



iOS Development

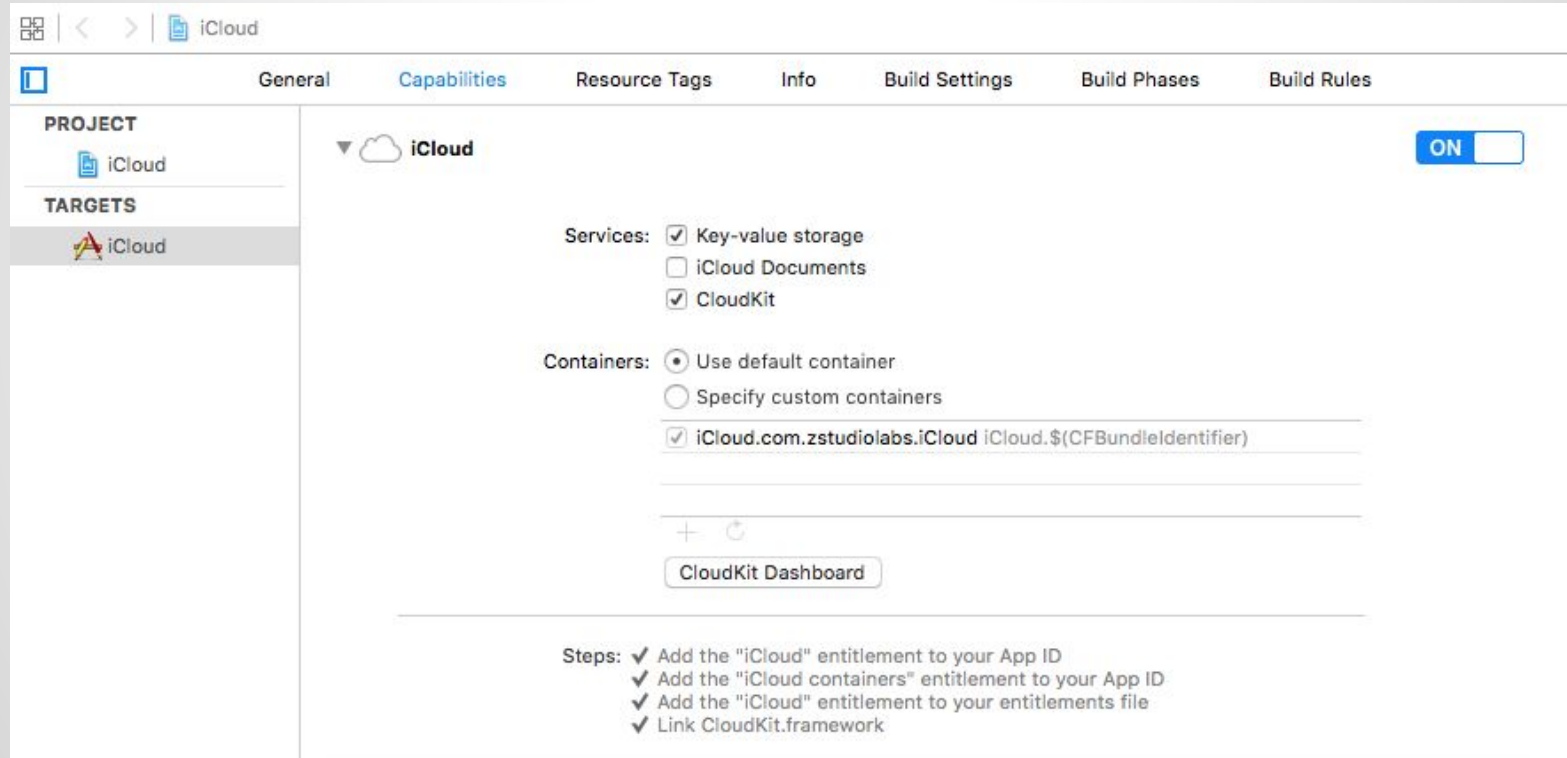
iCloud

Readings

[iCloud Tutorial](#) - Part 1

[iCloud Tutorial](#) - Ray Wenderlich

Enable iCloud in Your Project



Save a record

```
import CloudKit

class CloudKitHelper{

    static var container : CKContainer = CKContainer.defaultContainer()
    static var publicDB : CKDatabase = CKContainer.defaultContainer().publicCloudDatabase
    static let privateDB : CKDatabase = CKContainer.defaultContainer().privateCloudDatabase

    static func saveRecord(recordValue : String){

        let newRecord = CKRecord(recordType: "Buildings")
        newRecord.setValue(recordValue, forKey: "Name")
        publicDB.saveRecord(newRecord, completionHandler: { record, error in
            if error != nil {
                print(error)
            }
        })
    }
}
```

Retrieve a Record

```
let container = CKContainer.defaultContainer()
let publicData = container.publicCloudDatabase
let query = CKQuery(recordType: "Building", predicate: NSPredicate(format: "TRUEPREDICATE", argumentArray: nil))

publicData.performQuery(query, inZoneWithID: nil) { results, error in
    if error == nil { // There is no error
        for bug in results {
            let newBuilding          = Building()
            newBuilding.name          = newBuilding["Name"] as! String
            newBuilding.description = newBuilding["Description"] as! String

            self.objects.append(newBuilding)

            dispatch_async(dispatch_get_main_queue(), { () -> Void in
                self.tableView.reloadData() // Data retrieved, reload a table view
            })
        }
    }
    else {
        print(error)
    }
}
```

Deleting a Record

```
let container = CKContainer.defaultContainer()
let publicData = container.publicCloudDatabase
let bug = self.objects[indexPath.row]
let query = CKQuery(recordType: "Building", predicate: NSPredicate(format: "(Title == %@) AND (Description == %@)", argumentArray: [bug.title, bug.description]))

publicData.performQuery(query, inZoneWithID: nil, completionHandler: { results, error in
    if error == nil {
        if results.count > 0 {
            let record: CKRecord! = results[0] as! CKRecord
            println(record)

            publicData.deleteRecordWithID(record.recordID, completionHandler: { recordID, error in
                if error != nil {
                    print(error)
                }
            })
        }
    }
    else {
        println(error)
    }
})
```