

Детальная инструкция интеграции ViewInspector в Xcode

 **ВАЖНО:** Эта инструкция требует ручных действий в Xcode UI

 Предварительные требования

- Xcode 15.0 или выше
 - macOS Sonoma или выше
 - Проект LMS должен компилироваться без ошибок
 - ~15 минут времени
-

Шаг 1: Открытие проекта в Xcode

1.1. Закройте все окна Xcode

```
# В терминале проверьте, что Xcode не запущен
ps aux | grep Xcode
# Если есть процессы Xcode, закройте их
```




1.2. Перейдите в директорию проекта

```
cd /Users/ishirokov/lms_docs/LMS_App/LMS
```

1.3. Откройте проект

```
open LMS.xcodeproj
```

1.4. Дождитесь полной загрузки проекта

-  Индексация файлов завершена (прогресс-бар вверху исчез)
 -  В навигаторе слева видны все файлы проекта
 -  Нет красных ошибок в навигаторе
-

Шаг 2: Добавление ViewInspector через Package Dependencies

2.1. Откройте настройки проекта

1. В левой панели (Navigator) кликните на **корневой элемент "LMS"** (синяя иконка)
2. Убедитесь, что выбран **PROJECT "LMS"** (не TARGET)

3. Перейдите на вкладку "**Package Dependencies**"

2.2. Добавьте новый пакет

1. Нажмите кнопку "+" под списком пакетов
2. В появившемся окне "Add Package" введите URL:

```
https://github.com/nalexn/ViewInspector
```

3. Нажмите "**Add Package**" (правый нижний угол)

2.3. Настройте версию пакета

1. В следующем окне выберите правило версии:
 - **Dependency Rule:** Up to Next Major Version
 - **Version:** 0.9.8
2. Нажмите "**Add Package**"

2.4. Выберите таргеты для пакета

⚠ **КРИТИЧЕСКИ ВАЖНО:** Правильный выбор таргетов!

1. В окне "Choose Package Products" вы увидите:
 - Product: **ViewInspector**
 - Targets: список доступных таргетов
2. **ОБЯЗАТЕЛЬНО** поставьте галочку только напротив:
 - ☒ **LMSTests**
3. **НЕ СТАВЬТЕ** галочки напротив:
 - ☒ **LMS**
 - ☒ **LMSUITests**
4. Нажмите "**Add Package**"

2.5. Проверьте успешное добавление

1. В списке Package Dependencies должен появиться:
 - ViewInspector 0.9.8
2. В навигаторе слева внизу появится секция "Package Dependencies"
3. Раскройте её и убедитесь, что там есть ViewInspector

Шаг 3: Включение отключенных тестов

3.1. Найдите отключенные тесты

В терминале выполните:

```
# Покажет все отключенные файлы тестов
find LMSTests -name "*.swift.disabled" -type f
```

Вы должны увидеть:

```
LMSTests/Views/LoginViewInspectorTests.swift.disabled
LMSTests/Views/ContentViewInspectorTests.swift.disabled
LMSTests/Views/SettingsViewInspectorTests.swift.disabled
LMSTests/Views/ProfileViewInspectorTests.swift.disabled
LMSTests/Views/CourseListViewInspectorTests.swift.disabled
LMSTests/Helpers/ViewInspectorHelper.swift.disabled
```

3.2. Переименуйте файлы обратно

```
# Включить все тесты одной командой
for file in LMSTests/**/*.swift.disabled; do
    mv "$file" "${file%.disabled}"
done
```

3.3. Проверьте результат

```
# Должно показать 0 файлов
find LMSTests -name "*.swift.disabled" -type f | wc -l
```

3.4. Обновите Xcode

1. Вернитесь в Xcode
2. В меню выберите **File → Packages → Reset Package Caches**
3. Подождите пока Xcode обновит индексы

3.5. Добавьте файлы в таргет LMSTests (если нужно)

Если файлы не появились автоматически:

1. В навигаторе найдите папку **LMSTests/Views**
2. Правый клик → **Add Files to "LMS"...**
3. Выберите все файлы ViewInspector тестов
4. **ВАЖНО:** В диалоге добавления:
 - ☒ Copy items if needed: **снять галочку**
 - ☒ Add to targets: **только LMSTests**
5. Нажмите **Add**

Шаг 4: Проверка компиляции тестов

4.1. Соберите тесты

1. Выберите схему **LMS** (вверху рядом с кнопкой Run)
2. Выберите симулятор **iPhone 16 Pro**
3. Нажмите **Cmd+Shift+U** (или Product → Build For → Testing)

