button, Cell, Window, etc.

Identifiers

- Accessibility identifier, label, title, etc.

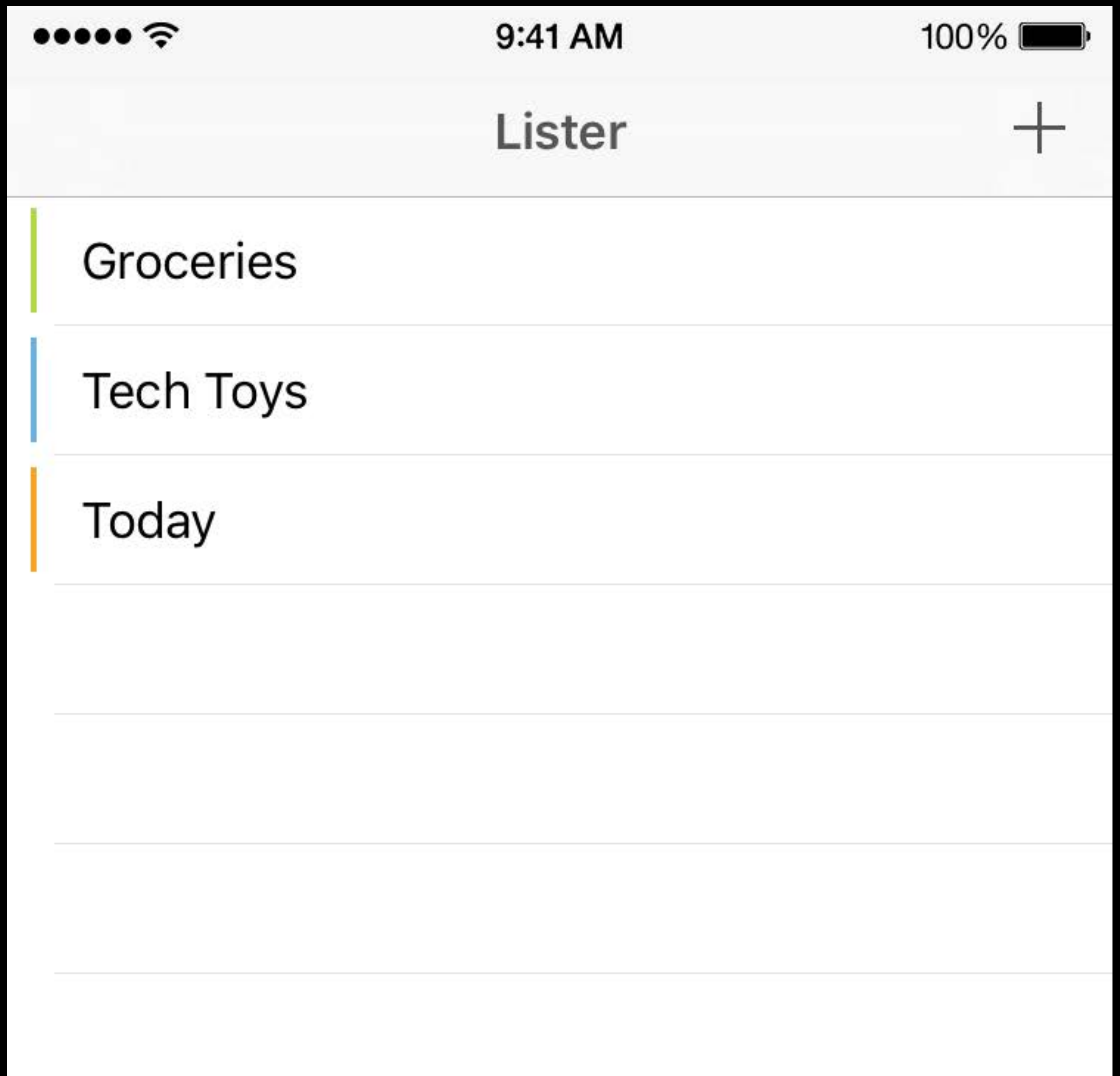
Most elements are found by combining type and identifier

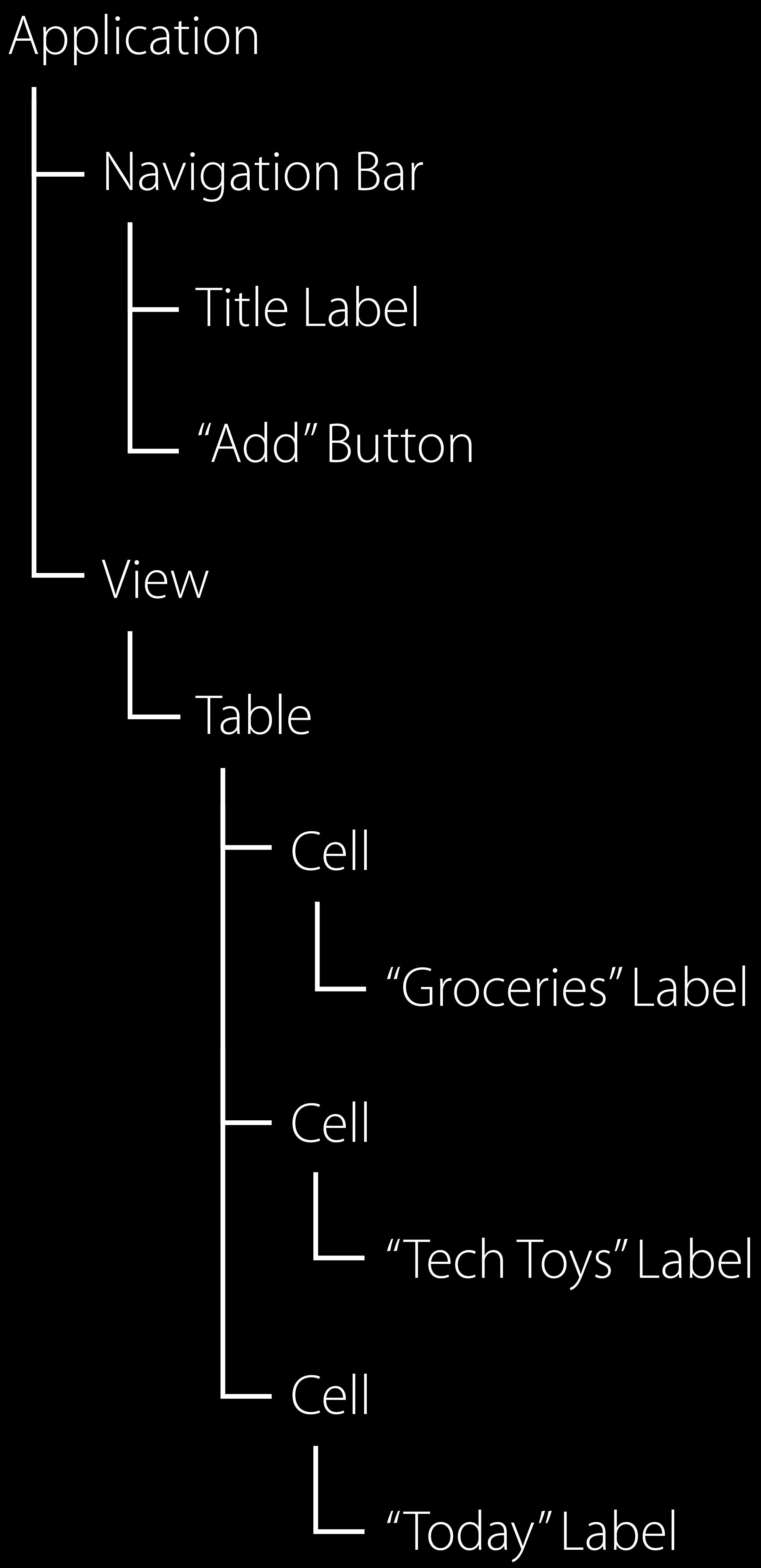
大多数元素的查找依靠类型和身份标识

Element Hierarchy

元素层级

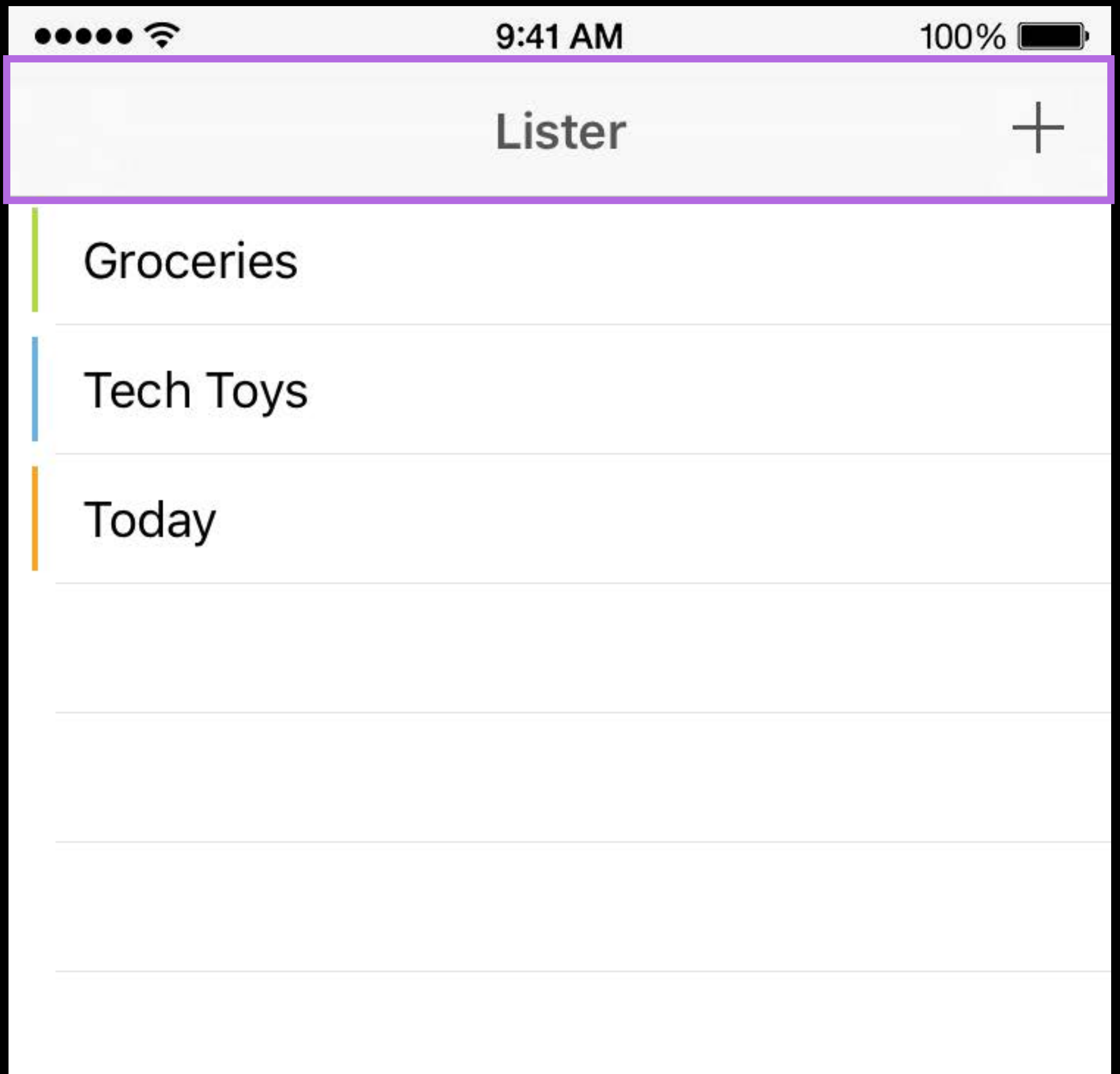
Application is the root of a tree of elements

