

Локализация приложения



Алексей
Жембловский



Алексей Жембловский

Senior iOS developer в EPAM



План занятия

1. [Работа со Storyboard файлами](#)
2. [Ключи в Info.plist](#)
3. [Локализация в коде](#)
4. [Работа с множественными формами](#)
5. [Форматирование дат](#)
6. [Локализация единиц измерений](#)



Работа со Storyboard файлами

Локализация Storyboard

Локализация приложения автоматически меняется при смене настроек на устройстве.

Для локализации Storyboard, необходимо выполнить ряд простых настроек в проекте. В результате у нас появляется еще один файл с расширением .string (RUSSIAN) и в названии указывается язык.

Этот файл содержит тип объекта, ключ объекта и текст, который нужно локализовать.

```
/* Class = "UILabel"; text = "Hellow"; ObjectID =  
"dZr-le-4Qg"; */"dZr-le-4Qg.text" = "Привет";
```

```
/* Class = "UIButton"; normalTitle = "Localized"; ObjectID =  
"eC3-00-2VE"; */"eC3-00-2VE.normalTitle" = "Локализация";
```

Локализация Storyboard

```
* Class = "UILabel"; text = "Hellow"; ObjectID = "dZr-le-4Qg";  
*/ "dZr-le-4Qg.text" = "Привет";
```

dZr-le-4Qg – это ключ объекта. Он продублирован еще раз перед переводом.

Ключ можно посмотреть в storyboard, кликнув на объекте.

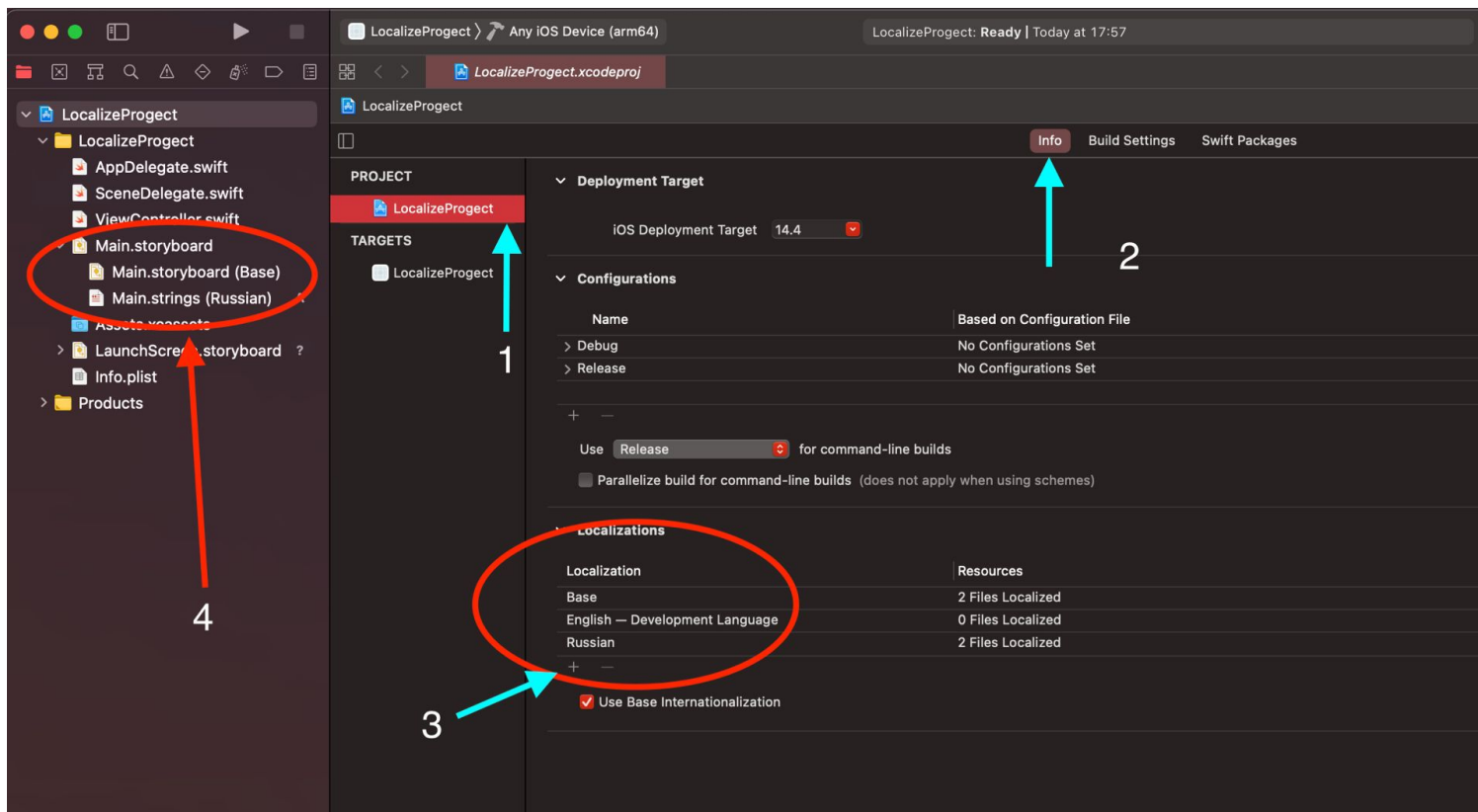
Если Вы создали проект и потом приступаете к локализации, то файл с расширением .string(Russian) может быть пустым. Чтобы начать локализацию достаточно скопировать верхнюю строчку и поменять ключи (это пример для UILabel)

