Contents

[**I. Clould Firestore** 1](#_Toc123737258)

[1. Clould Firestore Data Model: 1](#_Toc123737259)

[1.1 Collection and Document 1](#_Toc123737260)

[1.2 Reference 2](#_Toc123737261)

[2. Add Data to firebase database 3](#_Toc123737262)

[2.1 Set a document 3](#_Toc123737263)

[2.2 Add a document 4](#_Toc123737264)

[2.4 Update field in nested object 5](#_Toc123737265)

[3. Delete Data 6](#_Toc123737266)

[3.1 Delete document 6](#_Toc123737267)

[3.2 Delete fields 6](#_Toc123737268)

[4. Read Data from Firebase 6](#_Toc123737269)

[4.1 Get data 6](#_Toc123737270)

[4.1.1 Get a document 6](#_Toc123737271)

[4.1.2 Get multiple documents from a collection 7](#_Toc123737272)

[4.1.3 Get all documents 7](#_Toc123737273)

[4.2 Get realtime updates with clould Firestore 8](#_Toc123737274)

[5. Perform simple and compound queries in clould Firestore 9](#_Toc123737275)

[5.1 Simple Query 10](#_Toc123737276)

[6. Order and limit data with Clould Firestore 12](#_Toc123737277)

[II. Storage 13](#_Toc123737278)

# **I. Clould Firestore**

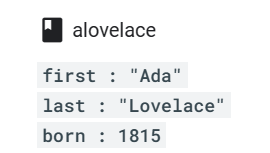
# 1. Clould Firestore Data Model:

## 1.1 Collection and Document

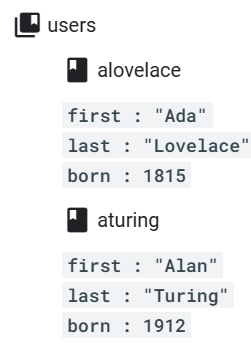
Cloud Firestore is a NoSQL, document-oriented database. Unlike a SQL database, there are no tables or rows. Instead, you store data in documents, which are organized into collections.

Nghĩa là Firestore là document database. Mỗi 1 data được lưu trong 1 document. Và 1 collection thì chứa nhiều documents.

Each document contains a set of key-value pairs. Cloud Firestore is optimized for storing large collections of small documents.

All documents must be stored in collections. Documents can contain subcollections and nested objects.

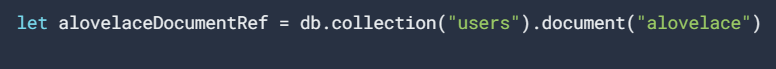
VD: document Alovelace chứa các cặp key value.



VD: collection users chứa 2 docmument là alovelace và aturing

## 1.2 Reference

Cùng trong 1 collection, mỗi 1 document trong Firebase database có 1 identified duy nhất trong firebase. Identified là 1 chuỗi string bất kì nào đó. Để có thể reference đến document đó thì trong code ta thực hiện như sau.



Lúc này biến alovelaceDocumentRef sẽ reference đến document có id là **alovelace** trong collection **user.**

Ta cũng có thể reference đến bất kì vùng nhớ nào trên database mà ta muốn:

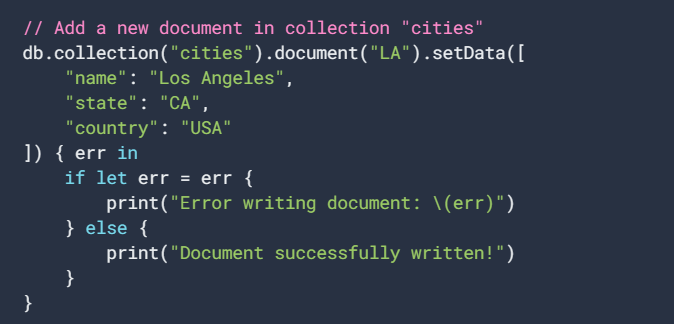


VD trên là ta reference đến collection **users**

# 2. Add Data to firebase database

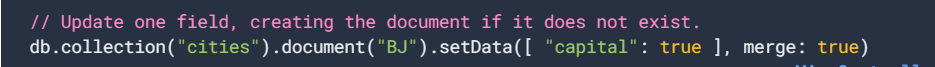
## 2.1 Set a document

To **create or overwrite** a single document, use the following language-specific set() methods. If the document does not exist, it will be created. If the document does exist, its contents will be overwritten with the newly provided data



Lúc này document có id là **LA** sẽ được tạo trong collection **cities** với data như kia, nếu đã tồn tại id là **LA**  thì sẽ ghi đè lên data cũ.