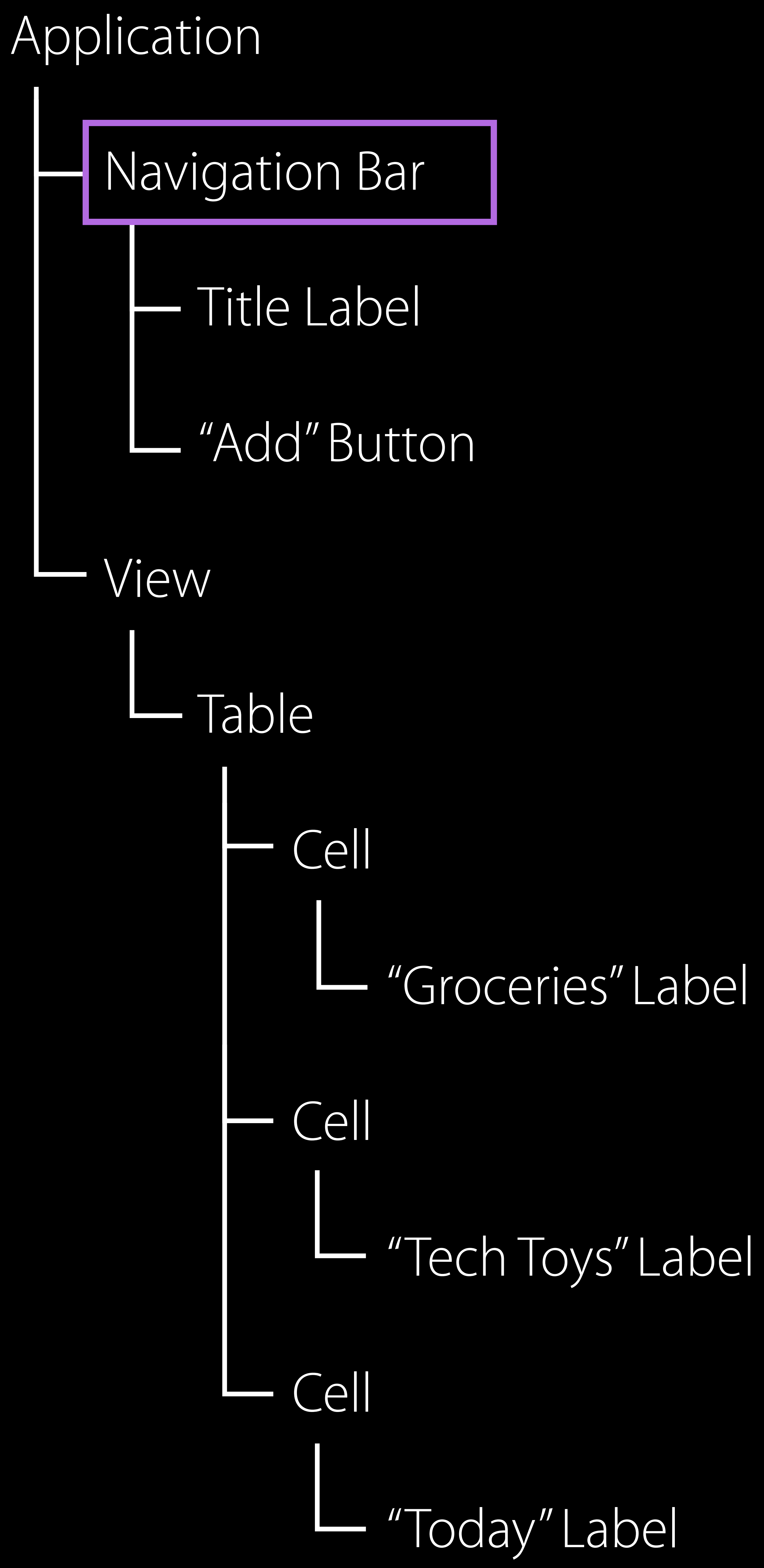




Element Hierarchy

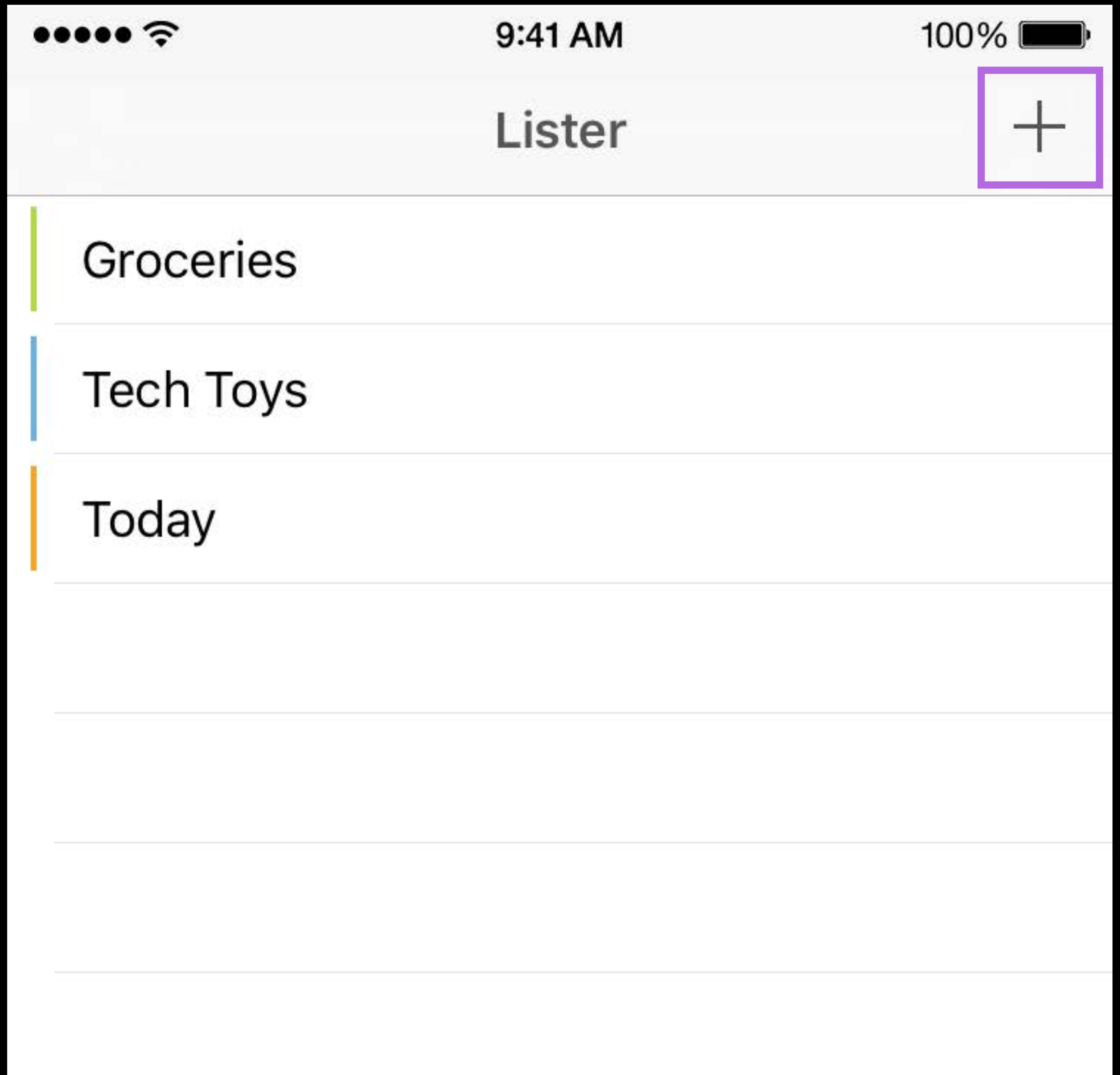
Application is the root of a tree of elements

