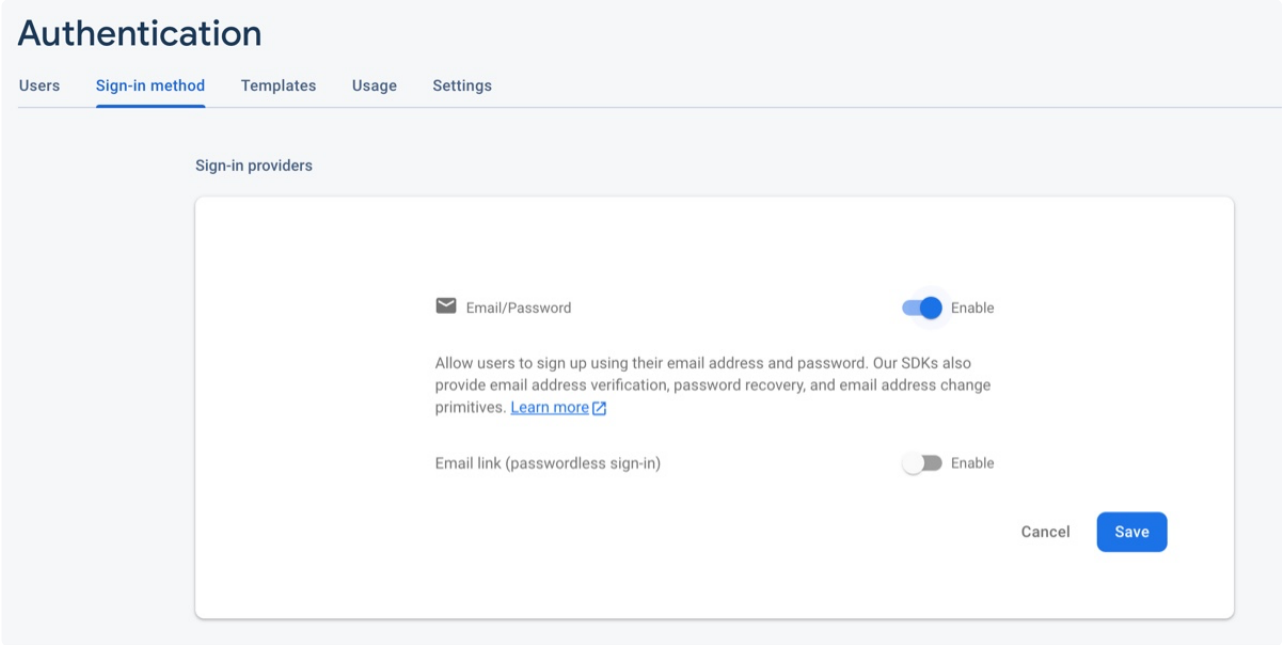
# Övningsuppgift 13 - Pokéfire - Autentisering

I denna övningen skall du implementera autentisering för din Firebase-applikation.