4.2. Исправьте ошибки импорта (если есть)

Если появились ошибки "No such module 'ViewInspector'":

1. Откройте любой файл с ошибкой
2. В начале файла должно быть:

```
import XCTest
import SwiftUI
import ViewInspector
@testable import LMS
```

3. Если ViewInspector подчеркнут красным:
 - Cmd+Shift+K (Clean Build Folder)
 - Cmd+B (Build)

Шаг 5: Запуск тестов с измерением покрытия

5.1. Включите сбор покрытия кода

1. В Xcode выберите схему **LMS** → **Edit Scheme...**
2. Слева выберите **Test**
3. Перейдите на вкладку **Options**
4. ☒ Поставьте галочку **Code Coverage**
5. ☒ Gather coverage for: **All Targets** (или выберите specific targets)
6. Нажмите **Close**

5.2. Запустите BCE тесты



```
# Вариант 1: Через Xcode UI
# Нажмите Cmd+U

# Вариант 2: Через терминал с детальным выводом
xcodebuild test \
  -scheme LMS \
  -destination 'platform=iOS Simulator,name=iPhone 16 Pro' \
  -enableCodeCoverage YES \
```

```
-resultBundlePath testResultsCoverage.xcresult \  
2>&1 | tee test_output_viewinspector_enabled.log
```

5.3. Следите за прогрессом

В Xcode вы увидите:

- Индикатор выполнения тестов в верхней части
- Результаты в Test Navigator (Cmd+6)
-  Зеленые галочки = тесты прошли
-  Красные крестики = тесты провалились

Шаг 6: Просмотр результатов покрытия

6.1. В Xcode

1. После завершения тестов откройте **Report Navigator** (Cmd+9)
2. Выберите последний отчет о тестировании
3. Перейдите на вкладку **Coverage**
4. Вы увидите процент покрытия для каждого файла

6.2. Через xcresult

```
# Экспорт отчета о покрытии  
xcrun xccov view --report testResultsCoverage.xcresult >  
coverage_report.txt  
  
# Просмотр общего покрытия  
xcrun xccov view --report --only-targets testResultsCoverage.xcresult
```

6.3. Создание HTML отчета

```
# Установите xcov если нет  
gem install xcov  
  
# Сгенерируйте HTML отчет  
xcov -p LMS.xcodeproj -s LMS -o coverage_html
```

Возможные проблемы и решения

Проблема 1: ViewInspector не видит SwiftUI Views

Решение: Убедитесь, что в Build Settings таргета LMS:

- **Enable Testability** = YES

- **Build Active Architecture Only** (Debug) = NO

Проблема 2: Тесты падают с "View not found"

Решение: ViewInspector требует специальной структуры View. Проверьте:

```
// Правильно
try sut.inspect().find(button: "Login")

// Неправильно
try sut.inspect().button("Login")
```

Проблема 3: Медленная компиляция

Решение:

1. Закройте другие приложения
2. В Xcode: Product → Clean Build Folder (Cmd+Shift+K)
3. Удалите DerivedData:

```
rm -rf ~/Library/Developer/Xcode/DerivedData/LMS-*
```

Проблема 4: "Missing package product 'ViewInspector'"

Решение:

1. File → Packages → Reset Package Caches
2. File → Packages → Update to Latest Package Versions
3. Перезапустите Xcode

✅ Проверка успешной интеграции

После выполнения всех шагов у вас должно быть:

1. **В Package Dependencies:** ViewInspector 0.9.8
2. **Количество тестов:** 1,159+ (было 1,051, добавилось 108)
3. **Все тесты проходят:** Зеленый статус
4. **Покрытие кода увеличилось:** С 7.2% до ~10-12%

Ожидаемые результаты

- **Время выполнения всех тестов:** 3-5 минут
 - **Новое покрытие кода:** 10-12%
 - **ViewInspector тесты:** 108 активных
 - **Общее количество тестов:** ~1,159
-

Следующие шаги

После успешной интеграции:

1. Закоммитьте изменения:

```
git add .  
git commit -m "feat: Enable ViewInspector UI tests (+108 tests)"
```

2. Создайте отчет о покрытии:

```
./generate-coverage-report.sh
```

3. Обновите Sprint 35 документацию с новыми метриками

Последнее обновление: 8 июля 2025

Время выполнения: ~15 минут