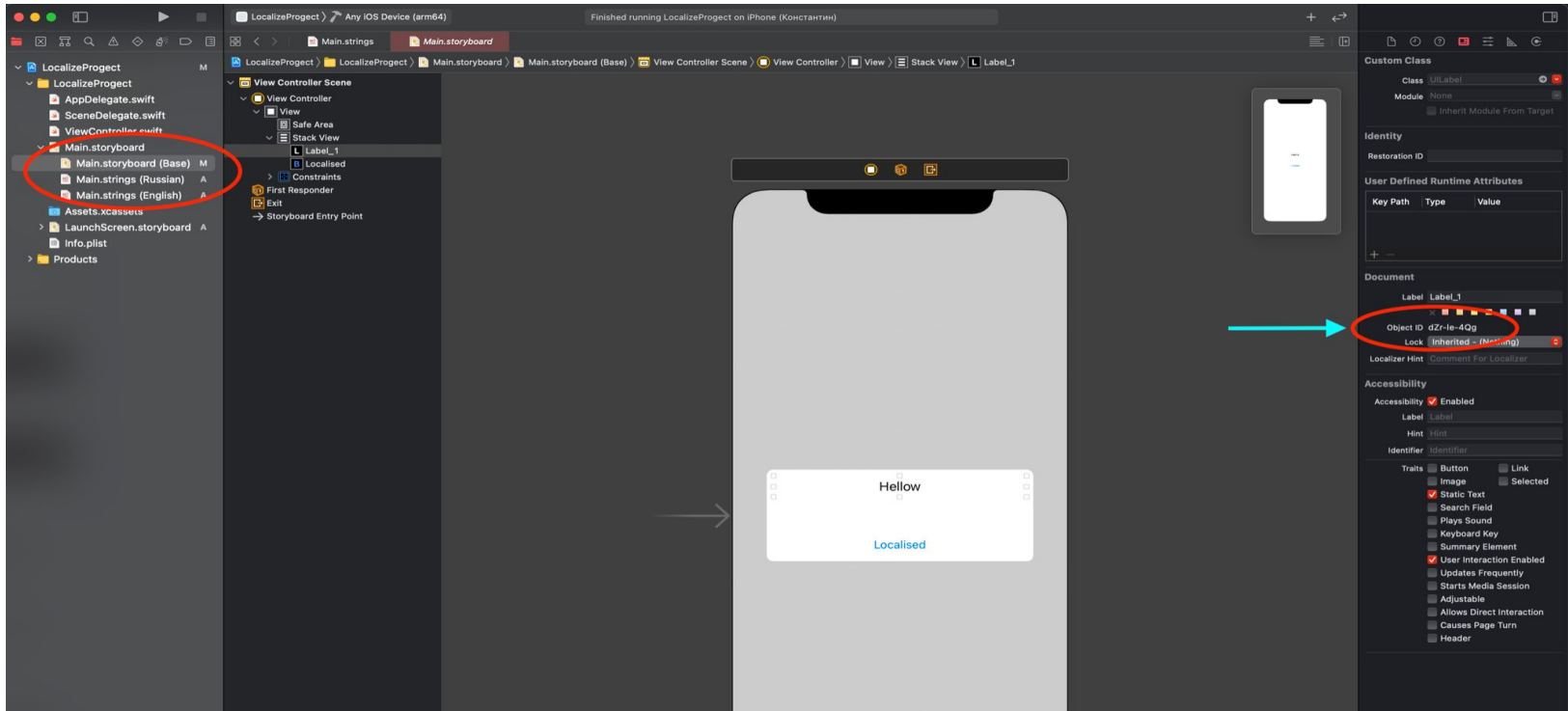
Так же надо поставить галочку рядом с языком локализации в диспетчере объекта. И появится второй файл с локализацией (English).