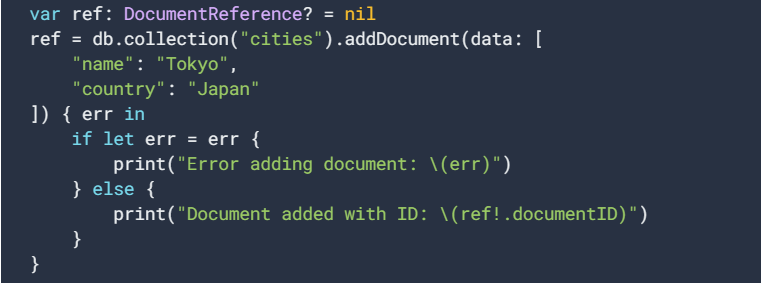
If you’re not sure whether the document exists**, psass the option to merge** the new data with any existing document to **avoid overwriting entire documents**.



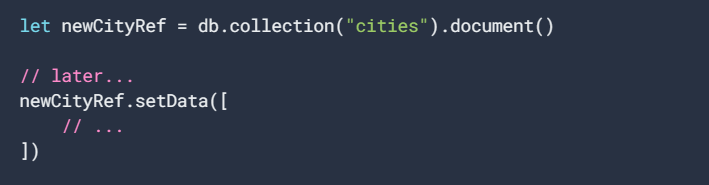
**Nhớ là ko có merge thì sẽ bị ghi đè đó, chứ ko phải update thêm**

## 2.2 Add a document

**When you use set() to create a document, you must specify an ID for the document to create. But sometimes there isn’t a meaningful ID for the document, and it’s more convenient to let Cloud Firestore auto-generate an ID for you**. You can do this by calling the following language-specific add() methods:



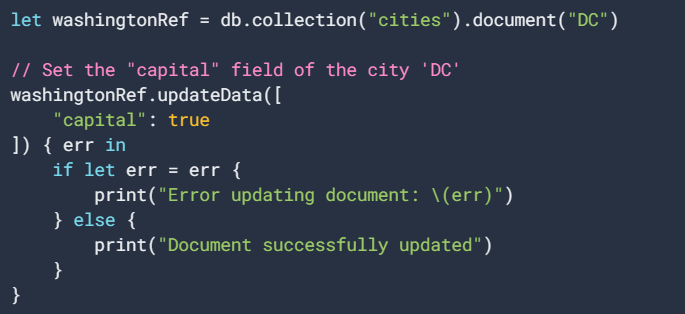
**In some cases, it can be useful to create a document reference with an auto-generated ID, then use the reference later.**



Nghĩa là trong document() ta không định danh id thì khi push lên clould sẽ định danh id cho document đó.

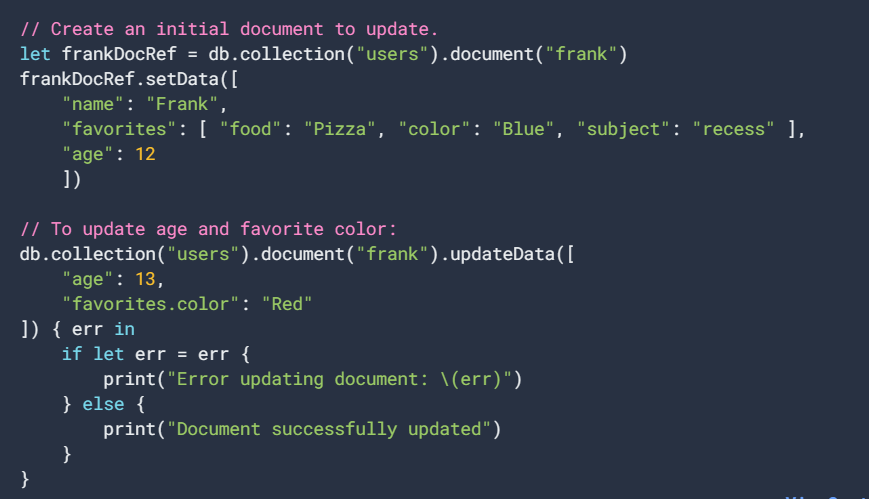
2.3 Update a document

To update some fields of a document without overwriting the entire document, use the following language-specific update() methods:



## 2.4 Update field in nested object

Chúng ta vẫn sử dụng phương thức update()

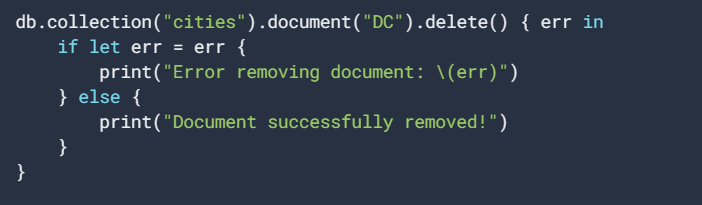


Chúng ta sử dụng dấu . như ở trên để truy cập vào từng phần tử trong favorites.