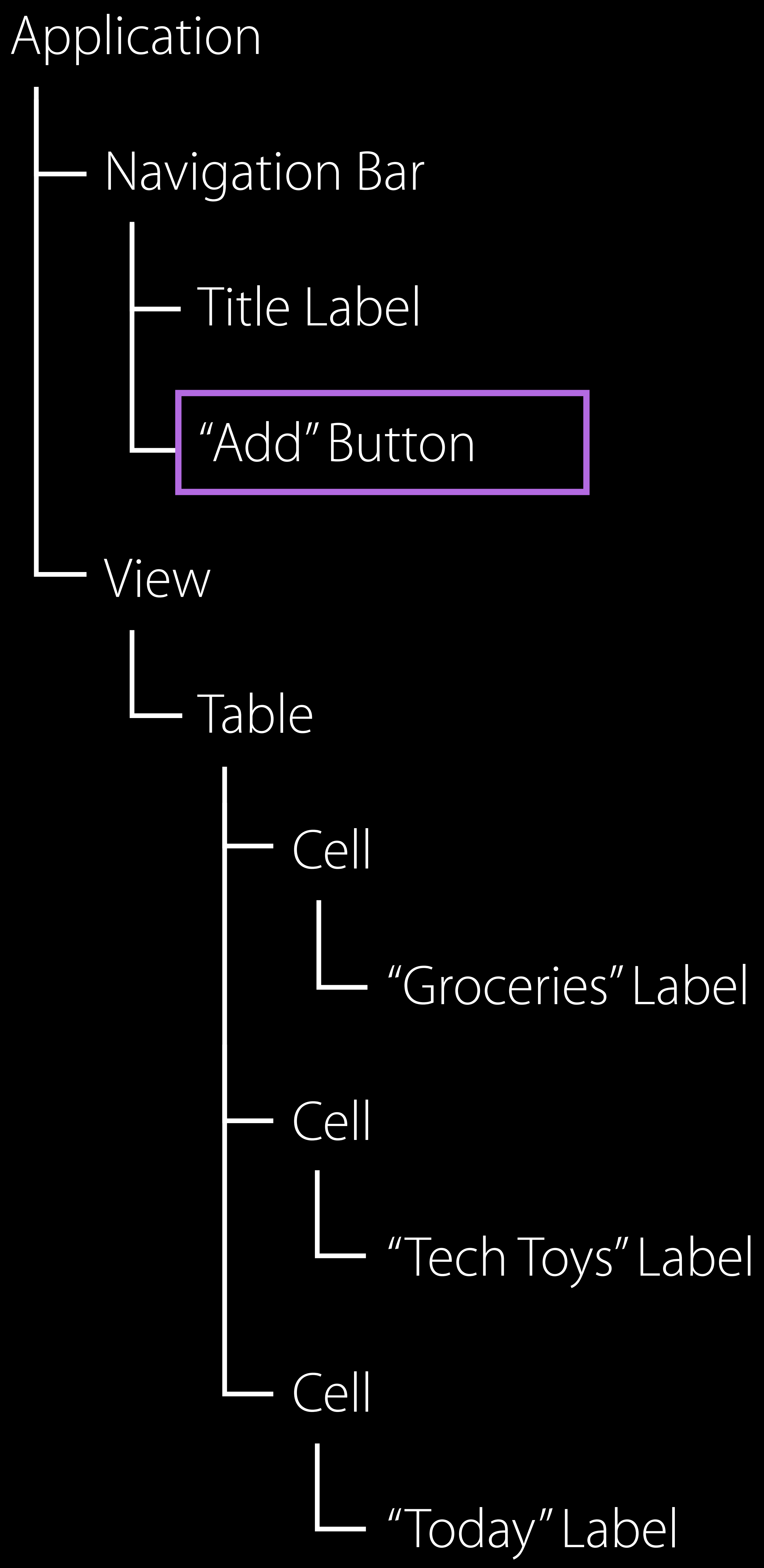




Element Hierarchy

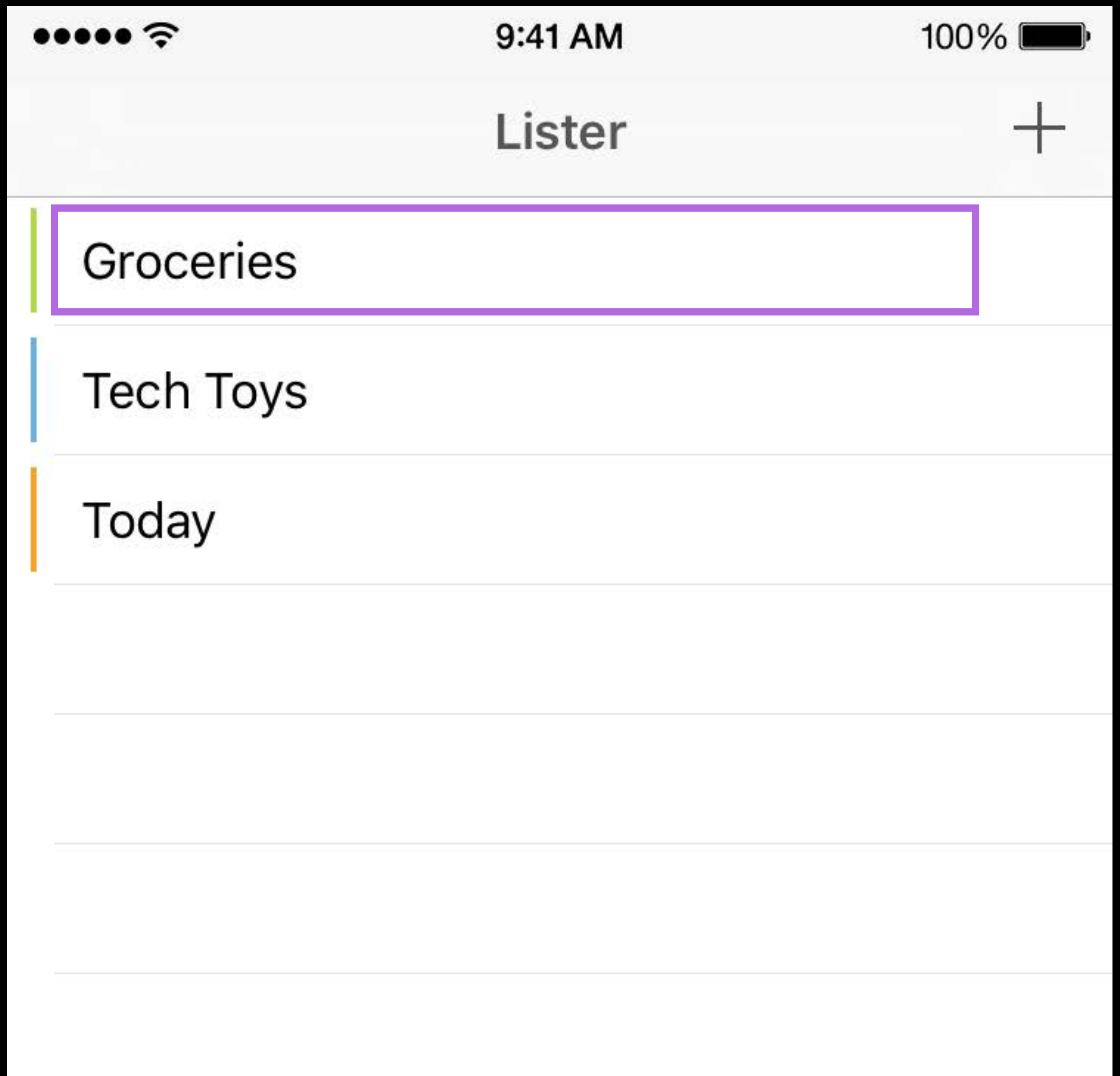
Application is the root of a tree of elements

