Firbase with Vue Cli

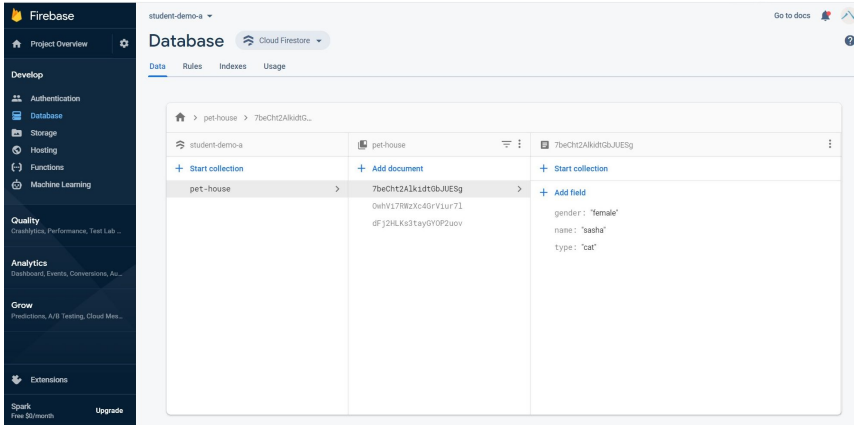
How to incorporate a firebase database into your Vue cli project.

Step 1 - Create an account on Firebase

Go to ***firebase.google.com*** and make an account, it’s free.

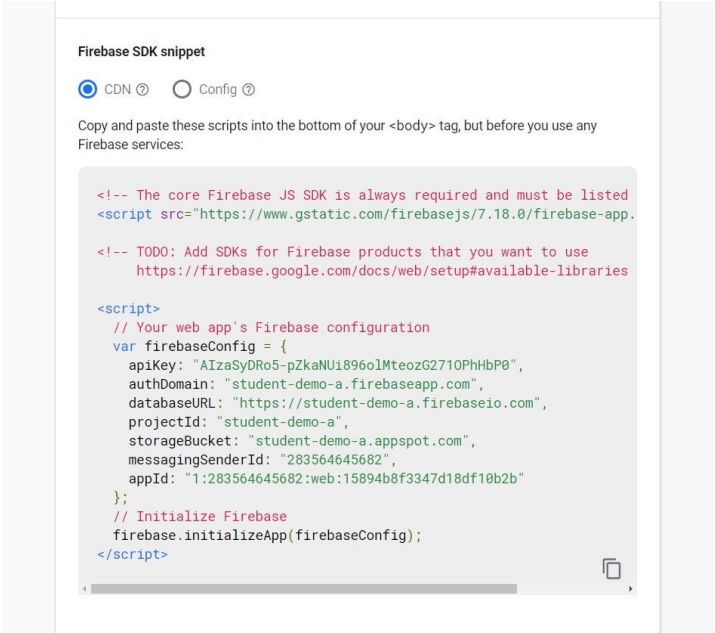
Step 2 - Create a database

Select firestore and create a test database with a collection and some documents. See pic on the right.



Step 3 - Get firebase config info

Go to the project settings page. The Config info is stored at the bottom of the settings page.



Step 4 - Install firebase into your project with npm

npm install firebase --save -dev

npm install firebase --save -dev

Step 5 - Import firebase into your project code

Use import to bring the firebase node module into the project

```
import firebase from 'firebase'
```

Step 6 - Initialize firebase into the code

Along with the import, place the javascript firebase initialise code within a .js file, or vue component <script> block.

```
import firebase from 'firebase'

// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: 'AIzaSyCIIBuT3P84SMkauTDcL5cUnYyUGmEtumg',
  authDomain: 'student-demo-live.firebaseapp.com',
  databaseURL: 'https://student-demo-live.firebaseio.com',
  projectId: 'student-demo-live',
  storageBucket: 'student-demo-live.appspot.com',
  messagingSenderId: '802402432813',
  appId: '1:802402432813:web:afbbac99a957785ead45a4'
}
// Initialize Firebase
var firebaseApp = firebase.initializeApp(firebaseConfig)
// Get the firebase database
var db = firebaseApp.firestore()
```

Step 7 - Use firebase functions

You can now access and use various firebase functions. Details at: https://firebase.google.com/docs/firestore/manage-data/add-data

```
deleteTheFish: function () {
  // 4. A method and js pattern that deletes a document from the database
  db.collection('pet-store').doc('fish-dude').delete().then(function () {
    console.log('Document successfully deleted!')
  }).catch(function (error) {
    console.error('Error removing document: ', error)
  })
}
```

```
// Pattern for getting and rendering all data in pet-house collection Ecma 6
// 1. A pattern that iterates through a collection listing all of the documents
db.collection('pet-store').onSnapshot(function (pets) {
  // pets is the data response or collection - we use a forEach
  // loop to loop through and then list
  pets.forEach(function (doc) {
    // eachDoc is a js object representing each document in the collection
    var eachDoc = doc.data()
    // logging eachDoc to the console
    console.log(eachDoc)
  })
})
```

```
Retrieving a single document Ecma 5
onSnapshot A snapshot is a picture of the data at a particular database reference
collection('pet-store').doc('fish-dude').onSnapshot(function (response) {
/ log the fish per object
console.log(response.data())


Adding a document to the collection
Add a new document in collection "cities"
collection('pet-store').doc('the-mouse').set({
ender: 'male',
ame: 'Squeek',
ype: 'mouse'

then(function () {
 console.log('Document successfully written!')
)
catch(function (error) {
 console.error('Error writing document: ', error)
```