## 3. Delete Data

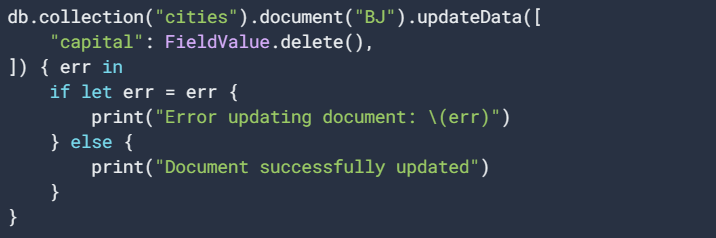
## 3.1 Delete document

Ta sử dụng method delete()



## 3.2 Delete fields

To delete specific fields from a document, use the following language-specific FieldValue.delete() methods when you update a document:

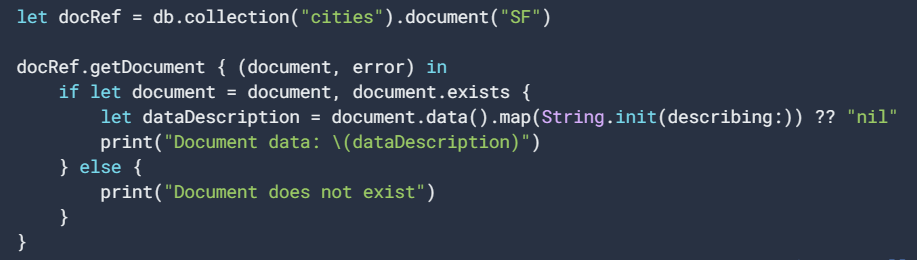


# 4. Read Data from Firebase

## 4.1 Get data

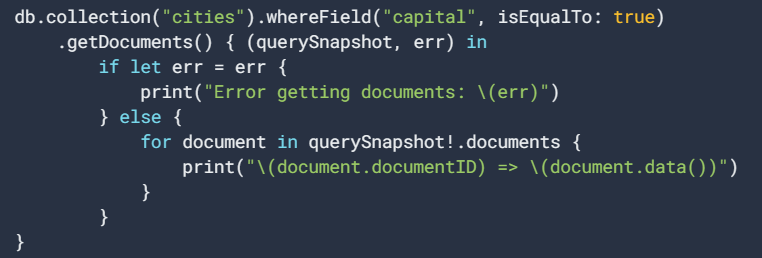
### 4.1.1 Get a document

The following example shows how to retrieve the contents of a single document using get():



### 4.1.2 Get multiple documents from a collection

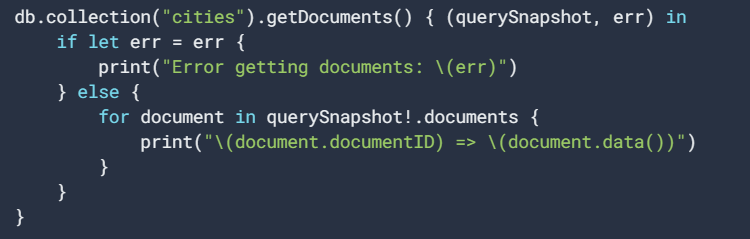
You can also retrieve multiple documents with one request by querying documents in a collection. For example, you can use where() to query for all of the documents that meet a certain condition, then use get() to retrieve the results:



Nghĩa là ta sẽ filter document với whereField(). Bất kì document nào có key “capital” mà có value là “true” sẽ được get về.