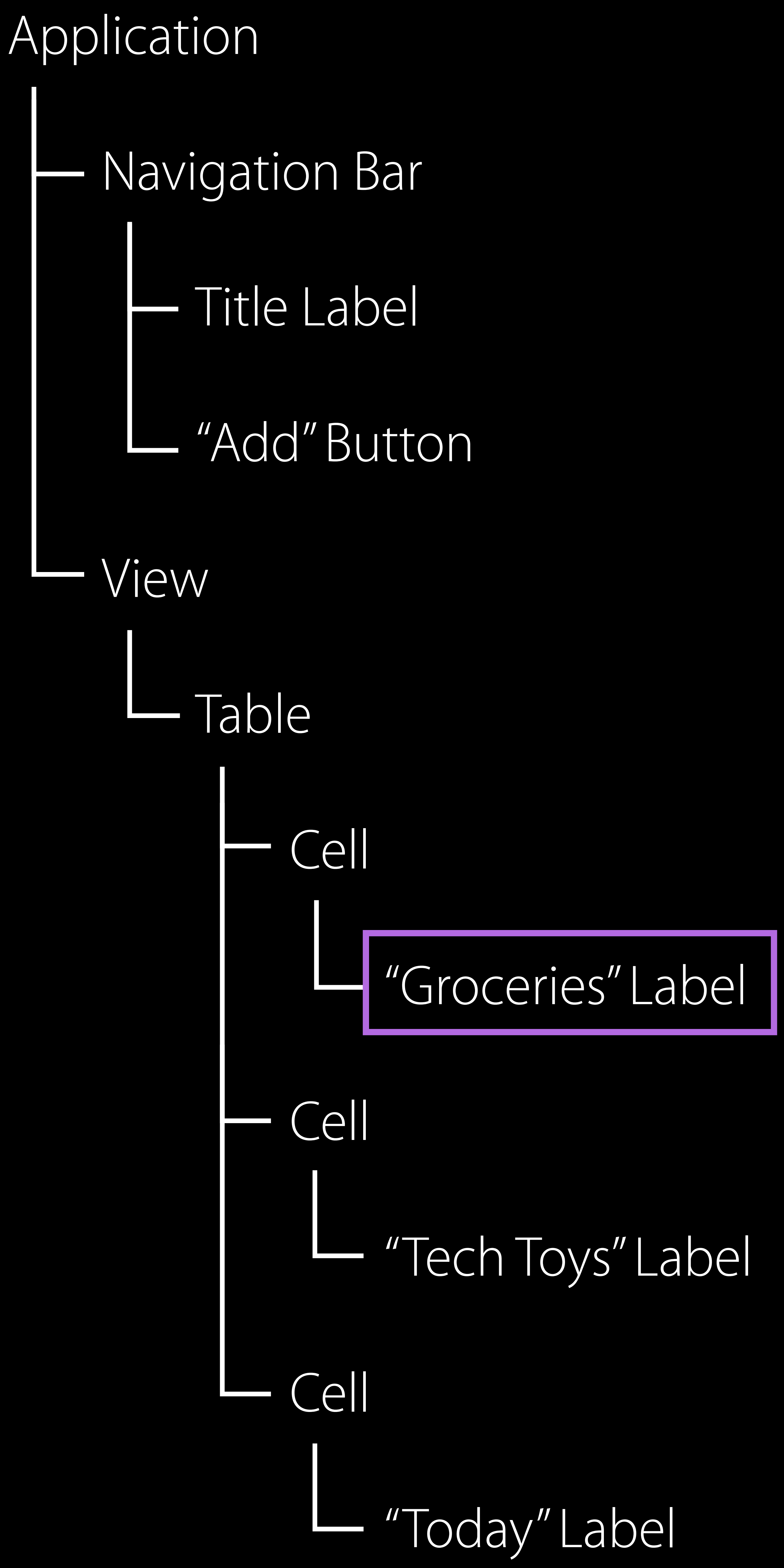




Element Hierarchy

Application is the root of a tree of elements

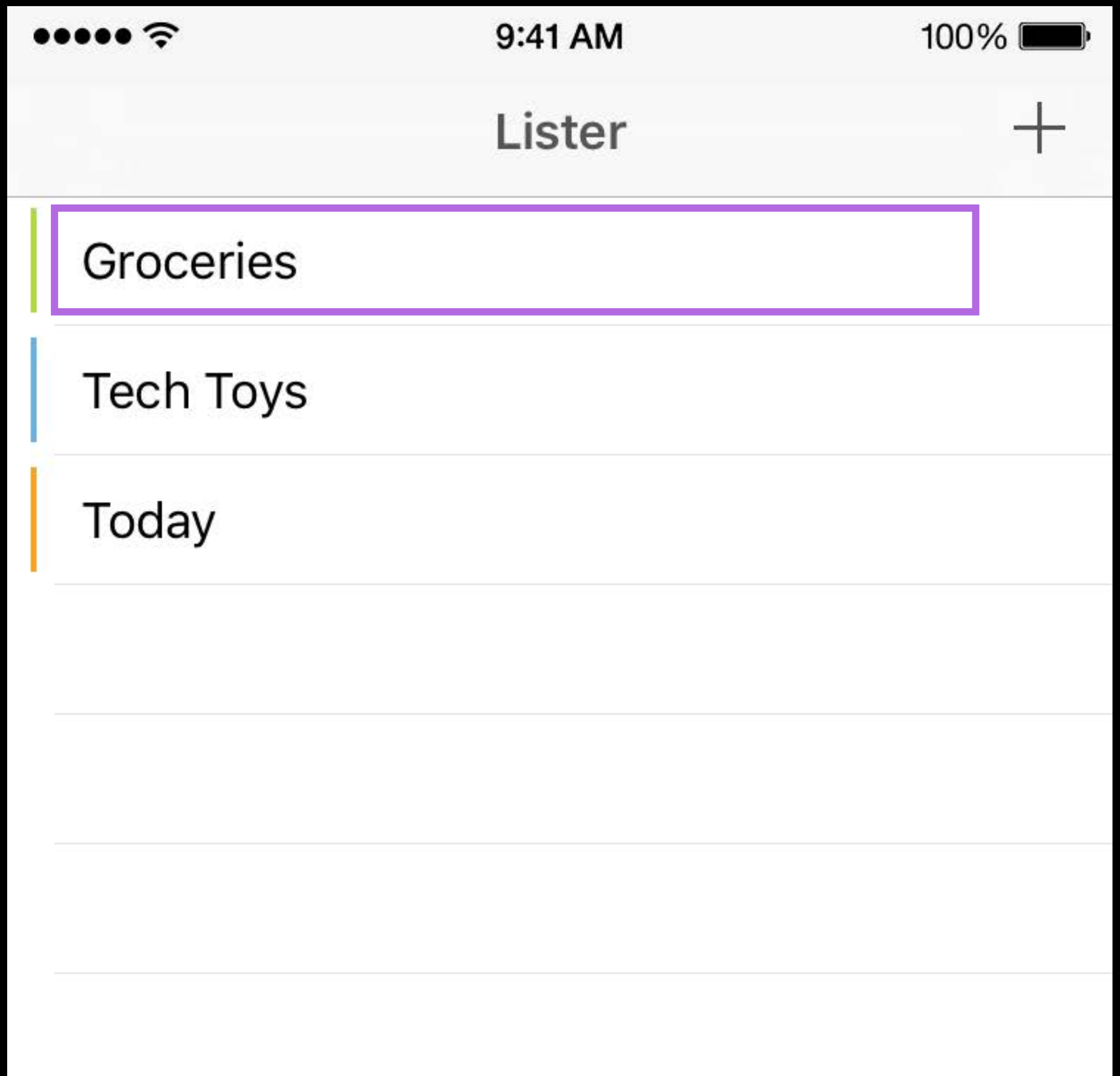


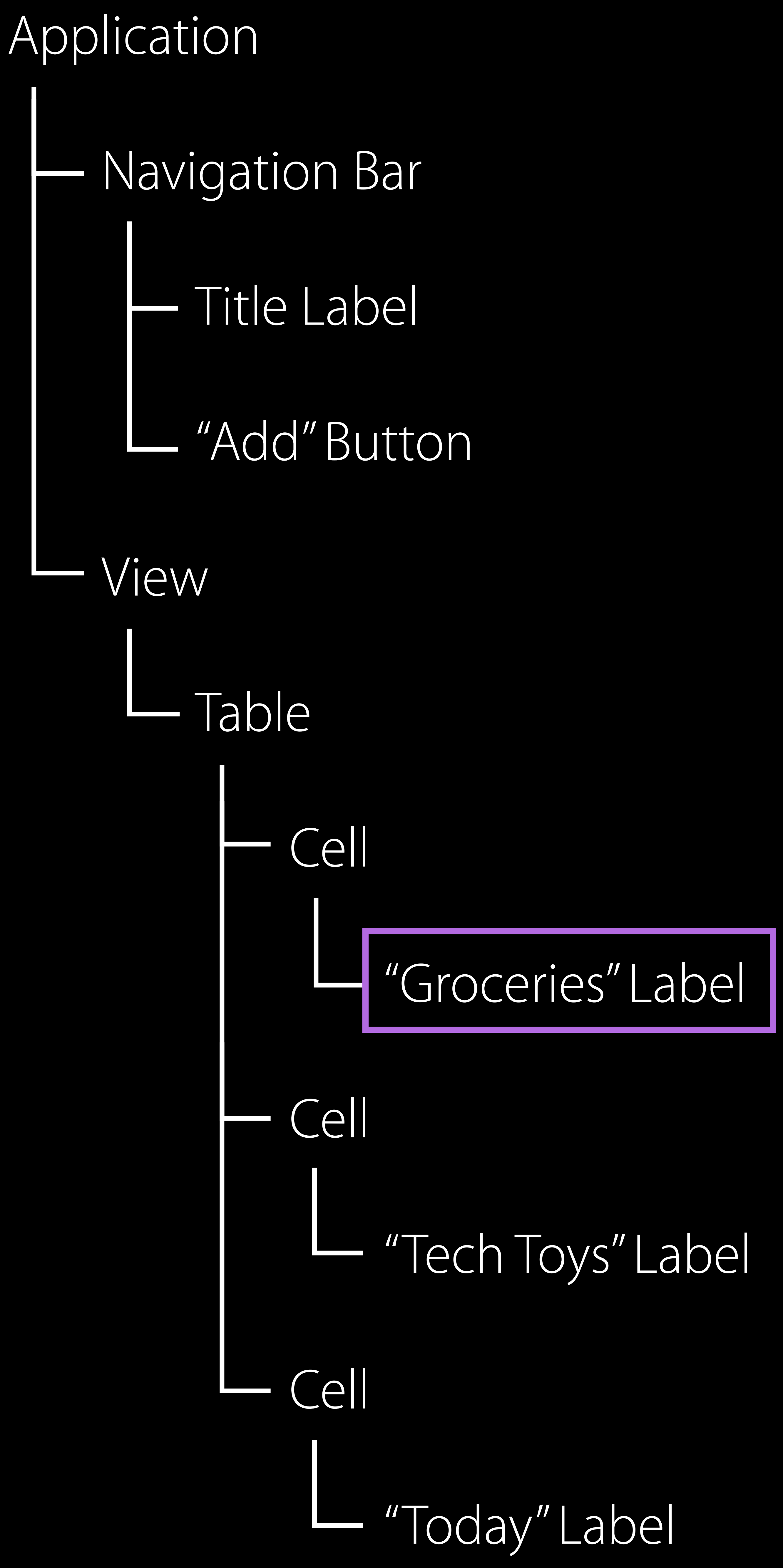


Element Hierarchy

Application is the root of a tree of elements

Used by queries with type and identifiers





Element Uniqueness

Element Uniqueness
元素唯一性

Every XCUIElement is backed by a query 所有的元素支持被询问

Query must resolve to exactly one match 询问必须确定唯一匹配

- No matches or multiple matches cause test failure 没有匹配或多匹配会导致测试错误
- Failure raised when element resolves query

Exception

- `exists` property

Event Synthesis

Simulate user interaction on elements

APIs are platform-specific

```
button.click() // OS X
button.tap() // iOS
textField.typeText(“Hello, World!”) // iOS & OS X
```

XCUIElementQuery

API for specifying elements

Queries resolve to collections of accessible elements

- Number of matches: `count`
- Specify by identifier: subscripting

XCUIElementQuery

API for specifying elements

Queries resolve to collections of accessible elements 询问可进入元素的集合

- Number of matches: `count` 匹配的计数
- Specify by identifier: subscripting 通过识别码查阅
- Specify by index: `elementAtIndex()`

