Minhas colas de Firebase­:

**db.collection("turmaA").get()**

**.then((snapshot)=>{**

**snapshot.forEach((doc)=>{**

**let aluno = doc.data()**

**console.log(aluno.nome, aluno.sobrenome)**

**})**

**})**

Esse código serve para eu conseguir printar os dados de uma coleção criada na database do Firebase, dentro de um console.log.

**let docRef = db.collection("turmaA").doc("I2FDwe55Mz6Zif8DdnGI");**

**docRef.get().then((doc) =>{**

**let aluno = doc.data()**

**console.log(aluno.nome)**

**})**

Esse código serve para eu conseguir ver dados de um ID específico de uma coleção no firebase, as palavras dentro de doc() na primeira linha é a id que desejo estruturar.

**db.collection("turmaA").where("nome", "==", "Zé").get().then(snapshot=>{**

**snapshot.forEach((doc) =>{**

**let aluno = doc.data();**

**console.log(aluno.nome, aluno.sobrenome)**

**})**

**})**

Esse código serve para eu conseguir ver dados específicos de um banco de dados usando Where, similar ao where do mysql no PHP, porém com a sintaxe referente ao JS.

**db.collection(TURMA).doc("AlunoNovo").set(**

**{**

**nome: "Marcele",**

**sobrenome: "Silva",**

**notas: {nota1: 6.8, nota2: 9.5},**

**}**

**).then(()=>{**

**console.log("Documento inserido com sucesso:");**

**}).catch(err=>{**

**console.log(err)**

**})**

Esse código serve para eu conseguir criar novos documentos das coleções no Firebase, se eu por o mesmo código e editar o sobrenome, por exemplo, ele edita o conteúdo do ID que está no doc(“...”) e sobrescreve o atual.

OBS.: Sempre que for mudar algum dado, reescrever todos, se escrever apenas 1 ele apaga todos.

**db.collection(TURMA).doc("AlunoNovo").update(**

**{**

**sobrenome: "Santos",**

**}**

**).then(()=>{**

**console.log("Documento inserido com sucesso:");**

**}).catch(err=>{**

**console.log(err)**

**})**

Esse código serve para eu conseguir modificar e/ou adicionar elementos de um dado no firebase mas sem substituir todo o conteúdo, por isso uso aquele update para que possa atualizar dados e não substituir, para mudar as notas por exemplo eu colocaria “notas.nota1”: 9,

**db.collection(TURMA).doc("AlunoNovo").update(**

**{**

**Cidades: [“RJ”, “SP”]**

**}**

**).then(()=>{**

**console.log("Documento inserido com sucesso:");**

**}).catch(err=>{**

**console.log(err)**

**})**

Esse código serve para adicionar arrays ao banco de dados, o código abaixo vai ensinar como se modifica ele no lugar de cidade:xxxx

Cidades: firebase.firestore.FieldValue.arrayUnion(“novoobjetoarray”);

**db.collection(TURMA).doc("AlunoNovo").update(**

**{**

**Faltas : firebase.firestore.FieldValue.increment(1);**

**}**

**).then(()=>{**

**console.log("Documento inserido com sucesso:");**

**}).catch(err=>{**

**console.log(err)**

**})**

Esse increment faz o valor que está no banco de dados ser incrementado.

**db.collection(TURMA).onSnapshot((snapshot) =>{**

**snapshot.forEach((doc) => {**

**let aluno = doc.data();**

**console.log(aluno);**

**})**

**})**

Esse código faz atualizar e computar todas as alterações que o documento recebe em tempo real, podendo fazer alterações no sistema quando x dado mudar.

**db.collection(TURMA).doc("AlunoNovo").update(**

**{**

**Faltas : firebase.firestore.FieldValue.delete();**

**}**

Serve para excluir a aba Faltas do nosso documento, servindo para qualquer outra coisa, para apagar um documento inteiro, ficaria assim:

**db.collection(TURMA).doc("AlunoNovo").delete().then(()=>{**

**console.log("Documento apagado com sucesso:");**

**}).catch(err=>{**

**console.log(err)**

**})**

Esse serve para excluir um documento inteiro do ID apresentado.

**let newUserEmail = "novoteste@teste.com";**

**let newUserPassword = "123abc";**

**let auth = firebase.auth()**

**auth.createUserWithEmailAndPassword(newUserEmail, newUserPassword)**

**.then(user=>{**

**console.log(user);**

**}).catch(error=>{**

**console.log(error);**

**})**

Serve para criar usuários no firebase de modo manual, sem formulários.

**function login(){**

**let UserEmail = "novoteste@teste.com";**

**let UserPassword = "123abc";**

**auth.signInWithEmailAndPassword(UserEmail, UserPassword)**

**.then(loggedUser =>{**

**console.log(auth.currentUser)**

**}).catch(error =>{**

**console.log(error);**

**})**

**}**

**login();**

Esse Código serve para armazenar o usuário logado.

**function login(){**

**let UserEmail = "novoteste@teste.com";**

**let UserPassword = "123abc";**

**auth.signInWithEmailAndPassword(UserEmail, UserPassword)**

**.then(loggedUser =>{**

**console.log(auth.currentUser)**

**}).catch(error =>{**

**console.log(error);**

**})**

**}**

**// login();**

**auth.onAuthStateChanged(user =>{**

**if(user){**

**console.log(user);**

**}else{**

**console.log("Ninguém logado")**

**}**

**})**

**function logout(){**

**auth.signOut().then(() =>{**

**console.log("Usuário foi deslogado")**

**}).catch(error =>{**

**console.log(error)**

**})**

**}**

Nesse código tem um sistema de login e de logout de cada usuário.