### 4.1.3 Get all documents

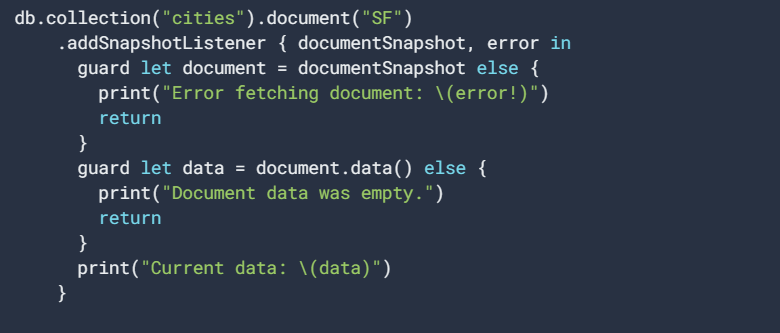
In addition, you can retrieve all documents in a collection by omitting(bỏ qua) the where() filter entirely:



**Nhớ là getDocuments chứ không phải getDocument(cái này dành cho get 1 document)**

## 4.2 Get realtime updates with clould Firestore

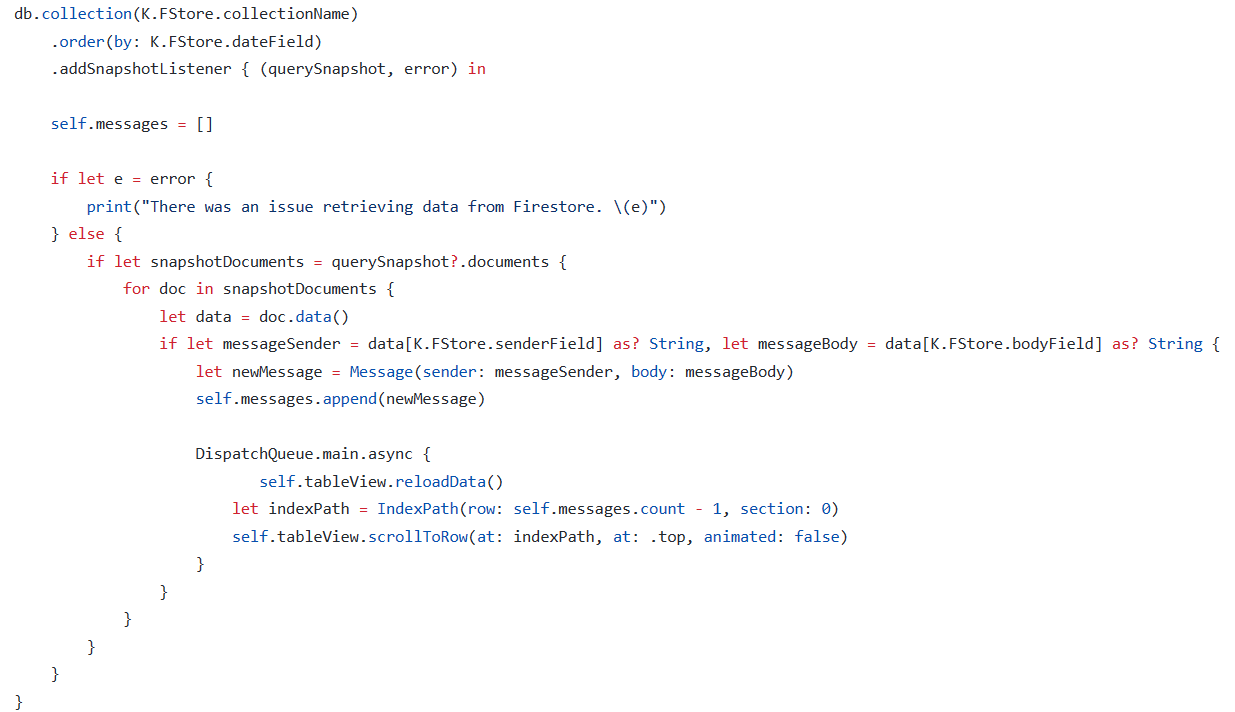
You can listen to a document with the onSnapshot() method. An initial call using the callback you provide creates a document snapshot immediately with the current contents of the single document. Then, each time the contents change, another call updates the document snapshot.



Local writes in your app will invoke snapshot listeners immediately. This is because of an important feature called "latency compensation(độ trễ)." When you perform a write, your listeners will be notified with the new data before the data is sent to the backend.

Nghĩa là mỗi khi ta thực hiện 1 action write lên database, như kia là collection **cities và document SF**, thì hàm listen kia sẽ được thông báo và cập nhập data.

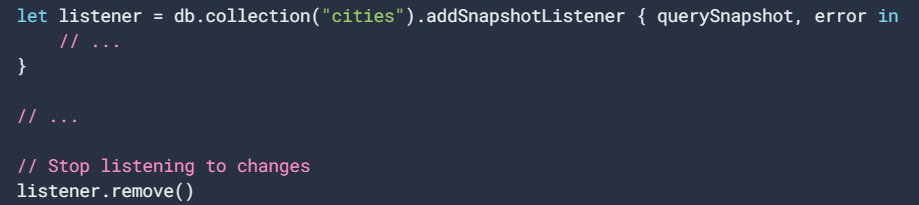
Tương tự khi ta muốn mỗi lần them document trong 1 collection nào đó ta cũng có thể làm như này:



Biến querySnapshot bây giờ lưu toàn bộ cá documents trong collection ta listen.

