Airsports Tracking API documentation

The Airsports tracking platform exposes a number of APIs. Most of these are for internal use, so this document details the endpoints and methods necessary for integrating Flight Contest (FC) with Airsports tracking. The details of the API itself is documented on <https://tracking.airsports.no/docs>. This document will not repeat that, but will give a short overview of how tasks will be accomplished. Attached is also JSON examples of data that can be used for the various endpoints.

# Creating a new contest

Before adding a new navigation task it is necessary to have a contest to which it can belong. This is created by POSTing to /api/v1/contests. There is not currently much information associated with a contest, except for its name, and its public status. A typical data block for the post would look like this:

{

“name”: “My big contest”,

“is\_public”:false

}

It is probably useful to leave contests and navigation tasks not public to begin with. This allows the task planner to finish everything in flight contest and synchronise everything to the tracking system, without exposing it to the public, yet. Note that this should be done with care since we do not want to publish track before all the contestants are in quarantine. The flag “is\_public” controls whether the object is visible for anonymous users of the tracking site. If a contest is not public, all associated navigation tasks will also be hidden. If a contest is public, then the visibility of associated tasks is controlled by their own is\_public flags.

# Creating a navigation task

Once a contest is created, use the special import endpoint to create a navigation task in one go. POST the data block to ​/api​/v1​/contests​/{contest\_pk}​/importnavigationtask​/ where contest\_pk is the ID of the object proceed in the response when creating the contest. An example data block is listed in the file importnavigationtask.json. Note that track\_file is a utf-8 base 64 encoded string with the contents of the route GPX file that FC exports.

Crews, aircraft, and teams are checked in the database to see if they already exist. If an aircraft of the same registration is already present, this object will be associated with the team that makes up the contestant. Similar for the crew, if the pilot and navigator matches, this object is reused. Otherwise new aircraft and crews are created. The same as the case for the team.if the crew and aircraft has already been a team earlier, the same team is reused here. This allows the system to maintain a log of the activity for the various aircraft and cruise in the system. This is not exploited at the moment, but could be useful later on. It is of course possible to manage these objects explicitly through appropriate rest endpoints, but these have not been exposed yet.

Successful posting should return an object similar to the one that was posted together with a 200 response.

## Modifying a navigation task

Once a navigation task is created, it is possible to change various aspects of it. It is allowable to change anything related to the contest as long as we are before the tracker\_start\_time for the affected contestants. This means that it is possible to change wind or tracker ID or airspeed, or anything similar for a contestant before tracker\_start\_time. This can be done using the PATCH method of /api/v1/contests/{contest\_pk}/navigationtasks/{navigationtask\_pk}/contestants/{id}/. All the IDs referenced in the URL are available in the response to the navigation task import method. It is not necessary to include the entire data block for the contestant, the data can contain only the affected items such as wind speed and direction:

{

“wind\_speed”:6,

“wind\_direction”: 120

}

## Publishing or hiding a task

Individual navigation task can be published or hidden using the appropriate /hide or /publish paths after navigationtask\_pk. This is similar to using PATCH to modify the is\_public property of the navigation task. Hide and publish provided as convenient shortcuts. Note that a navigation task is not published until it’s contest is published. The data for these methods is empty.

# Cleaning up

Both contests and tasks can be deleted. Deleting a contest removes all tasks and contestants below this.