Ch12 Lab

Firebase에 있는 Authentication과 Database 세트(real time database, firestore, storage)에 대한 간간한 실습

1. 준비사항

* Ch12-FirebaseLab라는 다운 받는다.
  + 터미널을 열어 ch12-FirebaseLab 디렉토리로 이동한다
  + Podfile을 다음과 같은 내용을 추가한다.(이미 만들어져 있다

|  |
| --- |
| # Uncomment the next line to define a global platform for your project  # platform :ios, '9.0'  target 'ch12-FirebaseLab' do  # Comment the next line if you don't want to use dynamic frameworks  use\_frameworks!  # Pods for ch12-FirebaseLab  pod 'Firebase/Auth'  pod 'Firebase/Database'  pod 'Firebase/Firestore'  pod 'Firebase/Storage'  end |

* + $ pod install을 실행한다// 시간이 많이 걸릴수도 있다.
  + 프로젝트를 열고 컴파일 하면 에러가 난다. 다음과 같은 조치를 하라.
* 텍스트이(가) 표시된 사진

  자동 생성된 설명텍스트이(가) 표시된 사진

  자동 생성된 설명
  + Bundle identifier을 자신에 맞게 적절히 수정하라.
    - hansung.ac.kr.ch12-FirebaseLab
* Firebase console로 로그인한다.
  + FirebaseLab라는 프로젝트를 만든다.
  + iOS앱을 추가한다. 번들 identifier을 카피하여 붙여넣기를 하라.
  + GoogleService-Info.plist를 다운 받아 Ch12-FirebaseLab에 포함하라(파일 이름이 정확히 맞는지 체크하라)
  + 콘솔로 이동할 때 까지 “다음” 클릭하라
  + Authentication을 선택하여 시작하기를 하라.
    - 이메일/비밀번호를 선택하라
    - 사용자 설정을 활성화 시키고 저장하라
    - “users”탭으로 이동하여 한명의 사용자를 등록하라
      * [jmlee@hansung.ac.kr](mailto:jmlee@hansung.ac.kr) / 123456
  + Firestore database를 선택하고 데이터베이스 만들기를 하라
    - 프로덕션 모드에서 다음을 클릭, 사용설정 클릭하라
    - “규칙”탭으로 이동하라
      * False -> true로 변경하고 “게시”를 클릭하라
  + Realtime database를 선택하고 데이터베이스 만들기를 하라
    - “잠금모드로 시작”을 선택하고 “사용자 설정” 클릭하라
    - “규칙”탭으로 이동하라
      * 모든 False -> true로 변경하고 “게시”를 클릭하라

* + Storage를 선택하라
    - “규칙”탭으로 이동하라
      * False -> true로 변경하고 “게시”를 클릭하라

1. Authentication: 메일 ID로 계정관리 및 로그인 지원

* 컴파일 에러시 다음과 같은 조치를 하라
* Main.storboard를 MainOld.storyboard로 변경하고 Authentication.storyboard를 Main.storyboard로 수정하라
  + Xcode 13이상에서는 info.plist에서 아래와 같이 스토리보드를 Main 🡪 Authentication으로 변경하라

텍스트이(가) 표시된 사진

자동 생성된 설명

* AuthenticationViewController.swift
  + Firebase와 FirebaseAuth를 임포트하라
  + viewDidLoad에 다음과 같이 코딩하라
    - FirebaseApp.configure()
  + Login함수에 다음을 코딩하고 로그인이 되는지 체크하라