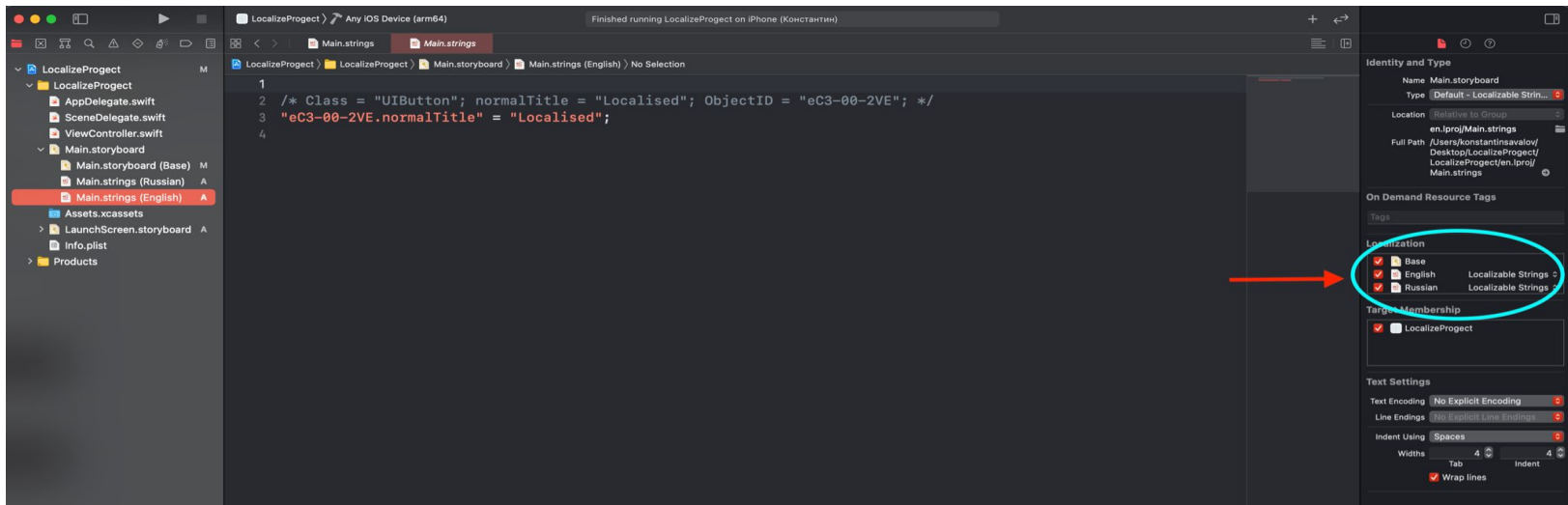
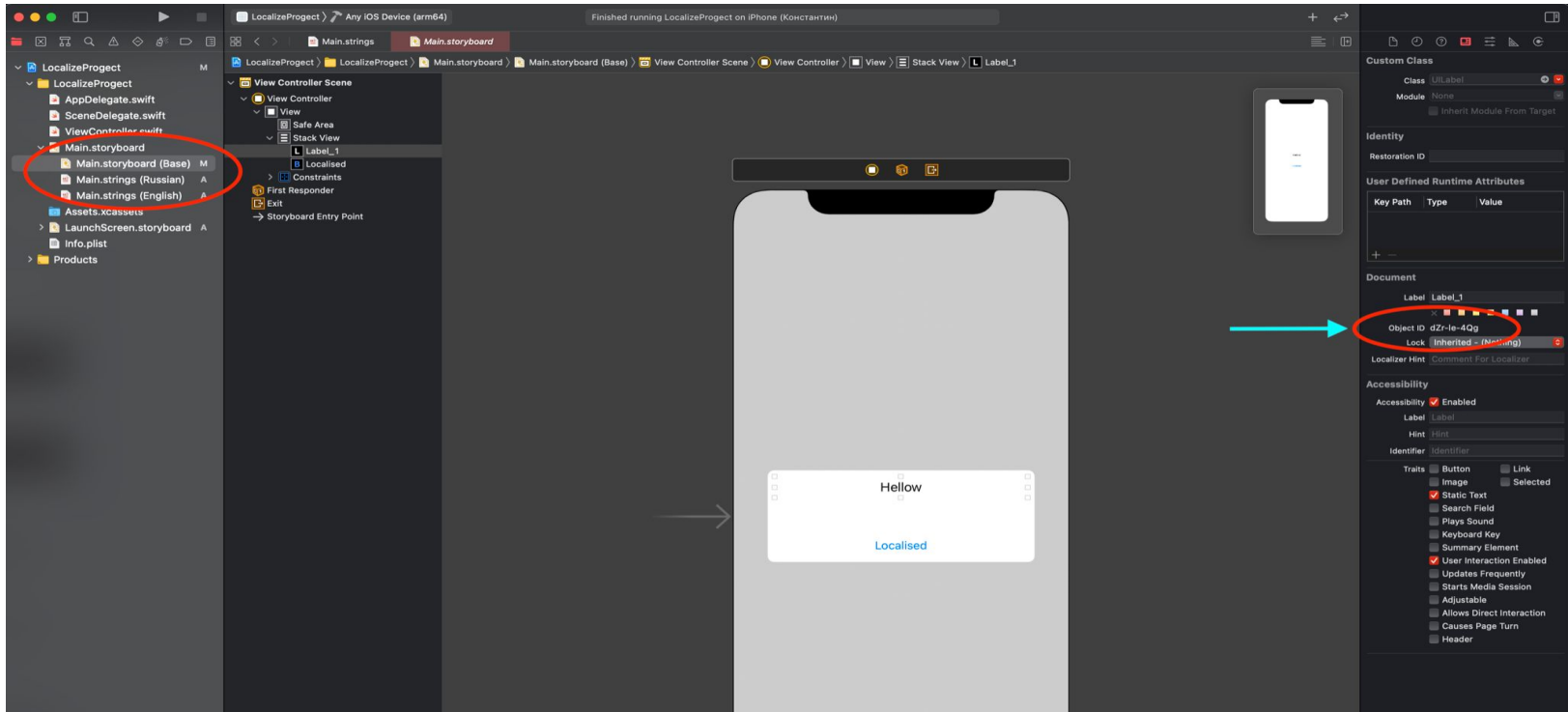
Далее при смене локализации в настройках устройства локализация будет меняться автоматически.