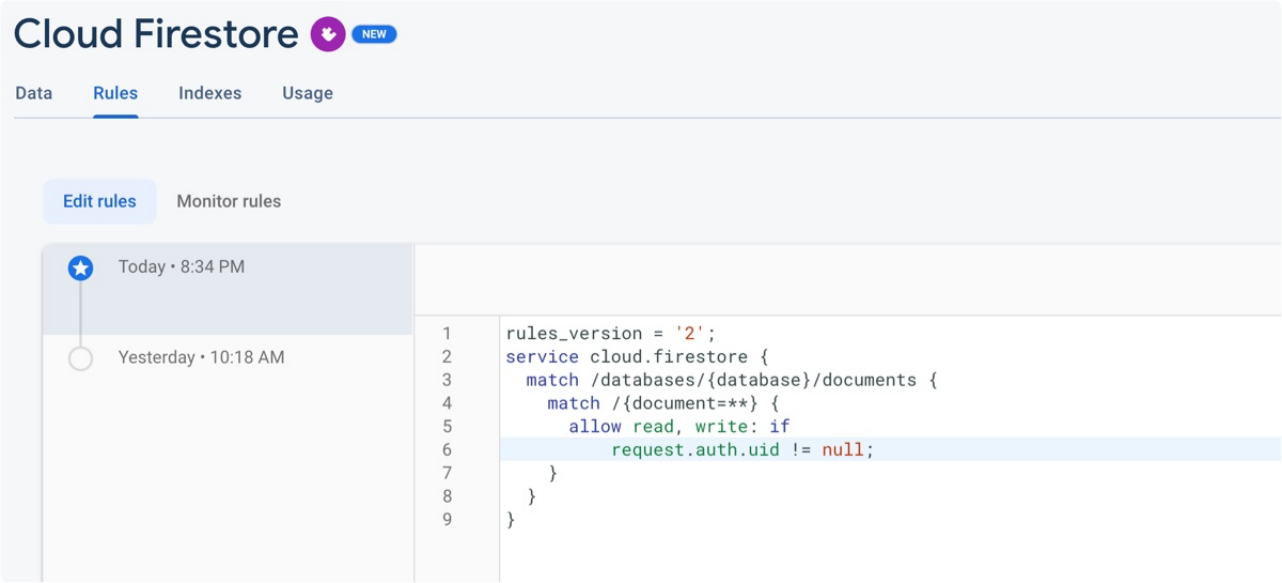
### Uppdatera ditt Firebase Projekt

1. Gå till console.firebase.google.com - navigera till firebase projektet som du skapade i Övning 12.
2. Gå till "Build" och sedan "Authentication". Välj "Getting Started" och lägg till Email/Password som Signin-metod. Vi kommer inte använda "Email link" metoden i denna övningsuppgiften.




1. Nu kan vi ändra reglerna för din firebase databas. Navigera till "Firestore Database". Välj "Rules" i toppen. Uppdatera din regel så att den nu kräver en inloggad användare:

```
rules_version = '2';
service cloud.firestore {
    match /databases/{database}/documents {
        match /{document=**} {
            allow read, write: if request.auth.uid != null;
        }
    }
}
```




4. Tryck på ”Publish”-knappen för att aktivera din förändring. Det kan ta någon minut innan ändringen går igenom.
5. Testa din app - du skall nu inte se någon av dina Pokemon eftersom du inte är inloggad.

**Inloggning och Skapande av användare**

1. Sakapa en ny komponent - kalla den för Signin. Vi kommer att använda samma component för både inloggning och registrering.
2. Skapa routes för din komponent.

   signin
   signin/:error

   Båda skall ta dig till din nya komponent.
3. Skapa en ny datatyp, User, den behöver två sträng-attribut; username och password.
4. Skapa ett formulär i din signin-komponent. Behöver att ha följande;

   Input för användarnamn (email, required)
   Input för lösenord (password, required)
   Knapp - "Skapa användare"
   Knapp - "Logga in"
5. Skapa en metod; onSignupClick(), som du kopplar till ”Skapa användare”-knappen. Den kan vara tom för tillfället.
6. Skapa en metod onLoginClick() som du kopplar till ”Logga in”-knappen. Den kan vara tom för tillfället.
7. Skapa ett nytt klassattribut; error, detta ska vara en sträng. Implementera OnInit och läs in parametern ”error” till den. Om du har ett värde i attributet skall det skrivas ut i en div på i komponenten.

**LoginService**

1. Skapa en ny service: LoginService. Glöm inte att lägga till den i [providers] i din app.modules.
2. Skapa ett klass-attribut som du kan kalla isLoggedIn - detta ska vara en boolean. Sätt den som false som standard och deklarera den som "private".
3. Skapa en metod som returnerar ”isLoggedIn”.
4. Implementera en metod som använder AngularFireAuth och metoden createUserWithEmailAndPassword för att skapa en ny användare. Vid fel så ska felmeddelandet skickas m.h.a. routern till signin-komponenten. Om det lyckas ska användaren anses som inloggad och ”isLoggedIn” ska sättas till sant och användaren ska navigeras till den tomma path:en.
5. Implementera en metod som använder AngularFireAuth och metoden signInWithEmailAndPassword. Vid fel skickas felmeddelandet m.h.a. routern tillbaka till sigin-komponenten. Om det lyckas ska isLoggedIn sättas till sant och användaren ska navigeras till den tomma path:en.

**Slutför din SignIn-komponenten**

1. Skapa en instans av LoginServicen m.h.a. dependency injection
2. Koppla ihop onSignupClick() med motsvarande metod i LoginServicen och onLoginClick() med motsvarade metoder i LoginServicen.
3. Du skall nu kunna skapa en ny användare. Den skall du också kunna logga in med efter att du har laddat om din applikation eftersom du har en databas med användare i firebase.
4. Om du vill radera användare kan du göra det från din firebase-konsoll.

**Vilkorlig rendering**

1. Låt din LoginService implementera CanActivate - interfacet. CanActivate()-metoden ska returnera samma värde som isLoggedIn.
2. Uppdatera dina routes så att canActivates kontrolleras på editPokemon-pathen.
3. Skapa vilkorlig visning av pokemon-listan m.h.a. ng-container/ng-template om användaren är inloggad eller ej. Du behöver lägga till LoginService genom dependency injection. Om användaren inte är inloggad skall hen få en uppmaning om att antingen logga in eller registrera sig (signup) för ett konto.
4. Du skall nu kunna logga in, se och arbeta med data. Laddar du om sidan behöver du logga in på nytt eller skapa ett nytt konto.

Du har nu implementerat en väldigt basal inloggnings-service. I en riktigt applikation behöver man behålla sin inloggning om sidan laddas om och även hantera en hel del andra saker. Det finns mycket mer funktionallitet i firebase som vi kan använda för att underlätta de olika processerna runt konto-administration.