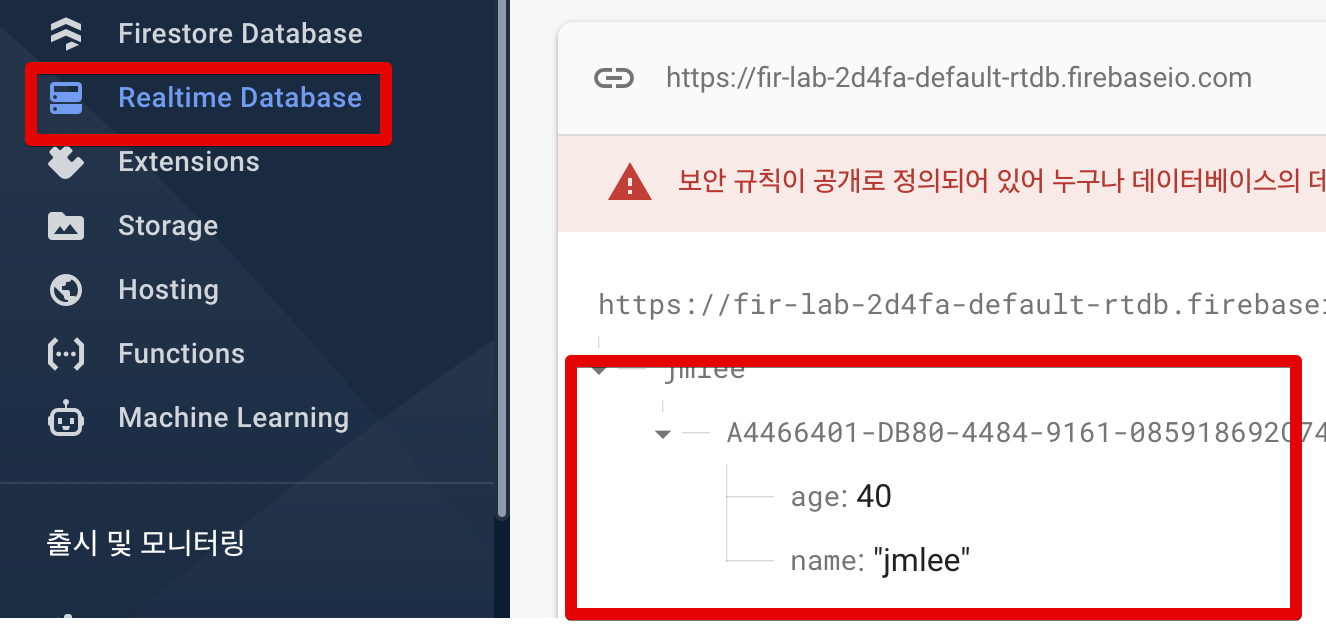
|  |
| --- |
| if let id = idTextField.text, let passwd = passwdTextField.text{  let email = id + "@hansung.ac.kr"  Auth.auth().signIn(withEmail: email, password: passwd) { result, error in  if let error = error{  print("Login Error: \(error)")  return  }  print("Login Success")  }  } |

* + Register 함수에 다음을 코딩하고 로그인이 되는지 체크하라
    - Id와 passwd를 읽어와서 email = id + “@hansung.ac.kr”를 만들어라
    - Auth.auth().createUser 함수를 호출하여 새로운 사용자를 등록하고 로그인인 되는지 체크하라

1. Realtime Database사용

* Main.storboard를 Authentication.storyboard로 변경하고 Database.storyboard를 Main.storyboard로 수정하라
  + Xcode 13이상에서는 info.plist에서 스토리보드를 Authentication 🡪 Database로 변경하라
* Firebase, FirebaseDatabase를 임포트하라.
* DatabaseViewController.swift에 다음 코딩을 하라
  + viewDidLoad에 다음과 같이 코딩하라
    - FirebaseApp.configure()
    - ref = Database.database().reference()
  + Add함수를 다음과 같이 코딩하고 실행해보라. 그리고 firebase console의 realtime database 에서 데이터가 저장되는지 확인하라

|  |
| --- |
| let key = UUID().uuidString  let nindex = Int(arc4random\_uniform(UInt32(names.count)))  let aindex = Int(arc4random\_uniform(UInt32(ages.count)))  let data = ["name": names[nindex], "age": ages[aindex]] as [String : Any]  ref.child("jmlee").child(key).setValue(data) |



* + viewDidLoad 함수에 다음 코드를 작성하라.

|  |
| --- |
| FirebaseApp.configure()  ref = Database.database().reference()  ref.child("jmlee").observe(.childAdded){  (snapshot) in  let key = snapshot.key  let name = (snapshot.value as! [String: Any])["name"] as? String  let age = (snapshot.value as! [String: Any])["age"] as? Int  print("\(key), \(name!), \(age!)")    self.keys.append(key)  print("added------------------------")  }    ref.child("jmlee").observe(.childRemoved){  (snapshot) in    let key = snapshot.key  let name = (snapshot.value as! [String: Any])["name"] as? String  let age = (snapshot.value as! [String: Any])["age"] as? Int  print("\(key), \(name!), \(age!)")    for i in 0..<self.keys.count{  if key == self.keys[i] {  self.keys.remove(at: i)  break  }  }  print("removed------------------------")  } |

* + Add함수를 클릭하는 경우 위의 .childAdd가 호출되어 print되는지 확인하라.
  + Remove함수를 다음과 같이 코딩하고 remnove가 될때마다 상기 .childRemoved가 호출되는지 확인하라.

|  |
| --- |
| if keys.count == 0{  print("Nothing to be removed")  return  }  let key = keys[keys.count-1]  ref.child("jmlee").child(key).removeValue() |

1. Firestore 사용

* Main.storboard를 Database.storyboard로 변경하고 Firestore.storyboard를 Main.storyboard로 수정하라
  + Xcode 13이상에서는 info.plist에서 스토리보드를 Database 🡪 Firestore로 변경하라
* 필요한 것들을 임포트하라.
* FirestoreViewController.swift에 다음 코딩을 하라
  + viewDidLoad에 다음과 같이 코딩하라
    - FirebaseApp.configure()
    - ref = Firestore.firestore().collection("jmlee")
    - Add함수를 다음과 같이 코딩하고 실행해보라. 그리고 firebase console의 firestore에 데이터가 저장되었는지 확인하라.