* Detach a listener

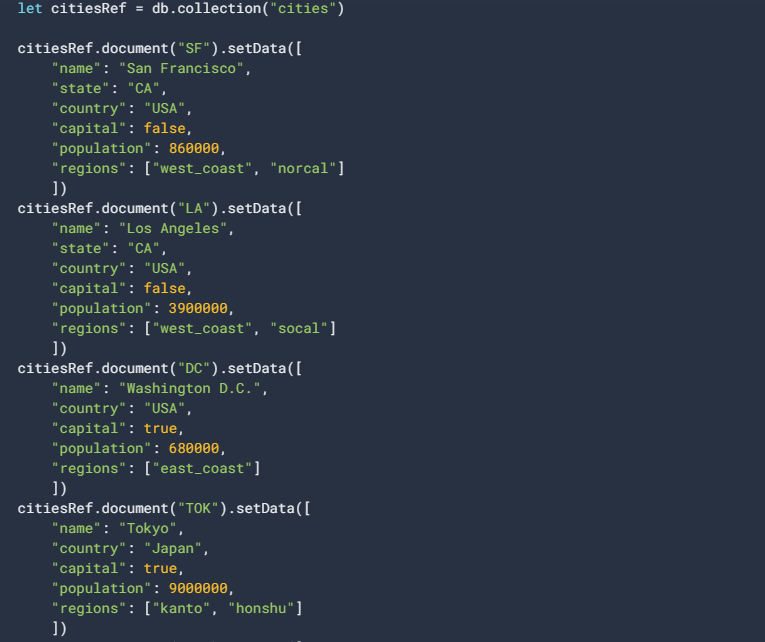
When you are no longer interested in listening to your data, you must detach your listener so that your event callbacks stop getting called. This allows the client to stop using bandwidth to receive updates. For example:



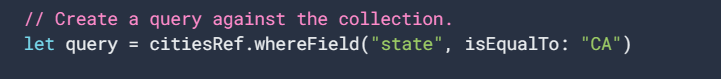
# 5. Perform simple and compound queries in clould Firestore

<Thực hiện các lệnh truy vấn đơn giản và phức tạp>

VD: Ta có data vừa push lên như thế này



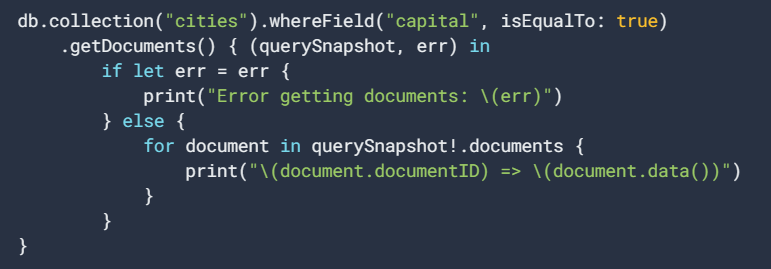
## 5.1 Simple Query



The following query returns all cities with **state** mà bằng **CA**:

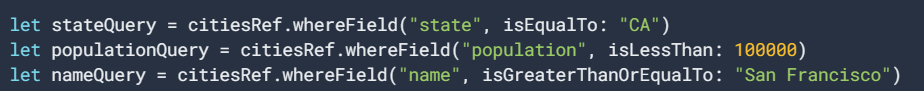


The following query returns all the **capital** cities mà có value là true



Ngoài **isEqualTo** ta còn có các lệnh khác như:

* isGreaterThanOrEqualTo
* isLessThan

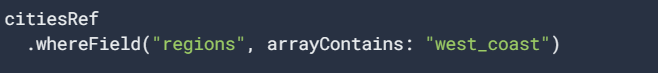


* isNotEqualTo

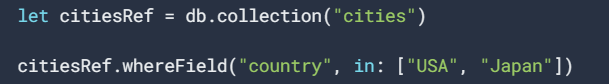


This query returns every city document where the capital field exists with a value other than false or null

* arrayContains



* In



Use the in operator to combine up to 10 equality (==) clauses on the same field with a logical OR. An in query returns documents where the given field matches any of the comparison values. Cứ key là country mà có value là USA hoặc Japan thì sẽ được get về.

* notIn



This query returns every city document where the country field exists and is not set to USA, Japan, or null.

