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.

## Fetching existing teams

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 should usually not be necessary to modify any of the information retrieved here, the purpose is to use this to reference the correct teams when pushing the navigation task to the ASLT platform.

## Pushing a navigation task

There are two ways of pushing a navigation task. Either the task can be pushed with reference to existing teams (fetched earlier), or with definitions of the teams themselves. The lateral show and is for the case where you opt to not use ASLT to define teams but rather want this to be handled by the third-party application. However, we suggest that you use the team functionality in ASLT and push the navigation task with reference to the existing teams.

### With existing team references

To push a navigation task with reference to the existing teams, use the endpoint POST /api/v1/contests/{contest\_pk}/importnavigationtaskteamid/. The content of the push is a list of contestants (with all information about timing, that is take off time, landing time, all gate times, et cetera), information about which scoring parameters should be overridden with provided values, basic information about the navigation task itself, and a route file. Each contestant references the id of an existing team.

For the route file we currently only support the GPX format that is generated by Flight Contest. The format of this file is quite straightforward and covers most of what is required for specifying a route. Adding support for additional files can be considered, but is not guaranteed.

### With new teams

To push navigation task which includes team definitions, use the endpoint POST /api/v1/contests/{contest\_pk}/importnavigationtask/. The contents of the push is exactly the same as for existing team references, with the small change that instead of including a team ID for each contestant, a full team definition is required.

## Modifying contestants

after pushing the initial task, it is possible to modify existing contestants. All of the parameters can be modified such as the score override, gate times, team information, and so on. This is achieved using the endpoint /api/v1/contests/{contest\_pk}/navigationtasks/{navigationtask\_pk}/contestants/{id}/.

## 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.