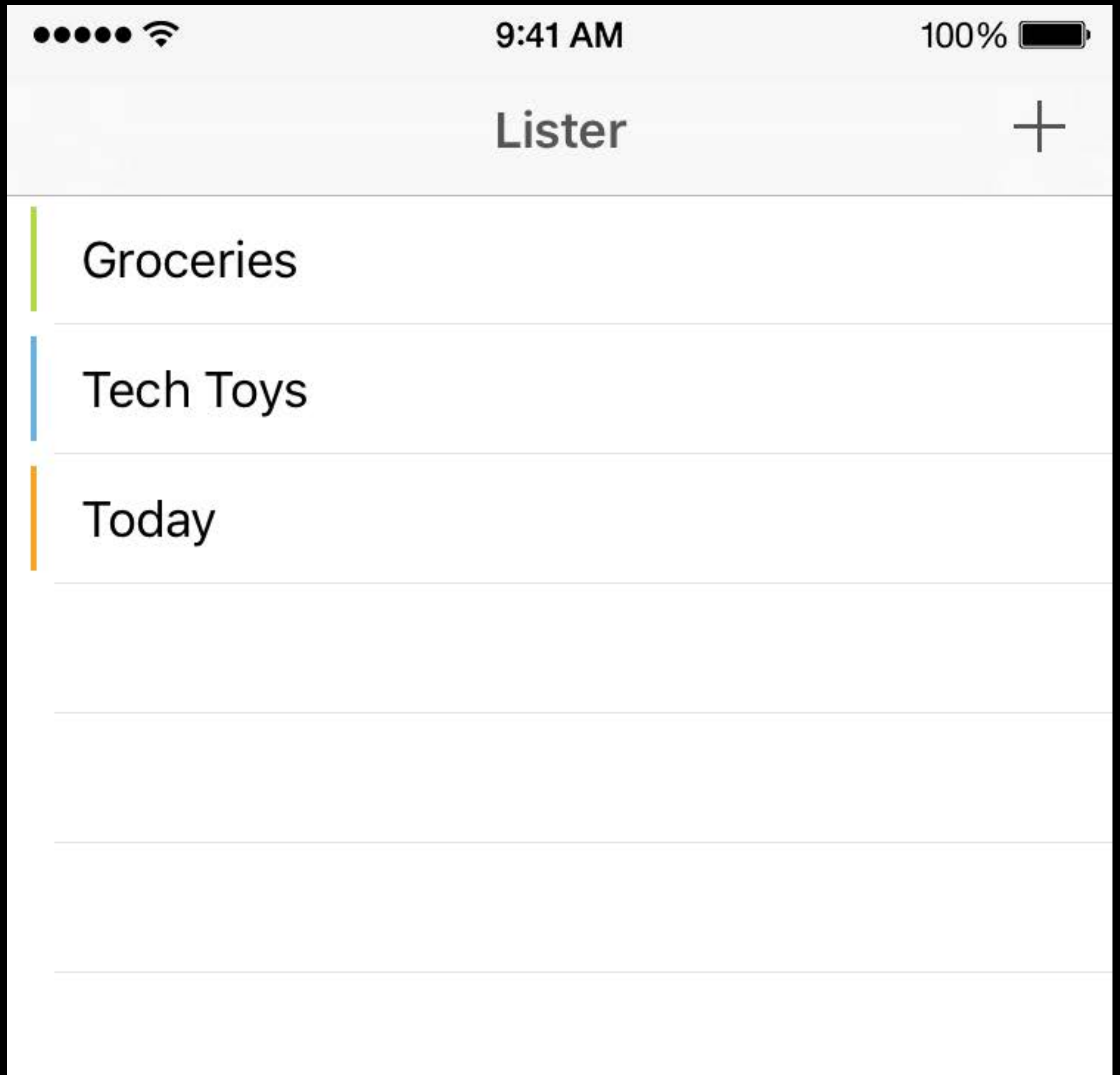
XCUIElementQuery

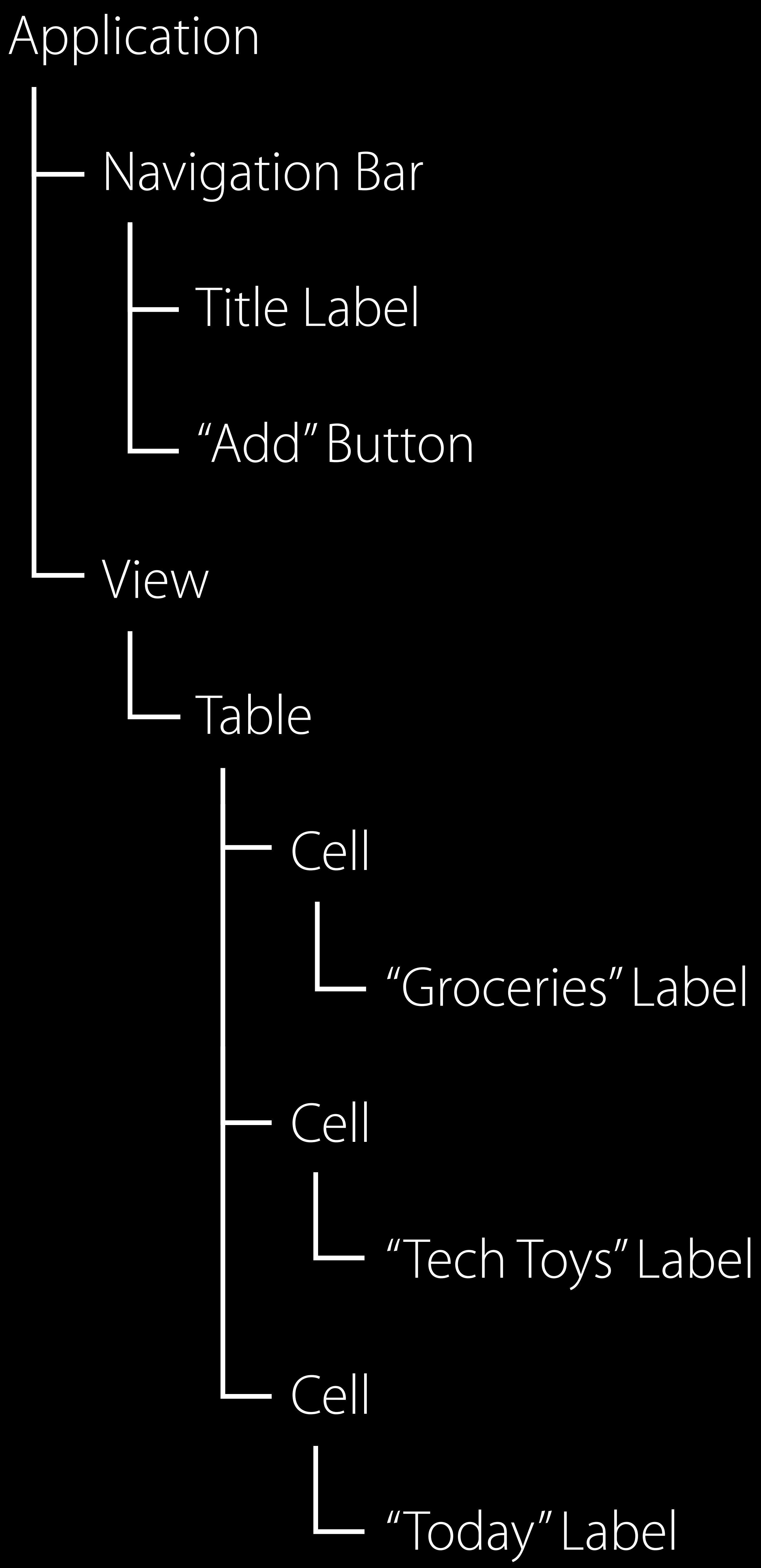
Expressing relationships

Descendants

Children

Containment





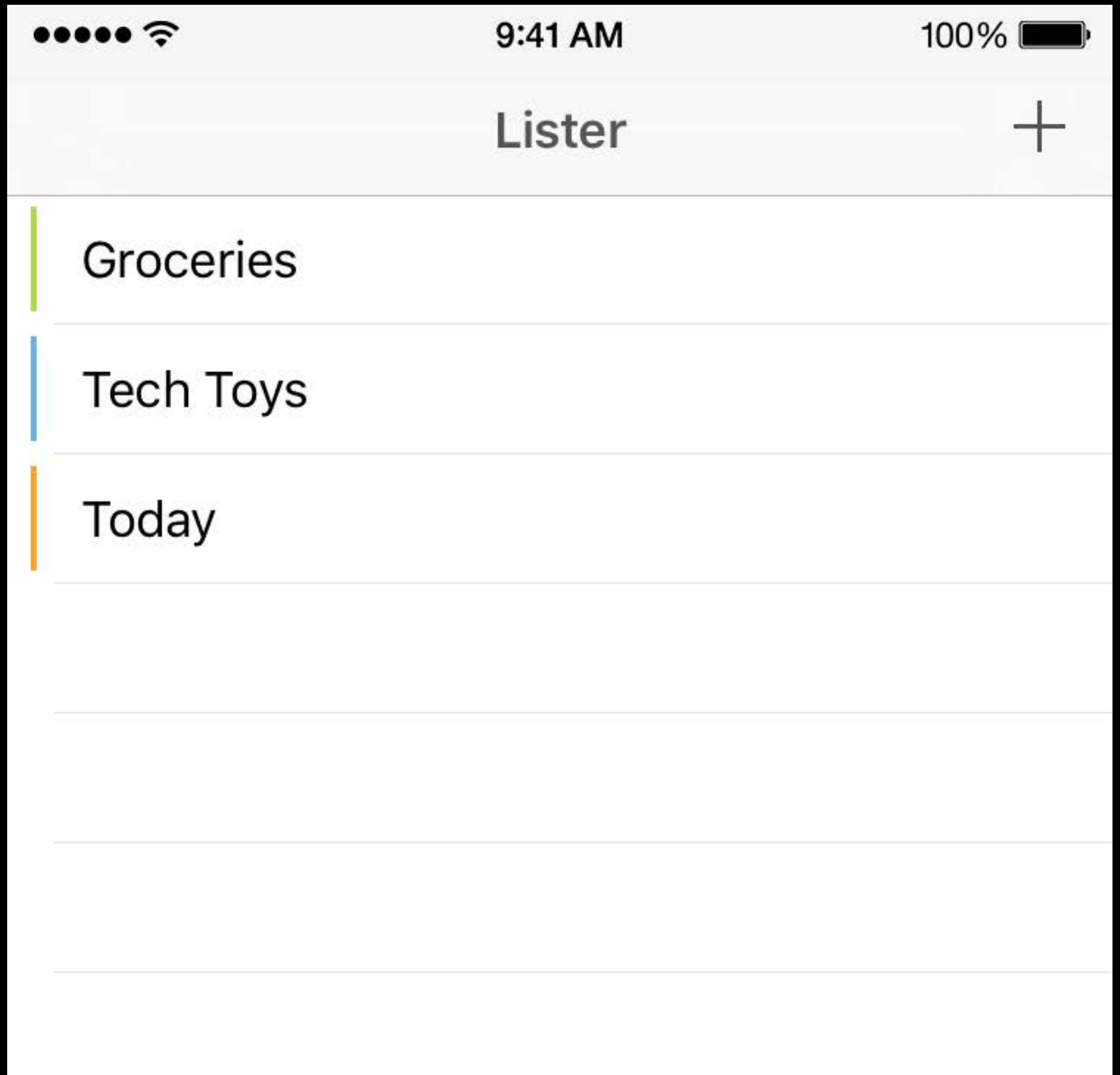
XCUIElementQuery

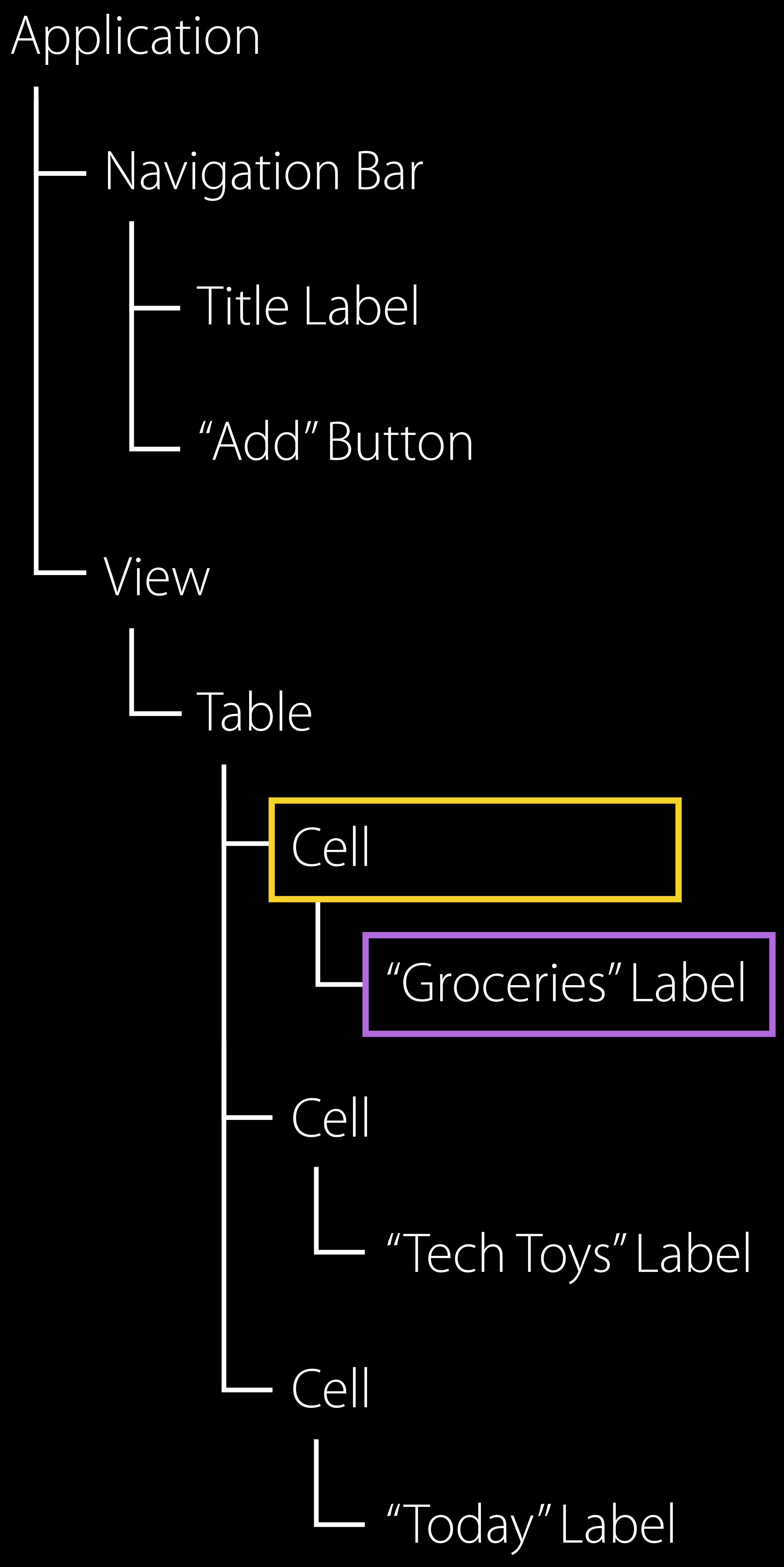
Expressing relationships

Descendants

Children

Containment





XCUIElementQuery

Filtering

XCUIElementQuery

Filtering

Element type

- Button, table, menu, etc.

Identifiers

- Accessibility identifier, label, title, etc.

Predicates

- Value, partial matching, etc.

Combining Relationships and Filtering

descendantsMatchingType()

Combining Relationships and Filtering

descendantsMatchingType()

So common, we provide convenience API for each type

```
let allButtons = app.buttons

let allCellsInTable = table.cells

let allMenuItemsInMenu = menu.descendantsMatchingType(.MenuItem)
```

Combining Relationships and Filtering

childrenMatchingType()

Combining Relationships and Filtering

childrenMatchingType()

Differentiates between any descendant and a direct child relationship

区分后代与直系孩子的关系

Combining Relationships and Filtering

childrenMatchingType()

Differentiates between any descendant and a direct child relationship

```
let allButtons = app.buttons // descendantsMatchingType(.Button)
```

Combining Relationships and Filtering

childrenMatchingType()

Differentiates between any descendant and a direct child relationship

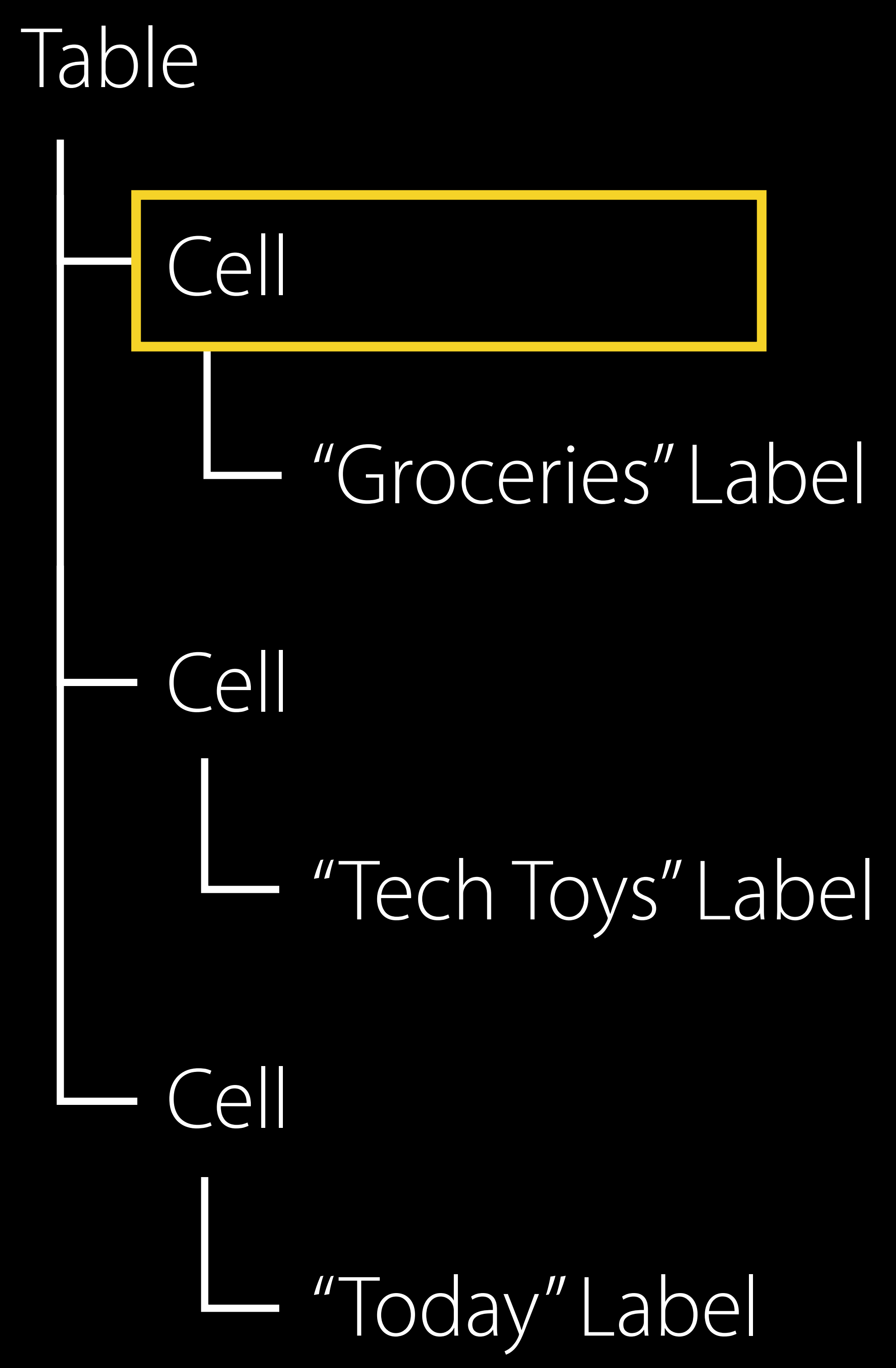
```
let allButtons = app.buttons // descendantsMatchingType(.Button)

let childButtons = navBar.childrenMatchingType(.Button)
```

Combining Relationships and Filtering

containingType()

Find elements by describing their descendants

通过后代的表述找到元素

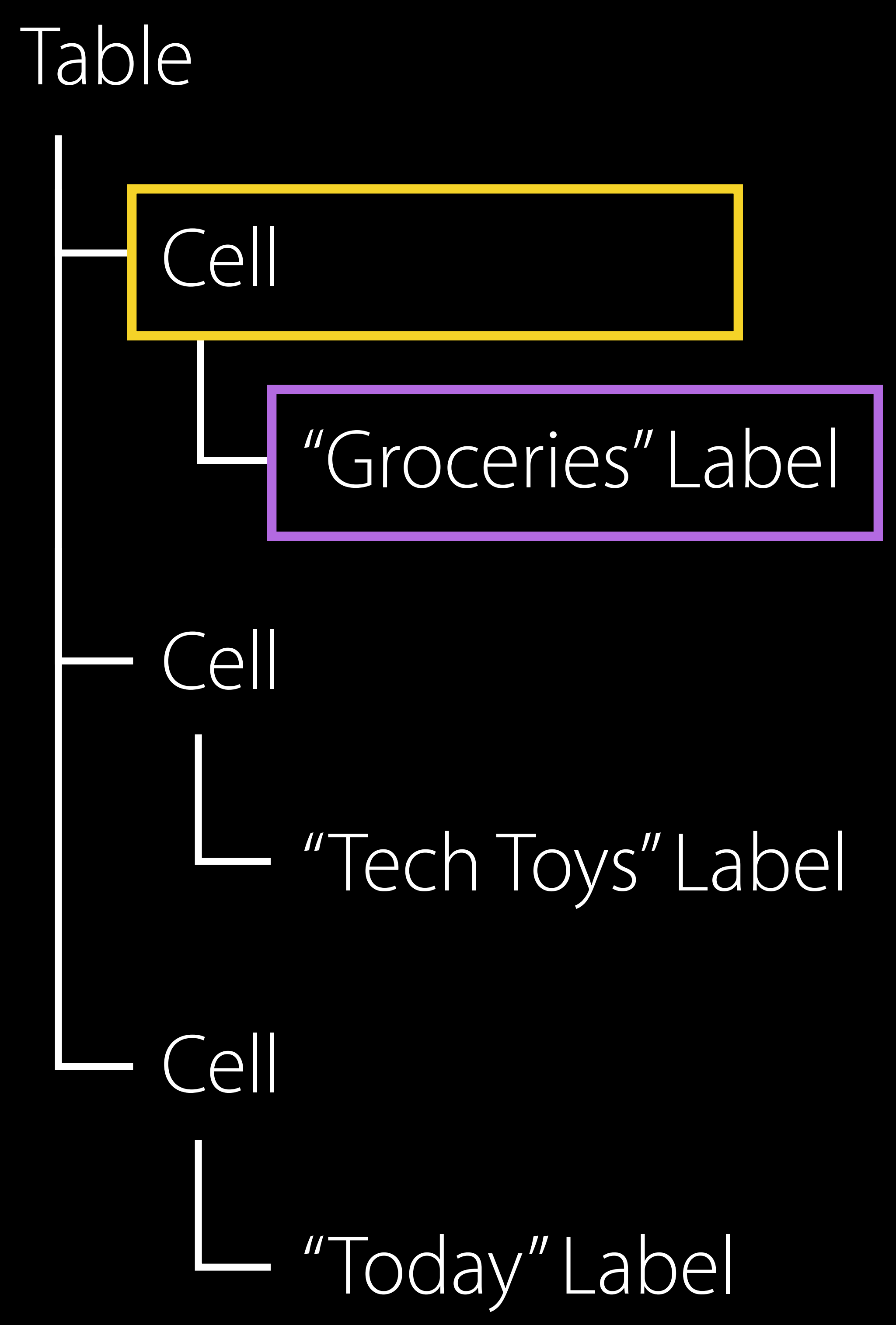


Combining Relationships and Filtering

containingType()

Find elements by describing their descendants

```
let cellQuery = cells.containingType(.StaticText,
                                     identifier:"Groceries")
```



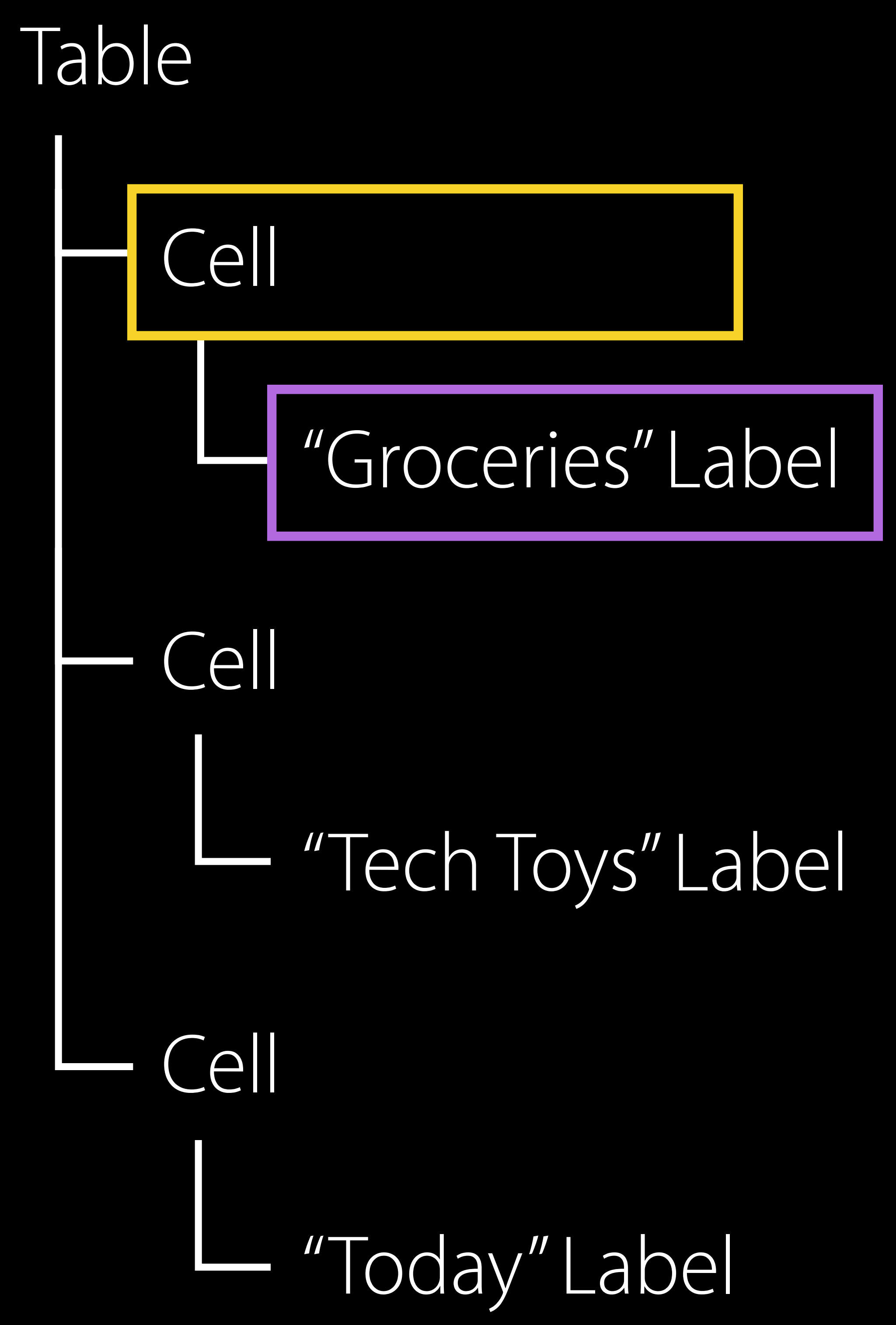
Combining Relationships and Filtering

containingType()

Find elements by describing their descendants

```
let cellQuery = cells.containingType(.StaticText,
                                     identifier:"Groceries")
```

Predicate variant also available



XCUIElementQuery

Combining relationships and filtering

联合关系与过滤

```
descendantsMatchingType()
childrenMatchingType()
containingType()
```

Combining Queries

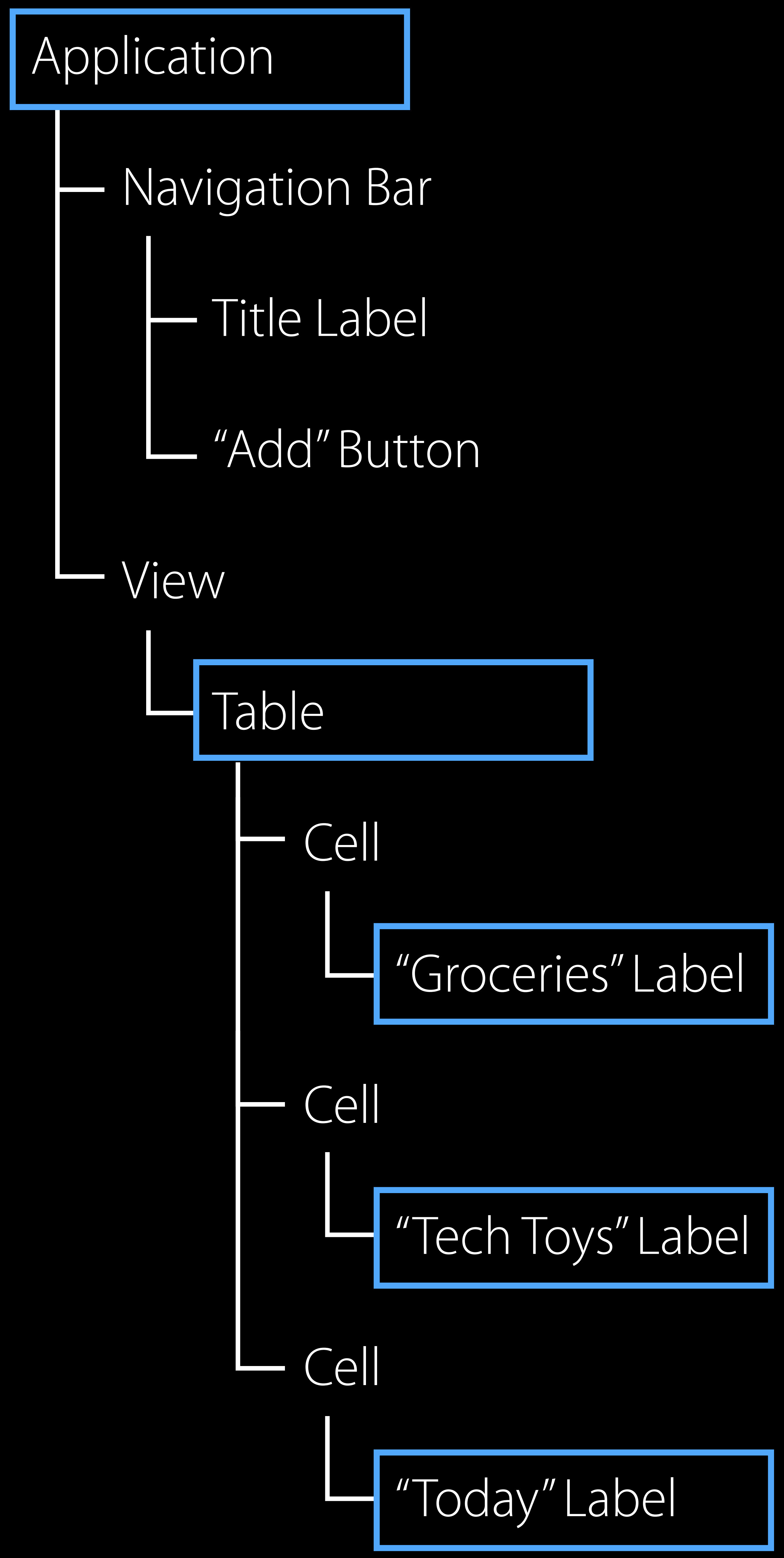
Queries can be “chained” together 询问语句支持链式编程