* arrayContainsAny



An array-contains-any query returns documents where the given field is an array that contains one or more of the comparison values

# 6. Order and limit data with Clould Firestore

Cloud Firestore provides powerful query functionality for specifying which documents you want to retrieve from a collection. These queries can also be used with either get() or addSnapshotListener(). By default, a query retrieves all documents that satisfy the query in ascending order by document ID. You can specify the sort order for your data using orderBy(), and you can limit the number of documents retrieved using limit().

Tóm lại:

* Ta dùng **query** đê filter các documents ta muốn lấy
* Ta dùng **Order** để sắp xếp các document theo thứ tự ta thu về
* Ta dùng **Limit** để giới hạn documents ta muốn lấy

For example, you could query for the first 3 cities alphabetically with:



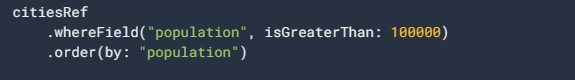
You could also sort in descending order to get the last 3 cities:



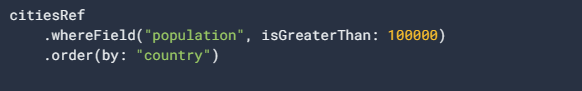
Note the following restrictions for orderBy() clauses:

* An orderBy() clause also filters for existence of the given fields. The result set will not include documents that do not contain the given fields.
* If you include a filter with a range comparison (<, <=, >, >=), your first ordering must be on the same field:

**Valid**: Range filter and orderBy on the same field



**Invalid**: Range filter and first orderBy on different fields



Tóm lại là order và whereField phải cùng 1 filed

# **II. Storage**

Ta thấy ở phần I Clouse Firestore, data ta push lên chỉ được lưu dưới dạng key, value là các chuỗi string, int,… Ta không thể lưu được cái file như ảnh, video.

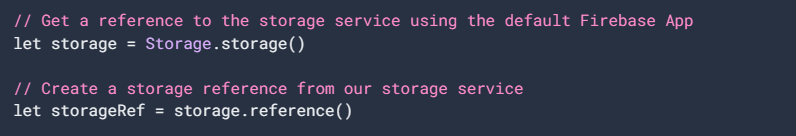
Storage được cung cấp để giải quyết vấn đề này. Cloud Storage for Firebase lets you upload and share user generated content, such as images and video, which allows you to build rich media content into your apps.

# Create Reference

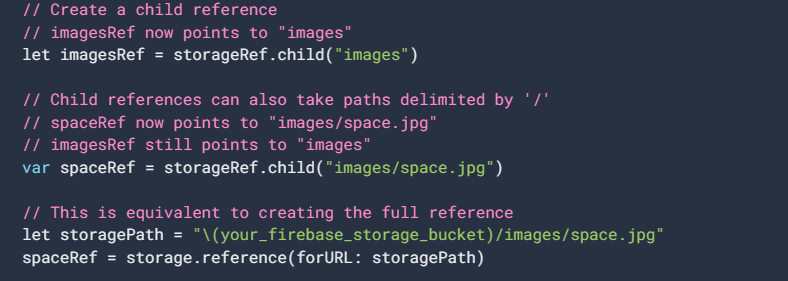
. By creating a reference to a file, your app gains access to it. These references can then be used to upload or download data, get or update metadata or delete the file. A reference can either point to a specific file or to a higher level node in the hierarchy(hệ thống cấp bậc).

Create a reference to upload, download, or delete a file, or to get or update its metadata. A reference can be thought of as a pointer to a file in the cloud. References are lightweight, so you can create as many as you need.

References are created using the FirebaseStorage service and calling its reference method.

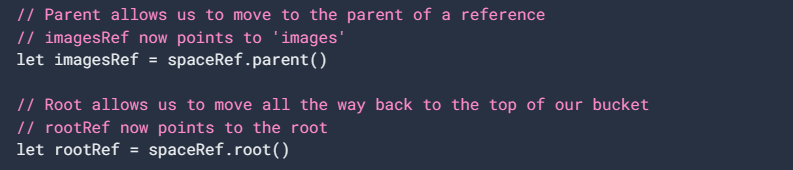


You can create a reference to a location lower in the tree, say 'images/space.jpg', by using the child method on an existing reference.

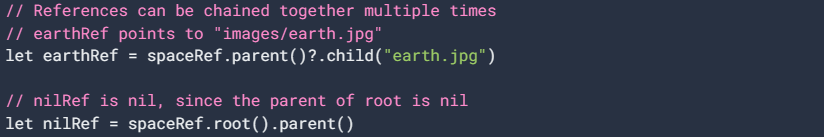


* Navigate with reference

You can also use the parent and root methods to navigate up in our file hierarchy. parent navigates up one level, while root navigates all the way to the top.

