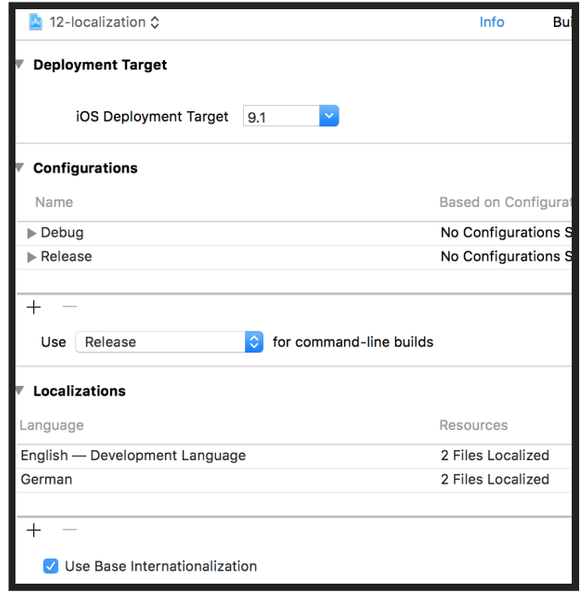
LOCALIZATION

LOCALIZATION SETUP I

- You can implement your app in different languages.
- Typically, language dependent strings are in your Storyboard and in your code
- In a first step, add your supported locations to your project settings (Click on project root and choose your project, not the target).

LOCALIZATION SETUP II

- Use base as your standard language (ie. English) and add all other languages (ie. German)
- You can choose to create a new storyboard for each language, typically a strings file will be sufficient.



LOCALIZATION OF STRINGS IN STORYBOARD

After you have setup the languages, you can make changes in your storyboard. If you have (language dependent) strings in your storyboard, do a trick to update all strings of your local .strings file:

1. Select your Base Storyboard file.
2. Check the Localization section and swap "Localizable Strings" to "Interface Builder Storyboard". Choose "convert".
3. Swap the "Interface Builder Storyboard" back to "Localizable Strings"
4. You should now find all newly created strings in your .strings file

LOCALIZATION OF STRINGS IN CODE

- Add a file called "Localizable.strings"
- Open Localizable.strings and check all languages that you want to support on the right side.

```
//you can define a key with a language dependent value in your Localizable.strings file
//for base
"MyFirstKey"="This is my first key";

//for German
"MyFirstKey"="Das ist mein erster Schlüssel";

//now you can load the key in your ViewController
label.text = NSLocalizedString("MyFirstKey", comment: "")

//By default, the language from your phone is used. If possible, try to retain this.
```

CHOSEN LANGUAGE

```
//you can get all installed languages
let languages : [String] = NSUserDefaults.standardUserDefaults().objectForKey("AppleLangua
NSLog(languages[0]);

//better use this method if you only need the current one
let currentLang : String = NSLocale.currentLocale().objectForKey(NSLocaleLanguageCode) as!
NSLog(currentLang); //en or de_ch ...
```

EXTERNAL LIBRARIES

USING NON APPLE LIBRARIES IN XCODE

There are two paket managers available:

- CocoaPods https://cocoapods.org/: For external libraries in your project
- Alcatraz http://alcatraz.io/: For Xcode plugins/themes

UNIT TESTS

UNIT AND UI TESTS

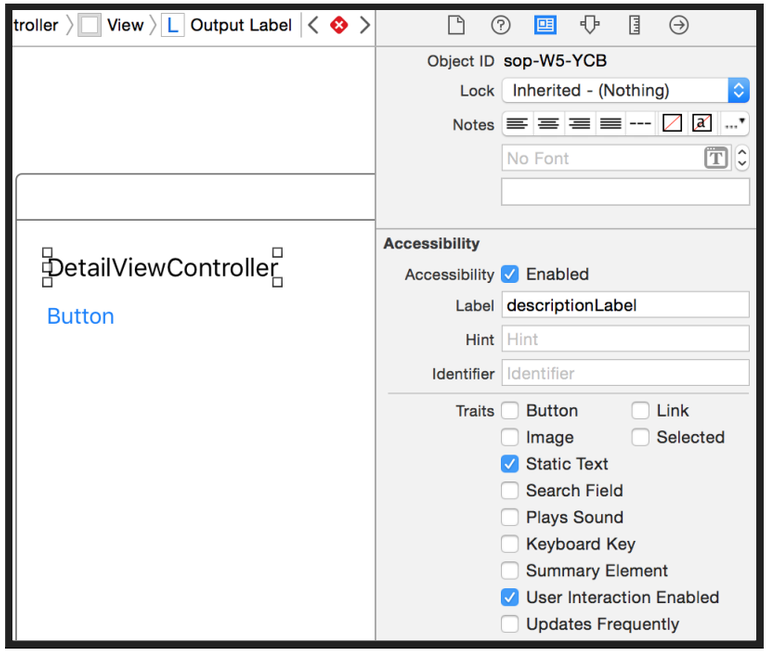
- You write test classes by inheriting from *XCTestCase* and adding a method that begins with *test*.
- Xcode provides a special language for writing UI tests.
- You can automatically generate a UI test by recording it.
- Very interesting, but right now very much work in progress. There are important things missing, right now (September 2015) it is not possible to get the value of a UILabel. There is also no documentation available.
- Watch the video for more information https://developer.apple.com/videos/play/wwdc2015-406/

A SIMPLE UI TEST I

```
class FirstProjectUITests: XCTestCase {

   //our test method
   func testButtonClick() {
        //you can use print, however, the console is automatically cleared after
        //the test is finished
        print ("testButtonClick started");

        let app = XCUIApplication()

        //we have a UIToolbar on our view that has a button named "Extend".
        //The tap() command simulates a tap event
        app.toolbars.buttons["Extend"].tap()

        print ("testButtonClick ended");
    }
```

ACCESSIBILITY LABEL

- For all elements that are not buttons, you need to manually set the accessibility label if you want to use subscription (app.staticTexts["descriptionLabel"]).
- The accessibility label can be set in your storyboard or programmatically.



