AirSports app - live tracking website interactions

App authentication is done using Google firebase. Users need not be registered with the system beforehand.

# Basic flow

## Initial start-up

If the app does not have existing user state (email address, token?), It should present a login screen where the user may enter an email address. The user should be encouraged to enter the same email address as they have used in competitions earlier (if any).

The app should perform email address login/validation through the firebase API with email authentication. The firebase project ID is “airsports-613ce”. I have configured “dynamic linking” in the firebase account for the domain links.airsports.no with the prefix /links. I think this is all information required to configure the use of dynamic links (redirecting to the app from the validation link in the email).

## Successful login

Successful login should leave the application with a unique user token. The backend has been extended to support firebase authentication using the django plug-in https://pypi.org/project/drf-firebase-auth/. According to the documentations it requires “JWT <token>” to be present in the Authorization header for authentication to be successful. This should give access to all “Profile actions” listed below.

The first step after successfully validating the email address is to fetch the user profile with the correct profile action (retrieve profile). An updated specification for the Person object (the data model behind the profile) can the found on the documentations page: https://airsports.no/docs/

An important parameter in the person object is “validated”. If the authorised request does not match an existing user, a new user and user profile is created. This user profile is created with validated=False to indicate that this is an incomplete profile. When retrieving the user profile, the app needs to prompt the user to complete the user profile and submit it (the app should set validated = True in the submitted data). If the retrieved user profile already has validated = True, then this is an existing user profile and there is no need to prompt the user to look at it.

All user profiles that remain with validated = False for some period of time (a day or two) will be deleted together with their associated users. This is to prevent the system from being overrun with bogus signups.

## Profile updating

The user should be able to update his/her profile. This includes changing first name and last name, picture, phone, country, and biography. The email address should not be changed, since this is linked to the user address, and the tracking ID should not be displayed at all. The “validated” field should be handled as described above.

# Profile actions

Below is a list of all actions related to profile management.

## Retrieve profile

Endpoint: /api/v1/userprofile/retrieve\_profile/

Method: GET

### Description

Requires token authentication. Returns a “Person” object with details about the person tied to the user account.

## Update profile

Endpoint: /api/v1/userprofile/update\_profile/

Method: PUT

### Description

Requires token authentication. Allows putting an entire ”Person” object. Unless the app will be handling all profile fields, I suggest that the partial update method is employed.

## Update partial profile

Endpoint: /api/v1/userprofile/partial\_update\_profile/

Method: PATCH

### Description

Requires token authentication. Allows patching a subset of a ”Person” object.

## Person object

{

|  |  |
| --- | --- |
| id | integer title: ID readOnly: true |
| country\_flag\_url | string title: Country flag url readOnly: true maxLength: 200 minLength: 1 |
| country | string title: CountryEnum: Array [ 249 ] |
| phone | string title: Phone minLength: 1 |
| first\_name\* | string title: First name maxLength: 200 minLength: 1 |
| last\_name\* | string title: Last name maxLength: 200 minLength: 1 |
| email\* | string($email) title: Email maxLength: 254 minLength: 1 |
| creation\_time | string($date-time) title: Creation time readOnly: true  Used to figure out when a not validated personal and user should be deleted |
| validated | boolean title: Validated  Usually true, but set to false for persons created automatically during app API login. This is used to signify that the user profile must be updated. If this remains false for more than a few days, the person object and corresponding user will be deleted from the system. This must therefore be set to True when submitting an updated profile from the app. |
| app\_tracking\_id | string title: App tracking id readOnly: true minLength: 1  An automatically generated tracking ID which is distributed to the tracking app |
| simulator\_tracking\_id | string title: Simulator tracking id readOnly: true minLength: 1  An automatically generated tracking ID which is distributed to the simulator integration. Persons or contestants identified by this field should not be displayed on the global map. |
| app\_aircraft\_registration | string title: App aircraft registration maxLength: 100  The display name of person positions on the global tracking map (should be an aircraft registration |
| picture | string($uri) title: Picture readOnly: true x-nullable: true |
| biography | string title: Biography |
|  |  |

}