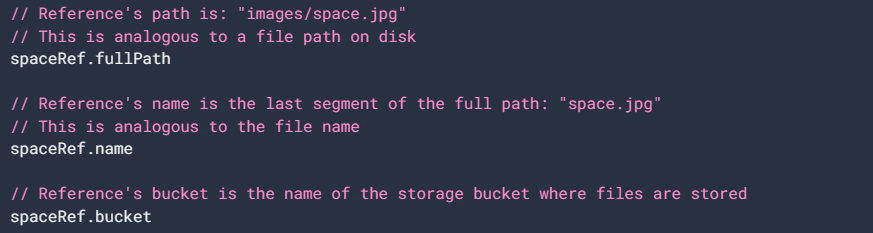


child, parent, and root can be chained together multiple times, as each returns a reference. The exception is the parent of root, which is nil.

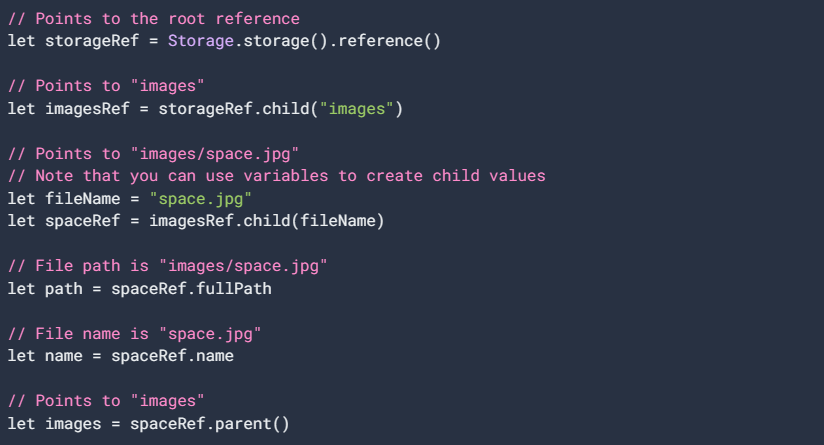


* Refeence Properties

You can inspect references to better understand the files they point to using the fullPath, name, and bucket properties. These properties get the file's full path, name, and bucket.



Tổng kết Example:

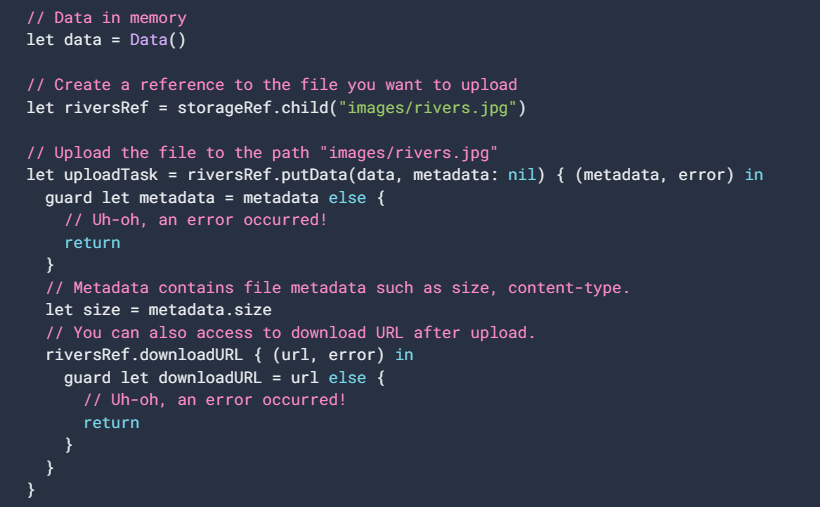


# Upload files

Once you have a reference, you can upload files to Cloud Storage in two ways:

1. Upload from data in memory
2. Upload from a URL representing a file on device

## 2.1 Upload from data in memory

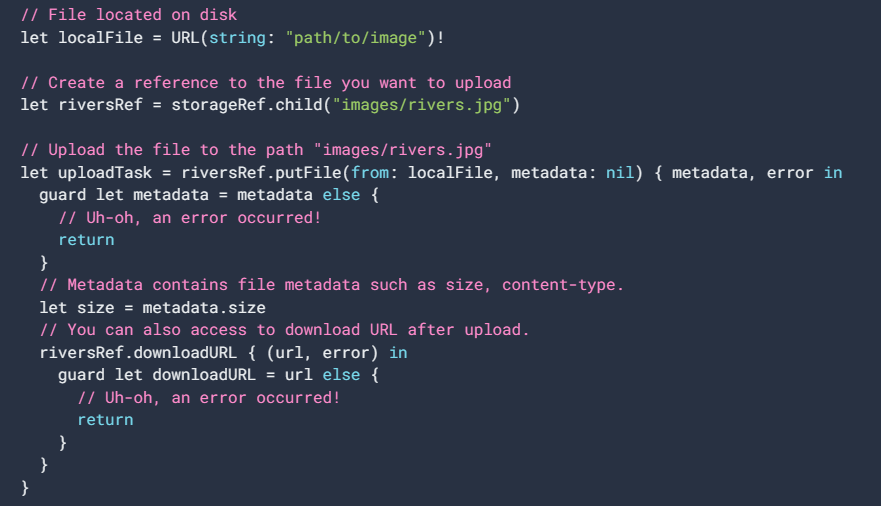


Suy luận là file lúc này được push lên dưới dạng file text😊))

Ta thấy biến reference **reversRes** reference tới file ta push lên, lên nó có quyền push data lên hoặc download data về thông qua reversRef.downloadURL

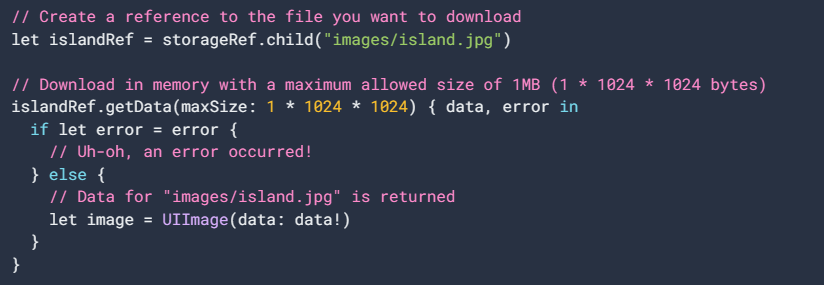
## 2.2 Upload from a local file

You can upload local files on the devices, such as photos and videos from the camera, with the putFile:metadata:completion: method. putFile:metadata:completion: takes an NSURL and returns an FIRStorageUploadTask, which you can use to manage your upload and monitor its status.



## 2.3 Dowload files

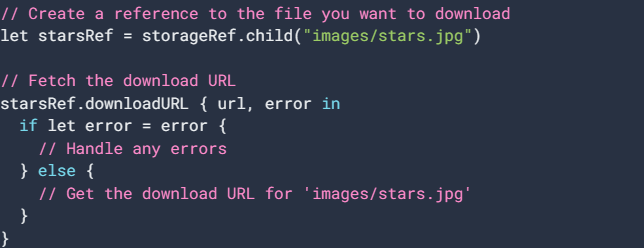
## 2.3.1 Download in memory



Download the file to an NSData object in memory using the dataWithMaxSize:completion: method. This is the easiest way to quickly download a file, but it must load entire contents of your file into memory**. If you request a file larger than your app's available memory, your app will crash**. To protect against memory issues, make sure to set the max size to something you know your app can handle, or use another download method.

## 2.3.2 Generate a download URL

If you already have download infrastructure based around URLs, or just want a URL to share, you can get the download URL for a file by calling the downloadURLWithCompletion: method on a Cloud Storage reference.



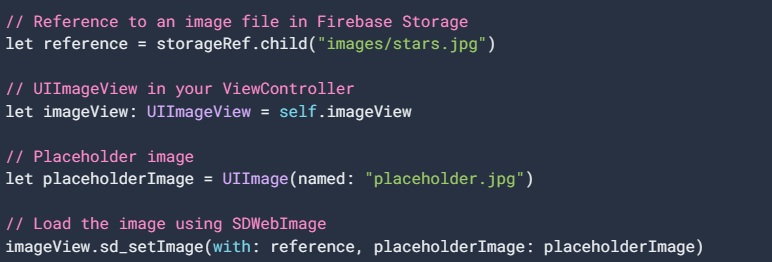
Biến url ở clousure sẽ chứa link file tới starts.jpg. ta sử dụng link đó để share,…

## 2.3.3 Download Images with firebaseUI

First, add FirebaseUI to your Podfile:



Then you can load images directly from Cloud Storage into a UIImageView:



Hmm placeholder không biết là nội dung ở dưới ảnh(kiểu tên ảnh) hay là tên ảnh nằm trong khung ảnh 😊))

VD về upload Profile, url cho image,…



Kết quả:

