

# Pokémon Cards API

## General Requirements:

1. Must support firebase Google Auth.
2. App must store user data and app data securely on Firebase.
3. Use Angular structure to implement routes, components, services, HTTP calls, and observables.
4. App should have zero critical bugs and up to three small bugs (a critical bug is defined as one that blocks a user from using or completing a major feature in your app and for which there is no workaround).
5. Commit and push your work to GitHub daily.
6. Publish web app on public website.

## App Specific Requirements:

The goal of the Pokémon Cards API is to create a matching game. It must have a minimum of four different pages, but you can add more pages/details than are listed here if you would like. It is up to you how you route between pages 2-4.

1. The first page (only page that order matters) that the user sees should be a login page to login with Firebase Google Auth.
2. The second page is for setting up a game. The user creating the game should have the ability to select the number of players (1-4), the number of matches to be found (should be evenly divided by number of players), and the Pokémon card set to play with. If more than one player is playing, the user creating the game should have to select the users who are playing from a list of people who have logged into the app previously (possibly save a users collection in Firebase...).
3. The third page is the matching page. At minimum, display the users who are playing (specify whose turn it is, all players should be playing on the same computer), how many total rounds have been played, how many matches each player has, how many matches remain, and the cards that belong to the set chosen in the game setup page. The number of cards displayed should be twice the amount of the number of matches selected in the game setup. If only one player is playing, give the user a specified amount of rounds to find all matches before losing. On initial game load, all cards should be face down; only when a user clicks a card will it show the detailed image. After each person's turn, if a match has not been made, flip both of the cards back over.
4. The fourth page is for displaying data about the logged in user. Display the total number of games played, the number of games won, the number games lost, a list of players you have lost to, and a list of players you have beaten. Consider using an Angular Material progress spinner (determinate) to help display data/percentages if needed.

I would recommend (optional) using the Angular Material 'mat-card' to help display content on the data page or throughout the application as needed.

("https://material.angular.io/components/card/overview").

**Setup (No API Key required):**

1. Go to "<https://pokemontcg.io/>" to test example API endpoints
2. Go to "<https://docs.pokemontcg.io/>" for API documentation

**Endpoints:**

The endpoints listed below are examples only and can be altered to the way you need them.

1. Get all sets - "<https://api.pokemontcg.io/v1/sets>"
2. Gets all cards by set -  
"<https://api.pokemontcg.io/v1/cards?setCode=SelectedSetCode>"

Additional info can be found at "[https://docs.pokemontcg.io/#api\\_v1cards\\_list](https://docs.pokemontcg.io/#api_v1cards_list)"