Output of each query is the input of the next query

输出的询问是下一个询问的输入

```
let labelsInTable = app.tables.staticTexts
```

如何简单设计一个链式编程



Getting Elements from Queries

Subscripting	`table.staticTexts[“Groceries”]`
Index	`table.staticTexts.elementAtIndex(0)`
Unique	`app.navigationBars.element`

Evaluating Queries

Queries are evaluated on demand

XCUIElement

- Synthesizing events
- Reading property values

XCUIElementQuery

- Getting number of matches (`.count`)
- Getting all matches (`.allElementsBoundByAccessibilityElement`)

Re-evaluated when UI changes

Queries and Elements

Similar to URLs

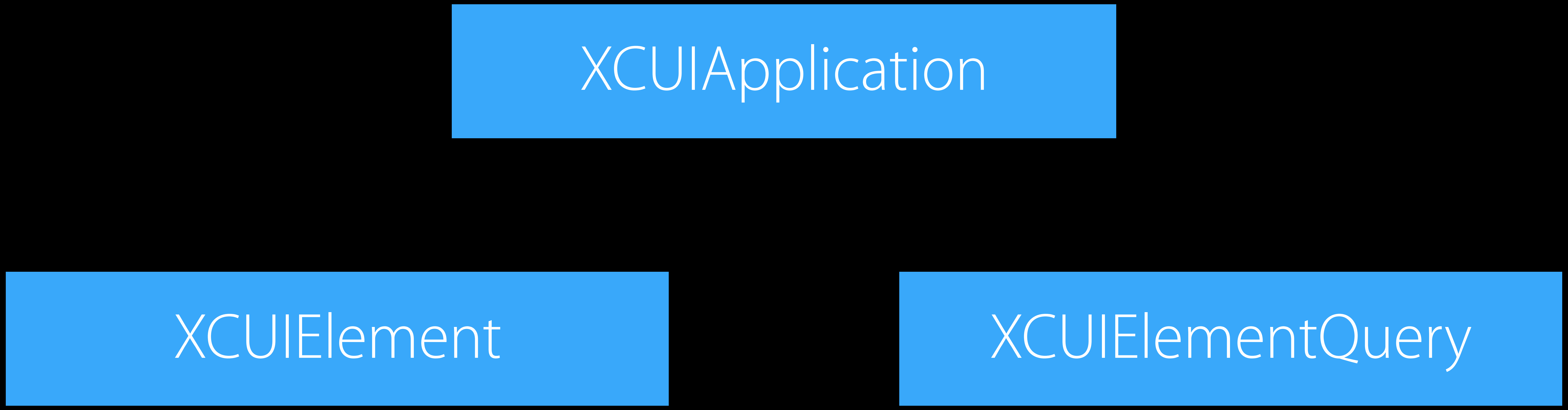
Creating a URL does not fetch a resource

- URL could be invalid, error raised when requested

Queries and elements

- Just a specification for accessible elements in the tested application
- Not resolved until needed

API Recap



Accessibility and UI Testing

可理解、可进入 与 UI测试

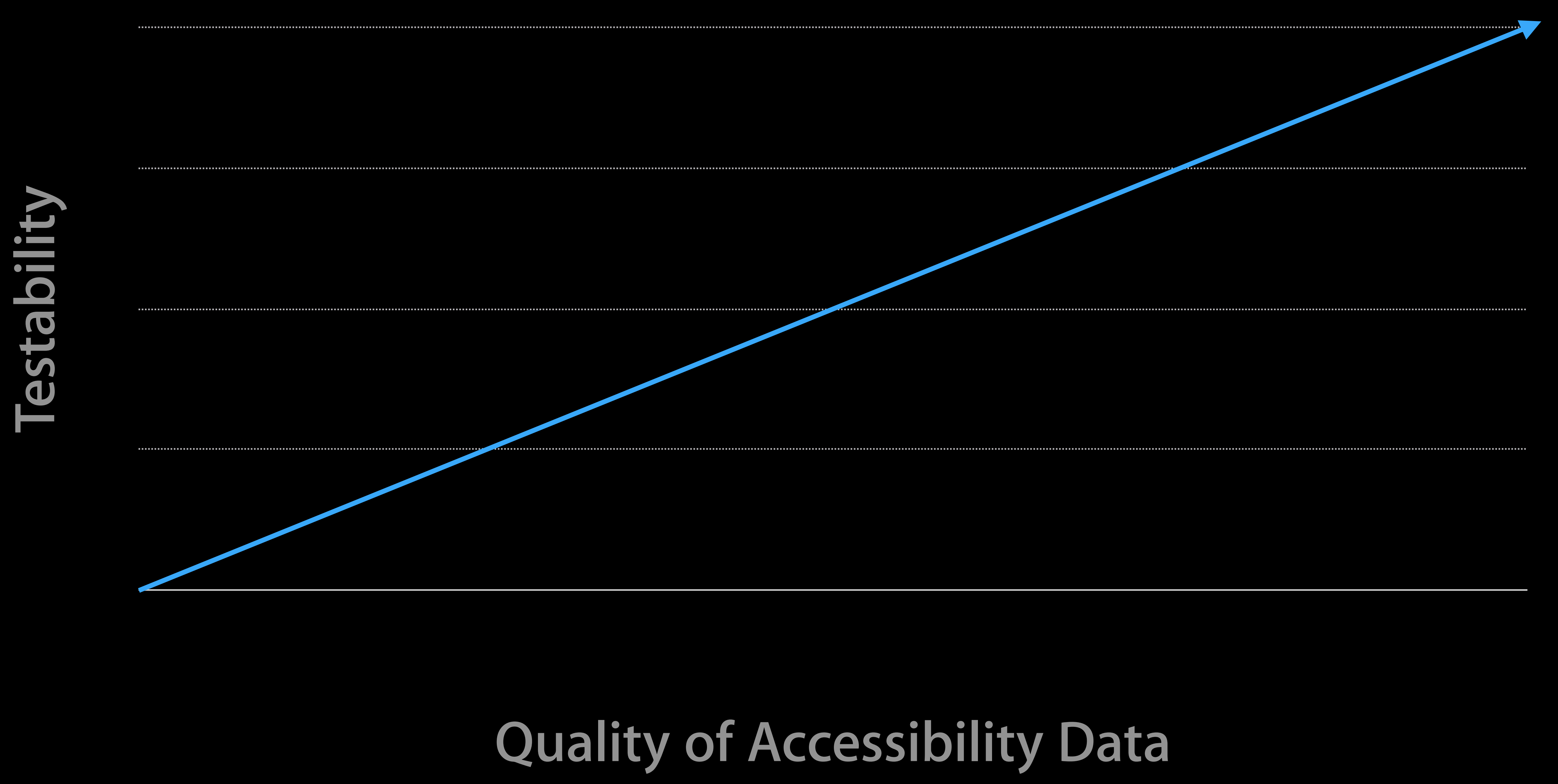
Accessibility and UI Testing

UI是什么？

如何表示UI视图？

Accessibility data makes UI testing possible

数据可理解让UI测试成为可能



Accessibility and UI Testing

Debugging tips

Not accessible

- Custom view subclasses
- Layers, sprites, and other graphics objects

Poor accessibility data

Tools

- UI recording
- Accessibility inspectors

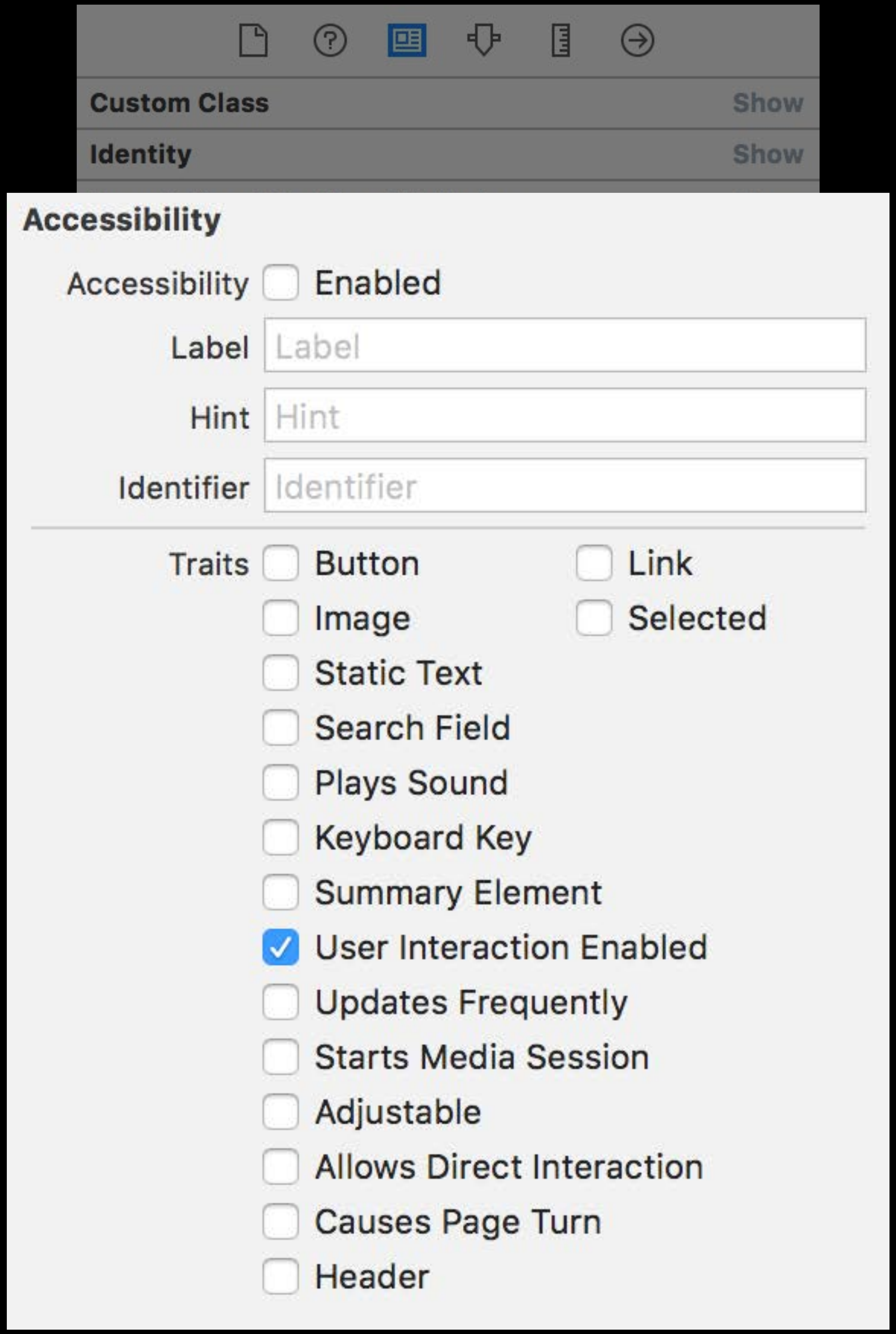
Accessibility and UI Testing

Improving data

Interface Builder inspector

API

- UIAccessibility (iOS)
- NSAccessibility (OS X)



Custom Class Show

Identity Show

Accessibility

Accessibility	[ ] Enabled
Label	Label
Hint	Hint
Identifier	Identifier

Traits:
- [ ] Button
- [ ] Link
- [ ] Image
- [ ] Selected
- [ ] Static Text
- [ ] Search Field
- [ ] Plays Sound
- [ ] Keyboard Key
- [ ] Summary Element
- [✓] User Interaction Enabled
- [ ] Updates Frequently
- [ ] Starts Media Session
- [ ] Adjustable
- [ ] Allows Direct Interaction
- [ ] Causes Page Turn
- [ ] Header

