Integrating with Air Sports Live Tracking for contest creation

This document describes a typical flow and required endpoints for integrating a third-party contest planning tool with air sports live tracking.

# Introduction

Air Sports Live Tracking (ASLT) is currently solely focused on team management and tracking and scoring flying tasks. Is well suited for making the flying sport more accessible by providing simple tools to create small navigation competitions. However, for the full contests with planning tasks, observation tasks, and so on, more powerful tools are required. Air Sports Live Tracking provides a REST API (https://airsports.no/docs/) that allows integration with third party planning tools for easy publishing of navigation tasks to the platform for live tracking and scoring.

# Proposed flow

Since Air Sports Live Tracking is primarily a visual tool, some effort has been put into gathering and handling information to make the online display interesting. Specifically, teams management is quite central to be able to present contestants in a pleasing way. It is possible to use the API to set up teams for a contest, we suggest that this is easier done through the Air Sports platform. The third party application can fetch the teams from the platform for use in the internal planning.

At a high level, the proposed flow is therefore as follows.

1. Create the contest in the ASLT platform
2. Use the ASLT tools to register teams (either by the users themselves or by the organiser)
3. Fetch the team list from ASLT through the REST API
4. Create the navigation task inside the third-party tool and push this to ASLT once it is complete
5. As aircraft arrive after flying the competition, fetch the aircraft track directly from ASLT for immediate scoring in the third-party tool
6. If required, import the track from the backup data logger to calculate the final score

Both the Air Sports Live Tracking app and any hardware trackers we recommend employ buffering of data when there is no network coverage. When network connection is restored, Boffa data will be transmitted to the ASLT platform, resulting in a complete track for the contestants. Position reports are approximately every second (within the limitations of the cell phone platform or hardware tracker) so the recorder track should be sufficient for most users. However, we recommend to always include a backup data logger in case of tracking failure or objections from the contestants.

## Setting up the contest (optional)

The contest is typically created through the Air Sports web interface, but this can also be achieved by POSTing to the endpoint /api/v1/contests/ according to the documentation.

## Creating a route

Before creating a navigation task the route to be used must have been built either through the route editor or posted through the API. Managing editable routes is done through /api/v1/editableroutes/. This allows you to POST, GET, PUT, PATCH, and DELETE the same routes that can be accessed through the web interface. The data format is the same as is used in the route editor, so any route added through the API is available in the route editor and vice versa.

## Creating a navigation task

Creating a navigation task is done through the endpoint POST /api/v1/contests/{contest\_pk}/navigationtasks/. This requires you to have the reference (id) of the route you wish to use in the navigation task, see the previous section. The type of navigation task is controlled by the selected original\_scorecard. The list of available scorecards can be retrieved from GET /api/v1/scorecards/, and use the shortcut\_name is to reference the desired scorecard. There are some parameters in the data that depends on the type of scorecard chosen. Specifically, any corridor related tasks (ANR, air sports race, or air sports challenge) require the parameters corridor\_width and rounded\_corners to be included.

## Contestants

Contestants can be added to or removed from the navigation task as desired. It is also possible to modify certain values, but this is quite restricted. It is usually safest to delete a contestant and recreate it if any changes are desired. A contestant is tied to a team, and it is required that this team is signed up to the contest. Signing up a team is best done by the individual contestants or through the web interface. This is to ensure that the correct persons and aircraft are included in the team. Teams are global for the air sports platform, so it is unfortunately not possible to modify these through the API.

### Fetching existing teams

When contestants have registered for the competition it is possible to retrieve the available teams. These can be used when creating new contestants.

The existing teams for a contest are fetched using the following endpoint: GET /api/v1/contests/{id}/teams/ where id is the id of the contest. It is not possible to modify any of the information here. The teams are fixed, and is possible to extract the airspeed entered by the team during registration for use when creating the contestant.

### Adding contestants

Managing contestants is done through /api/v1/contests/{contest\_pk}/navigationtasks/{navigationtask\_pk}/contestants/. To create a new contestant POST to this endpoint. If you have calculated your own gate times these can be entered into the gate\_times key in the POST data. Any waypoints that are not present in this data will be calculated internally. We recommend that you leave this field empty so that all gate times are calculated internally based on didn’t navigation task route, navigation task wind, and contestant airspeed.

Use the same endpoint to get the list of contestants, remove specific contestants or update them.

## Retrieving GPS tracks

After the flight, it is possible to retrieve the GPS track for the contestant. This is done using the following endpoint: GET /api/v1/contests/{contest\_pk}/navigationtasks/{navigationtask\_pk}/contestants/{id}/track/

## Posting GPS tracks

It is also possible to import GPS tracks into ASLT to allow previous tasks available in the third-party application to be pushed to ASLT and have it immediately score the contestants as if it were a real competition. This is very useful for debugging and comparisons between scoring algorithms. This is done by the following endpoint: POST ​/api​/v1​/contests​/{contest\_pk}​/navigationtasks​/{navigationtask\_pk}​/contestants​/{id}​/gpx\_track​/. The file format is a GPX file with a single “track” definition.

# Additional functionality

There is a plethora of API endpoint is available for controlling and modifying data. Everything listed in the previous section is the basic requirements for creating a full integration. In addition it is possible to integrate with the results service module of ASLT, but this documentation is TBD.