



**FREE eBook**

# LEARNING realm

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#realm**

# Table of Contents

About.....	1
<b>Chapter 1: Getting started with realm.....</b>	<b>2</b>
Remarks.....	2
Versions.....	2
Examples.....	2
Java.....	2
Objective-C.....	3
Swift.....	3
Xamarin.....	4
React-Native.....	4
<b>Chapter 2: Find Realm file location.....</b>	<b>5</b>
Examples.....	5
Print file location - Swift.....	5
Print file location - Objective-C.....	5
Print file location - Xamarin.....	5
How to reach the file :.....	5
Android.....	6
<b>Chapter 3: Installation and Setup.....</b>	<b>7</b>
Examples.....	7
Swift.....	7
React-Native.....	8
Xamarin.....	8
Objective C.....	8
Java.....	9
<b>Credits.....</b>	<b>10</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [realm](#)

It is an unofficial and free realm ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official realm.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with realm

## Remarks

Realm is a mobile-first database solution designed to provide fast queries with low-memory consumption through usage of lazy-evaluation, along with type-safe and live-updating model objects.

Realm is available for multiple platforms, and therefore encompasses multiple projects:

- [Realm Objective-C](#)
- [Realm Swift](#)
- [Realm Java](#)
- [Realm JavaScript](#)
- [Realm Xamarin](#)

## Versions

Platform	Version	Release date
<a href="#">Realm Objective-C</a>	2.4.4	2017-03-14
<a href="#">Realm Swift</a>	2.4.4	2017-03-14
<a href="#">Realm Java</a>	3.2.0	2017-05-16

## Examples

### Java

```
public class Dog extends RealmObject {
    public String name;
    public int age;
}

Dog dog = new Dog();
dog.name = "Rex";
dog.age = 1;

Realm realm = Realm.getDefaultInstance();
realm.executeTransaction(new Realm.Transaction() {
    @Override
    public void execute(Realm realm) {
        realm.copyToRealmOrUpdate(dog);
    }
});

RealmResults<Dog> pups = realm.where(Dog.class)
```

```
.lessThan("age", 2)
.findAll();
```

## Objective-C

```
@interface Dog : RLMObject
@property NSString *name;
@property NSInteger age;
@end
@implementation Dog
@end

Dog *dog = [Dog new];
dog.name = @"Rex";
dog.age = 1;

RLMRealm *realm = [RLMRealm defaultRealm];
[realm transactionWithBlock:^(
    [realm addObject:dog];
)];

RLMResults<Dog *> *allDogs = [Dog allObjects];
RLMResults<Dog *> *pups = [allDogs objectsWhere:@"age < 2"];
```

## Swift

### Create Object Class:

```
class Dog: Object {
    dynamic var name = ""
    dynamic var age = 0
}
```

### Assign Object's Values:

```
let dog = Dog()
dog.name = "Rex"
dog.age = 1
```

### Save Object:

```
let realm = try! Realm()
try! realm.write {
    realm.add(dog)
}
```

### Reading Objects:

```
let realm = try! Realm()
let pups = realm.objects(Dog.self)
```

### Filtering Objects:

```
let realm = try! Realm()
let filteredPups = realm.objects(Dog.self).filter("age < 2")
```

## Counting Objects:

```
let realm = try! Realm()
let pupsCount = realm.objects(Dog.self).count
```

## Xamarin

```
public class Dog : RealmObject
{
    public string Name { get; set; }
    public int Age { get; set; }
}

var realm = Realm.GetInstance();
realm.Write(() =>
{
    var mydog = realm.CreateObject<Dog>();
    mydog.Name = "Rex";
    mydog.Age = 1;
});

var pups = realm.All<Dog>().Where(d => d.Age < 2);
```

## React-Native

```
class Dog {}

Dog.schema = {
  name: 'Dog',
  properties: {
    name: 'string',
    age: 'int',
  }
};

let realm = new Realm();
realm.write(() => {
  realm.create('Dog', {name: 'Rex', age: 1});
});

let pups = realm.objects('Dog').filtered('age > 2');
```

Read Getting started with realm online: <https://riptutorial.com/realm/topic/1042/getting-started-with-realm>

---

# Chapter 2: Find Realm file location

## Examples

### Print file location - Swift

Add the following line to `ViewDidLoad` method:

```
print(Realm.Configuration.defaultConfiguration.fileURL!)
```

The line above will print the location to Xcode's console. Copy the file path, go to **Finder** → Go → Go to Folder... (or **++G**) → paste the path and hit Go.

### Print file location - Objective-C

Log the realm file location using:

```
NSLog(@"%@", [RLMRealmConfiguration defaultConfiguration].fileURL);
```

The line above will print the location to Xcode's console. Copy the file path, go to **Finder** → Go → Go to Folder... (or **++G**) → paste the path and hit Go.

### Print file location - Xamarin

First you need to implement Realm at the start of your class

```
using Realms;
```

Then to print the location to the console :

```
Console.WriteLine( RealmConfiguration.PathToRealm() );
```

Or if you're using `DefaultConfiguration`, you can use :

```
Console.WriteLine( RealmConfiguration.DefaultConfiguration.DatabasePath );
```

## How to reach the file :

If you're running on **IOS simulator** :

You can copy the file path, go to **Finder** → Go → Go to Folder... (or **++G**) → paste the path and hit Go.