Test Reports

UI Refresh

Test Reports

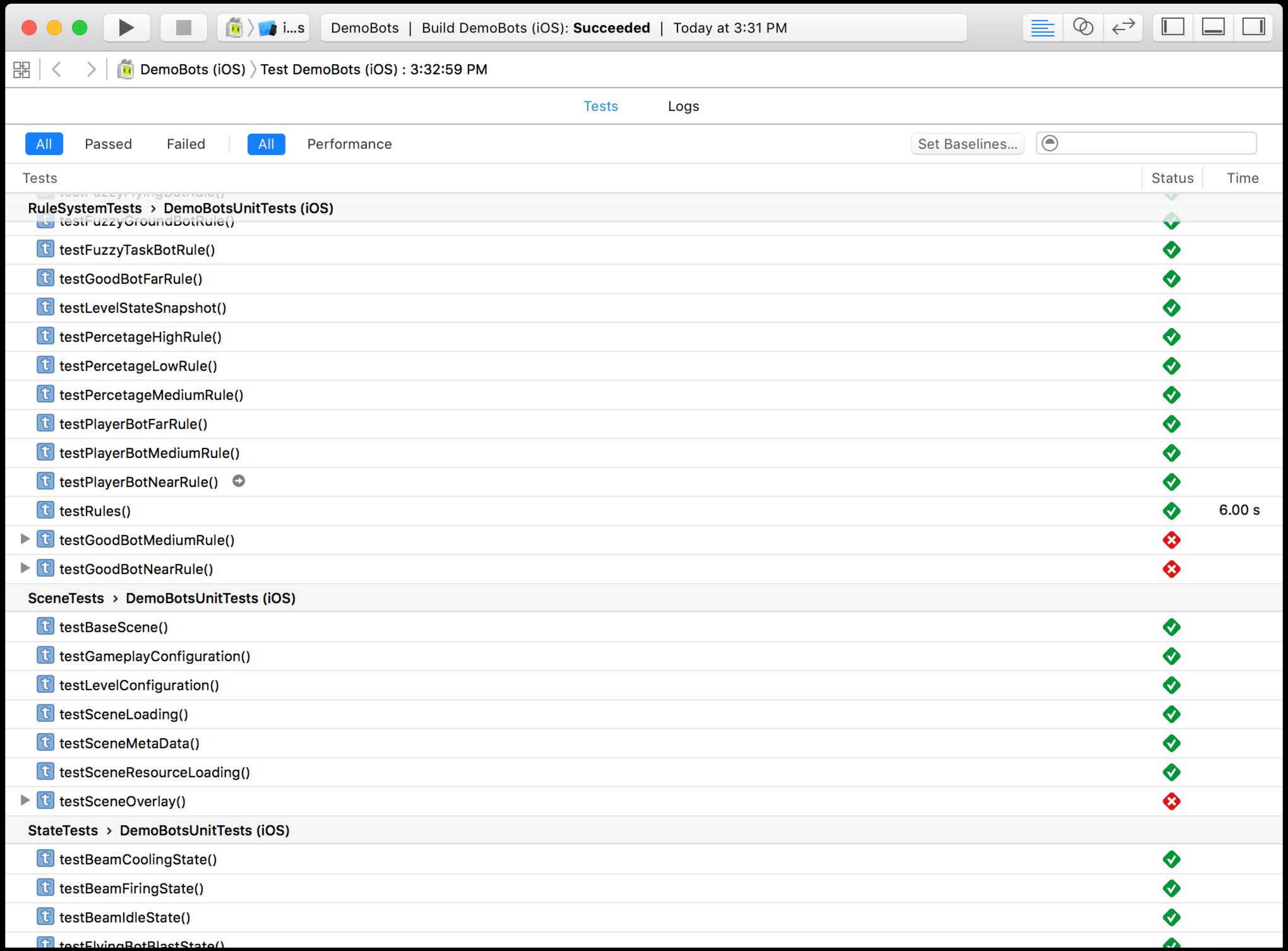
展示所有的测试结果

Show results for all tests

- Pass/fail
- Failure reason
- Performance metrics 性能计量

Same UI in Xcode and in Xcode Server

Per-device results for Xcode Server



DemoBots | Build DemoBots (iOS): **Succeeded** | Today at 3:31 PM

DemoBots (iOS) 〉 Test DemoBots (iOS) : 3:32:59 PM

Tests	Status	Time
testFuzzyFlyingBotRule()	✓	
RuleSystemTests › DemoBotsUnitTests (iOS)		
testFuzzyGroundBotRule()	✓	
testFuzzyTaskBotRule()	✓	
testGoodBotFarRule()	✓	
testLevelStateSnapshot()	✓	
testPercetageHighRule()	✓	
testPercetageLowRule()	✓	
testPercetageMediumRule()	✓	
testPlayerBotFarRule()	✓	
testPlayerBotMediumRule()	✓	
testPlayerBotNearRule()	✓	
testRules()	✓	6.00 s
testGoodBotMediumRule()	✗	
testGoodBotNearRule()	✗	
SceneTests › DemoBotsUnitTests (iOS)		
testBaseScene()	✓	
testGameplayConfiguration()	✓	
testLevelConfiguration()	✓	
testSceneLoading()	✓	
testSceneMetaData()	✓	
testSceneResourceLoading()	✓	
testSceneOverlay()	✗	
StateTests › DemoBotsUnitTests (iOS)		
testBeamCoolingState()	✓	
testBeamFiringState()	✓	
testBeamIdleState()	✓	
testFlyingBotBlastState()	✓	

Test Reports 测试报告

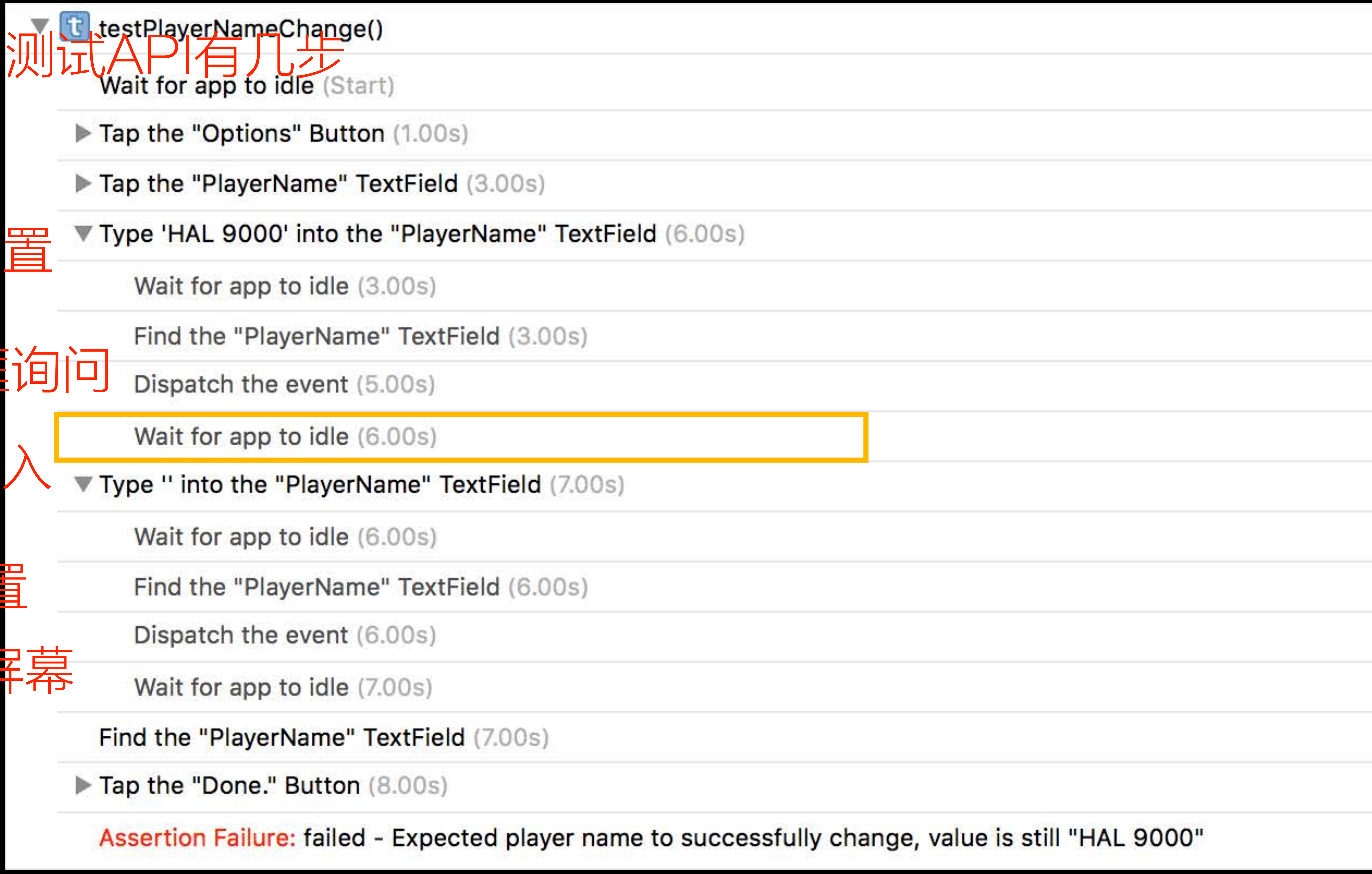
Nested activities

UI testing APIs have several steps UI测试API有几步

Typing into a textfield 输入文本框

- Wait for the app to idle 等待应用程序闲置
- Evaluate the textfield query 估计文本框询问
- Synthesize the text input 合成文本输入
- Wait for the app to idle 等待App闲置

QuickLook for screenshots 快速浏览屏幕



```
▼ testPlayerNameChange()
    Wait for app to idle (Start)
  ▶ Tap the "Options" Button (1.00s)
  ▶ Tap the "PlayerName" TextField (3.00s)
  ▼ Type 'HAL 9000' into the "PlayerName" TextField (6.00s)
      Wait for app to idle (3.00s)
      Find the "PlayerName" TextField (3.00s)
      Dispatch the event (5.00s)
      Wait for app to idle (6.00s)
  ▼ Type '' into the "PlayerName" TextField (7.00s)
      Wait for app to idle (6.00s)
      Find the "PlayerName" TextField (6.00s)
      Dispatch the event (6.00s)
      Wait for app to idle (7.00s)
    Find the "PlayerName" TextField (7.00s)
  ▶ Tap the "Done." Button (8.00s)
    Assertion Failure: failed - Expected player name to successfully change, value is still "HAL 9000"
```

When to Use UI Testing

什么时候使用UI测试

Using UI Testing

Complements unit testing 补充单元测试

Unit testing more precisely pinpoints failures 单元测试更精确地确定了失败

UI testing covers broader aspects of functionality UI测试覆盖了函数边界方面

Find the right blend of UI tests and unit tests for your project
找到好的方式融合UI测试和单元测试

Candidates for UI Testing

使用UI测试的情况

Demo sequences 一些列Demo

Common workflows 相同的工作流程

Custom views 相同的视图

Document creation, saving, and opening 文档的创建、保存、打开

Summary 总结

UI testing

- Find and interact with UI elements 找到UI交互元素
- Validate UI properties and state 让UI的属性和状态生效

UI recording UI记录

Test reports 测试报告

More Information

Testing in Xcode Documentation
http://developer.apple.com/testing

Accessibility for Developers Documentation
http://developer.apple.com/accessibility

Apple Developer Forums
http://developer.apple.com/forums

Stefan Lesser
Developer Tools Evangelist
slesser@apple.com

Related Sessions

iOS Accessibility	Pacific Heights	Tuesday 9:00 AM
Continuous Integration and Code Coverage in Xcode	Presidio	Thursday 10:00 AM

Labs

Testing and Continuous Integration	Developer Tools Lab B	Wednesday 1:30 PM
Testing and Continuous Integration	Developer Tools Lab B	Thursday 1:30 PM

WWDC15