A SIMPLE UI TEST II

```
func checkDescriptionTextFieldText() {

        //We want to make sure that the UITextField  text is equal
        //to "My description" (default setting from Storyboard).
        //The text of a UIText is stored in the attribute value
        let descriptionTextFieldEl = app.textFields["descriptionTextField"]

        //this will fail if the string is not "My description" (nil, other value
        //or other type)
        XCTAssertEqual(descriptionTextFieldEl.value as? String, "My description");
}
```

A SIMPLE UI TEST III

```
func setDescriptionTextFieldText() {

        //We want to set the value in a UITextField.
        let descriptionTextFieldEl = app.textFields["descriptionTextField"]

        //simulate a tap event on the text field so that it gets the focus
        //and the keyboard pops up
        descriptionTextFieldEl.tap()

        //Now type the text
        descriptionTextFieldEl.typeText("My description");
}
```

SEAMLESS LINKING

BASIC IDEA

- Use a web link to open a page in your app
- Fallback gracefully if the app is not installed by opening the link in a browser
- Create this service without the need to track the user's installed programs

EXAMPLE

Apples WWDC app and the following link:
http://developer.apple.com/videos/wwdc/2014/?include=101#101

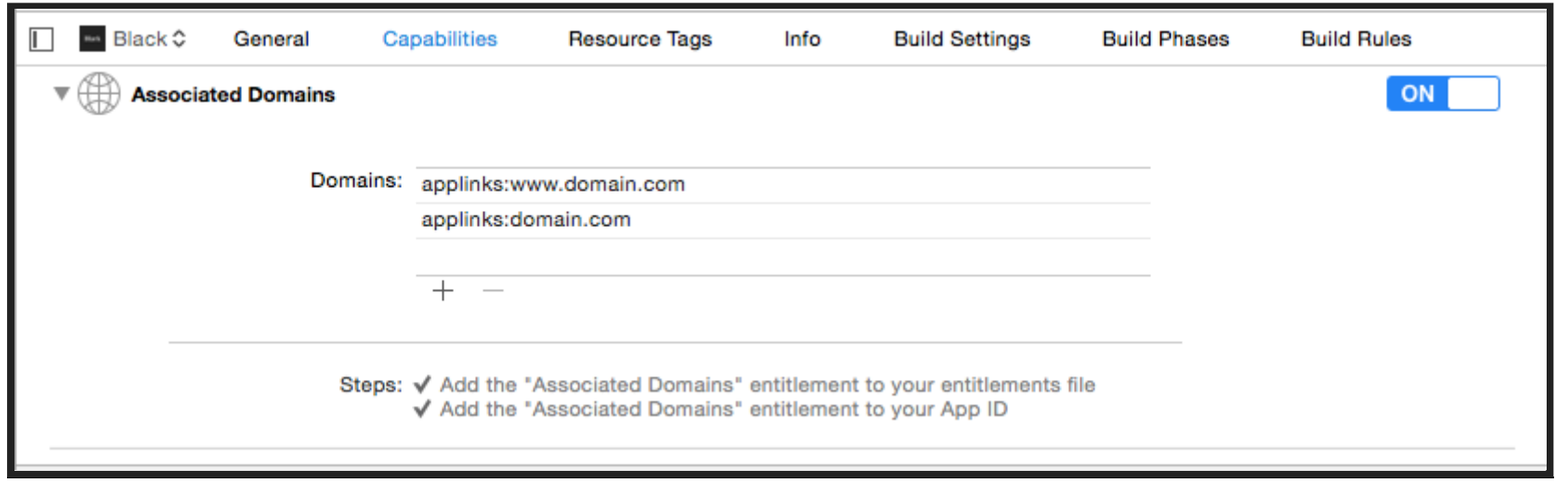
APPROACH

- Create a file called apple-app-site-association and put it on your web server
- Example: https://de.foursquare.com/apple-app-site-association
- The file specifies the App-Id of your app as well as the supported path(s) of the app

apple-app-site-association:

```
{
        "applinks": {
                "apps": [],
                "details": {
                        "MYTeamId.MyAppId": {
                                //"paths": ["*"]
                                "paths":
                                "/wwwdc/news",
                                "/videos/wwds/2015/*"
                        }
                }
        }
}
```

Add your domain to the Associated Domains section in Capabilities.



The following method will be called in your app (in AppDelegate):

```
func application(application: UIApplication,
        continueUserActivity userActivity: NSUserActivity,
        restorationHandler: ([AnyObject]?) -> Void) -> Bool {

                //check if the app was opened with a link
                if userActivity.activityType == NSUserActivityTypeBrowsingWeb {
                //if we are here, we have a value in webpageURL
                        let webpageURL = userActivity.webpageURL!
                        //parse webpageURL and do something in your app...
                        //open a special viewcontroller, etc.
                }
        ...
}
```

THINK ABOUT/DISCUSS

What do you think of this approach?

Copyright by Zürcher Hochschule für Angewandte Wissenschaften (ZHAW)