Последовательность настройки локализации:

1. Выбрать Project
2. Выбрать Info
3. Нажав на + выбрать язык
4. Появится файл с расширением .string







Локализация Info.plist

Если в проекте уже есть строки локализации, но вам необходимо добавить локализацию к некоторым ключам: `Info.plist`, например:

`NSCameraUsageDescription`, `NSLocationAlwaysUsageDescription` или `NSLocationUsageDescription`.

1. Создать `InfoPlist.string` (вы создаете обычный текстовый файл проекта, но название должно быть именно таким).
2. Добавить необходимый ключ в `InfoPlist.string`

```
"NSCameraUsageDescription" = "Allow camera access to scan QR code";
```

В данном случае мы локализуем алерт, который появляется при попытке доступа к камере.

Ключи в Info.plist

3. Добавить в InfoPlist.string(Russia) в ключе “CameraUsageDescription” фразу на русском.



Information Property List	Dictionary	(18 items)
Localization native development region	String	\$(DEVELOPMENT_LANGUAGE)
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	\$(PRODUCT_BUNDLE_PACKAGE_TYPE)
Bundle version string (short)	String	\$(MARKETING_VERSION)
Bundle version	String	\$(CURRENT_PROJECT_VERSION)
Application requires iPhone environment	Boolean	YES
> App Transport Security Settings	Dictionary	(1 item)
Privacy - Camera Usage Description	String	Разрешить доступ к камере, чтобы отсканировать QR-код
> Application Scene Manifest	Dictionary	(1 item)
> Required background modes	Array	(1 item)
Launch screen interface file base name	String	Main
Main storyboard file base name	String	Main
> Required device capabilities	Array	(1 item)
> Supported interface orientations	Array	(1 item)
> Supported interface orientations (iPad)	Array	(4 items)

Локализация в коде

Для работы с локализованными текстовыми ресурсами фреймворк Foundation предоставляет семейство методов `NSLocalizedString` в Swift:

1. `NSLocalizedString(_ key: String, comment: String)`
2. `NSLocalizedString(_ key: String, tableName: String?, bundle: Bundle, value: String, comment: String)`

Локализация в коде

Вариант объявления `NSLocalizedString` имеет больше параметров:

- `key` – ключ, по которому искать перевод (локализацию)
- `comment` – краткое описание ключа, (например, “локализация поля “имя””)
- `tableName` – имя String-файла, в котором находится необходимая строка по указанному ключу.

Возможность навигации между таблицами позволяет не хранить строковые ресурсы в одном файле, а распределять их по собственному усмотрению. Это позволяет избавиться от перегруженности файлов, упрощает их редактирование, минимизирует шанс появления ошибок.

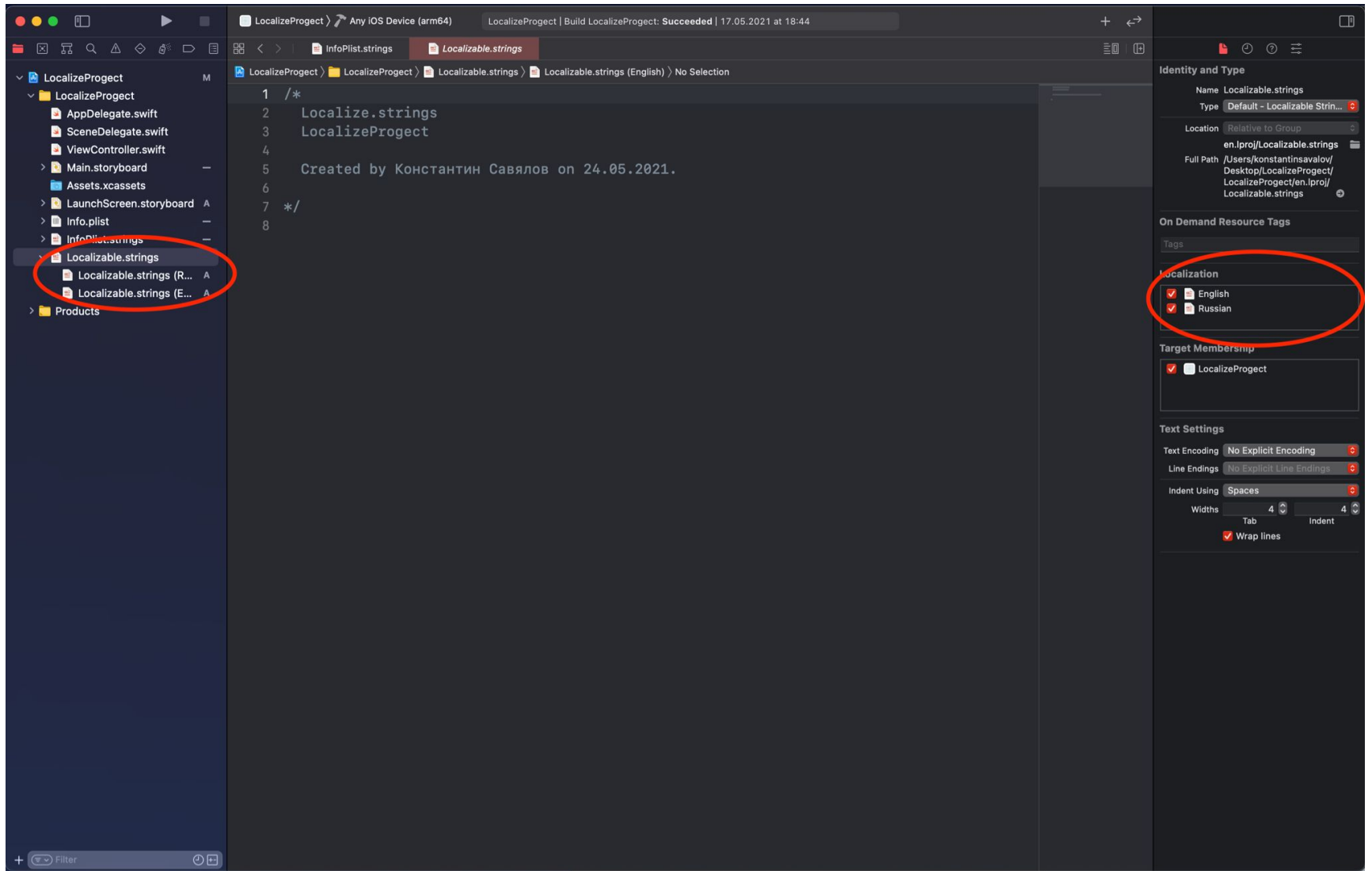
Локализация в коде

- bundle – это механизм доступа к ресурсам приложения, то есть мы можем самостоятельно определять источник ресурсов. Позволяет создавать локализацию в более широком смысле (проводить локализацию, используя ссылку на bundle).

В большинстве случаев достаточно использовать первый вариант.

Для описания локализации необходимо создать `.string` файл строго с именем Localizable.

В окне инспектора файла добавить галочку локализации для конкретного языка



Локализация единиц измерения

MeasurementFormatter – класс форматирования, обеспечивающий локализованное представление единиц и измерений.

Чтобы поменять локализацию расстояния, можно использовать *MeasurementFormatter* таким образом:

```
let dist = Measurement(value: 100, unit: UnitLength.kilometers)
```

UnitConverterLinear – класс преобразования единицы измерения с помощью линейного уравнения. Здесь мы получаем локализованное значение энергии (в данном случае в Джоули):

```
let energy = UnitEnergy(symbol: "prisedaniya", converter:
UnitConverterLinear(coefficient: 135.0))
let inPres = Measurement(value: 50, unit: energy)
```


Локализация множественных форм

Для локализации множественного количества чего либо, в отличии от простой локализации с созданием файла *Localizable.strings*, требуется по аналогии создать файл с названием *Localizable.stringsdict*.

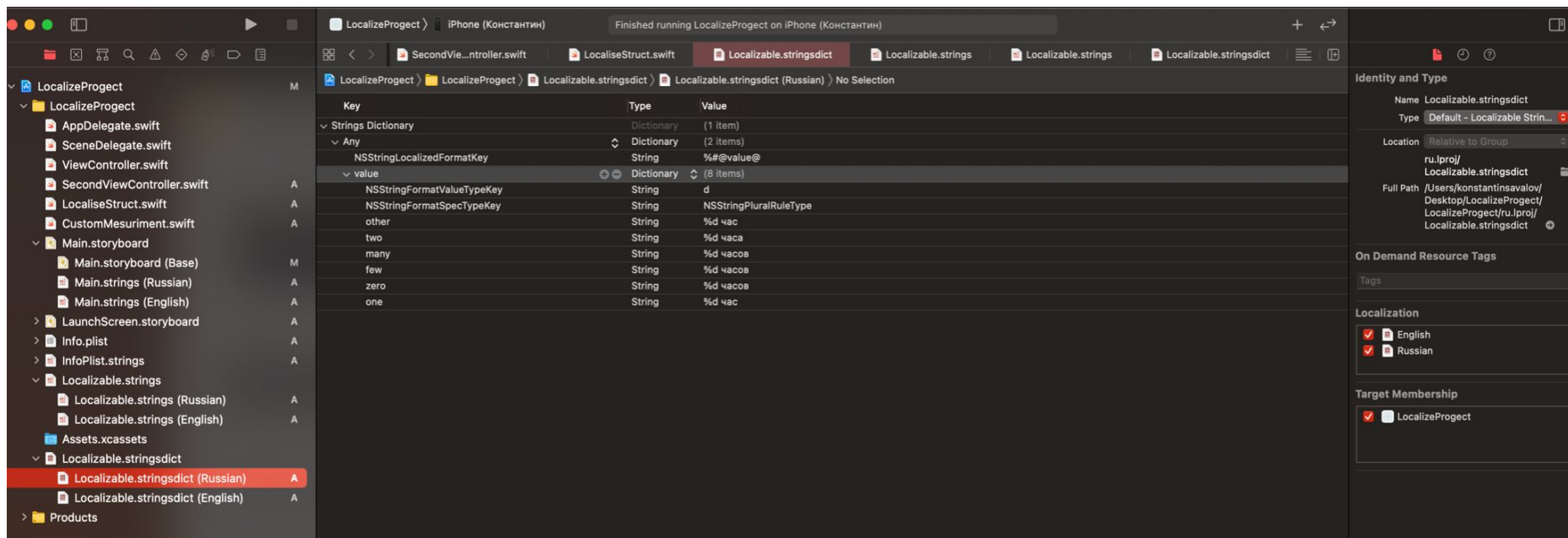
В отличие от `.strings`, файл `.stringsdict` представляет собой более сложную и функциональную структуру. Она применима не только для подмены входящей строки ее локализованным эквивалентом, но и способна учитывать параметры форматирования, включенные в строку и формировать результат в соответствии с выбранным правилом, в отношении параметра.



Локализация множественных форм

Plural Rule помогает использовать множественную форму слов для разных языков. Варианты для всех языков формализованы и работают согласно стандарту CLDR Language Plural Rules.

В нашем случае мы форматируем часы. То есть один час, пять часов, три часа.



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** учебной группы.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Алексей Жембловский