But if you're running on **Android emulator** :

Open **Android device Monitor** (on visual studio → tools menu → Android → Android device Monitor ) (on Xamarin studio → Tools menu → Open Android device Monitor ) → File Explorer tab → follow the file path

## Android

Copy the database from the emulator/phone to view it. It can be done by using ADB:

```
adb pull /data/data/<packagename>/files/
```

That command will pull all Realm files created by `Realm.getInstance(getContext())` or `Realm.getInstance(new RealmConfiguration.Builder(context).build())`. The default database file is called `default.realm`.

Note that this will only work on a emulator or rooted device.

Read Find Realm file location online: <https://riptutorial.com/realm/topic/2488/find-realm-file-location>



# Chapter 3: Installation and Setup

## Examples

### Swift

- [Prerequisites](#)

1. iOS 8 or later, macOS 10.9 or later, all versions of tvOS and watchOS.
2. Xcode 8.0 or later required. [Realm Swift 2.3.0](#) was the last version to support Swift 2.x and Xcode 7.3.

- [Installation](#)

- Dynamic Framework

1. Download [the latest release of Realm](#) and extract the zip.
2. Go to your Xcode project's "General" settings. Drag `RealmSwift.framework` and `Realm.framework` from the appropriate Swift-versioned directory for your project in `ios/`, `osx/`, `tvos/` or `watchos/` directory to the "Embedded Binaries" section. Make sure **Copy items if needed** is selected (except if using Realm on multiple platforms in your project) and click **Finish**.
3. In your unit test target's "Build Settings", add the parent path to `RealmSwift.framework` in the "Framework Search Paths" section.
4. If using Realm in an iOS, tvOS or watchOS project, create a new "Run Script Phase" in your app's target's "Build Phases" and paste the following snippet in the script text field:

```
bash "${BUILT_PRODUCTS_DIR}/${FRAMEWORKS_FOLDER_PATH}/Realm.framework/strip-frameworks.sh"
```

This step is required to work around an [App Store submission bug](#) when archiving universal binaries.

- CocoaPods

1. [Install CocoaPods 0.39.0 or later.](#)
2. Run `pod repo update` to make CocoaPods aware of the latest available Realm versions.
3. In your Podfile, add `use_frameworks!` and `pod 'RealmSwift'` to your main and test targets.
4. If using Xcode 8, paste the following at the bottom of your Podfile, updating the Swift version if necessary:

```
post_install do |installer|
  installer.pods_project.targets.each do |target|
    target.build_configurations.each do |config|
      config.build_settings['SWIFT_VERSION'] = '3.0'
    end
  end
end
```

5. From the command line, run `pod install`.
6. Use the `.xcworkspace` file generated by CocoaPods to work on your project!
  - Carthage
    1. [Install Carthage 0.17.0 or later](#).
    2. Add `github "realm/realm-cocoa"` to your Cartfile.
    3. Run `carthage update`.
    4. Drag `RealmSwift.framework` and `Realm.framework` from the appropriate platform directory in `Carthage/Build/` to the “Linked Frameworks and Libraries” section of your Xcode project’s “General” settings.
    5. **iOS/tvOS/watchOS:** On your application targets’ “Build Phases” settings tab, click the “+” icon and choose “New Run Script Phase”. Create a Run Script with the following contents:

```
/usr/local/bin/carthage copy-frameworks
```

and add the paths to the frameworks you want to use under “Input Files”, e.g.:

```
$(SRCROOT)/Carthage/Build/iOS/Realm.framework
$(SRCROOT)/Carthage/Build/iOS/RealmSwift.framework
```

This script works around an [App Store submission bug](#) triggered by universal binaries.

## React-Native

- [Prerequisites](#)
- [Installation](#)

## Xamarin

- [Prerequisites](#)
- [Installation](#)

## Objective C

- [Prerequisites](#)

1. iOS 7 or later, macOS 10.9 or later, all versions of tvOS and watchOS.
2. Xcode 7.3 or later required.

- [Installation](#)

1. Download the latest release of Realm files from [here](#) or from [Github link](#) and extract the zip.
2. Navigate to `ios/static/` directory
3. Drag `Realm.framework` to the File Navigator of your Xcode project. Make sure Copy items if needed is selected and click Finish.
4. Click on your project in the Xcode File Navigator. Select your app’s target and go to the Build

Phases tab. Under Link Binary with Libraries click + and add libc++.tbd.

## Java

- [Prerequisites](#)
- [Installation](#)

Read Installation and Setup online: <https://riptutorial.com/realm/topic/1521/installation-and-setup>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with realm	<a href="#">Community</a> , <a href="#">EpicPandaForce</a> , <a href="#">Marc</a> , <a href="#">sangjoon moon</a> , <a href="#">Sergey</a> , <a href="#">ZGski</a>
2	Find Realm file location	<a href="#">Agung Santoso</a> , <a href="#">Idan</a> , <a href="#">MujtabaFR</a>
3	Installation and Setup	<a href="#">EpicPandaForce</a> , <a href="#">Prav</a> , <a href="#">sangjoon moon</a>