|  |
| --- |
| let key = UUID().uuidString  let nindex = Int(arc4random\_uniform(UInt32(names.count)))  let aindex = Int(arc4random\_uniform(UInt32(ages.count)))  let data = ["name": names[nindex], "age": ages[aindex]] as [String : Any]  ref.document(key).setData(data) |

테이블이(가) 표시된 사진

자동 생성된 설명

* + viewDidLoad 함수에 다음 코드를 작성하라.

|  |
| --- |
| FirebaseApp.configure()  ref = Firestore.firestore().collection("jmlee")  if let existQuery = existQuery{ // 이미 적용 쿼리가 있으면 제거, 중복 방지  existQuery.remove()  }  let queryReference = ref.whereField("name", isGreaterThanOrEqualTo: "").whereField("name", isLessThanOrEqualTo: "z")  existQuery = queryReference.addSnapshotListener(){  (snapshot, error) in    if let e = error{  print("query error: \(e)")  return  }    for documentChange in snapshot!.documentChanges {    let key = documentChange.document.documentID  let name = documentChange.document.data()["name"]  let age = documentChange.document.data()["age"]  print("\(key), \(name!), \(age!)")    if documentChange.type == .added{  self.keys.append(key)  print("added------------------------")  }  if documentChange.type == .removed{  for i in 0..<self.keys.count{  if key == self.keys[i] {  self.keys.remove(at: i)  break  }  }  print("removed------------------------")  }  }  } |

* + 프로그램이 시작되면 데이터가 Query되어 출력되는지 확인하라. 그리고 Add를 클릭하면 add된 데이터가 출력되는지 확인하라.
  + Remove함수를 다음과 같이 작성하고 역시 remove를 클릭하면 remove된 데이터가 출력되는지 확인하라.

|  |
| --- |
| if keys.count == 0{  print("Nothing to be removed")  return  }  let key = keys[keys.count-1]  ref.document(key).delete() |

1. Storage 사용

* Main.storboard를 Firestore.storyboard로 변경하고 Storage.storyboard를 Main.storyboard로 수정하라
  + Xcode 13이상에서는 info.plist에서 스토리보드를 Firestore 🡪 Storage로 변경하라
* 필요한 것들을 임포트하라.
* StorageViewController.swift에 다음 코딩을 하라
  + viewDidLoad에 다음과 같이 코딩하라
    - FirebaseApp.configure()
    - ref = Firestore.firestore().collection("jmlee")
  + Add함수를 다음과 같이 코딩하고 실행해보라. 그리고 firebase console의 firestore에 데이터가 저장되었는지 확인하라.

|  |
| --- |
| let key = UUID().uuidString  let nindex = Int(arc4random\_uniform(UInt32(names.count)))  let fileName = names[nindex]+".jpeg"  let image = UIImage(named: fileName)!  var data = Data()  data = image.jpegData(compressionQuality: 1.0)!  let metaData = StorageMetadata()  metaData.contentType = "image/jpeg"  ref.child(key).putData(data, metadata: metaData){  (metaData,error) in if let error = error { //실패  print(error)  return  }else{ //성공  print("성공")  }  } |

* + viewDidLoad 함수 마지막에 다음과 같은 코드를 하라

|  |
| --- |
| let tap = UITapGestureRecognizer(target: self, action: #selector(loadImage))  imageView.addGestureRecognizer(tap) |

.

* + loadImage 함수를 다음과 같이 작성하라. 그리고 마지막 이미지가 나오는지 확인하라.

|  |
| --- |
| @objc func loadImage(sender: UITapGestureRecognizer){  imageView.image = nil  ref.listAll { [self]  (result, error) in  if let error = error {  // ...  print(error)  }  var key: String?  for item in result.items {  key = item.name  }    if let key = key {  // Create a reference to the file you want to download  let imageRef = self.ref.child(key)    // Download in memory with a maximum allowed size of 1MB (1 \* 1024 \* 1024 bytes)  imageRef.getData(maxSize: 1 \* 1024 \* 1024) { data, error in  if let error = error {  // Uh-oh, an error occurred!  } else {  // Data for "images/island.jpg" is returned  let image = UIImage(data: data!)  self.imageView.image = image  }  }  }  }  } |

* + remove 함수를 다음과 같이 작성하라. 이미지가 삭제 되는지 확인하라.

|  |
| --- |
| ref.listAll {  (result, error) in  if let error = error {  // ...  print(error)  }  var key: String?  for item in result.items {  key = item.name  }    if let key = key {  self.ref.child(key).delete(){  (error) in  if let e = error{  print("Failure")  }else{  print("remove sucessfully")  }  }  }  } |