Tames.io

Documentation

2021

Contents

[Setup process 2](#_Toc83134072)

[Backend Setup 2](#_Toc83134073)

[Frontend Setup 2](#_Toc83134074)

[Running Tames.io Offline 2](#_Toc83134075)

[Using Tames.io Online 3](#_Toc83134076)

[Interface Overview 5](#_Toc83134077)

[Flight Chart 6](#_Toc83134078)

[Inputs 7](#_Toc83134079)

[Charts 9](#_Toc83134080)

[Heatmap 10](#_Toc83134081)

[Significant Attributes 10](#_Toc83134082)

[Future Work 12](#_Toc83134083)

[Updating the source libraries 12](#_Toc83134084)

[Multi-Dimensional Data Display 12](#_Toc83134085)

# Setup process

## Backend Setup

Tames.io backend is a Python-based script that takes the input parameters and returns calculations back to the frontend for graph data and ideal ranges.

1. Install Python 3.9 from <https://www.python.org/downloads/>
2. Cloning the backend repository at <https://github.com/timm/espy>.

By default, the backend is hosted on port 5000. This is then accessed at either <http://localhost:5000/> or <http://127.0.0.1:5000/>.

## Frontend Setup

Tames.io frontend is based on using Angular (written with Typescript), with the UI API being PrimeNG v7.2.6. The following steps are performed to initialize the frontend.

1. Install NPM at <https://nodejs.org/en/download/>
2. Clone the frontend repository at (insert link here)
3. Navigate to the frontend’s local directory, and install all needed dependencies with **npm install**
4. Execute the frontend using **ng serve**.

By default, the frontend is hosted on port 4200. You can then access the UI in your web browser with <http://localhost:4200/> or <http://127.0.0.1:4200/>.

## Running Tames.io Offline

To use Tames.io, you must simultaneously boot the backend and the frontend as in the previous setup steps. To connect the frontend to the backend, put the backend address into the Backend configuration panel once you boot the UI. Please refer to the **Interface Overview** section for a full description of the

UI.

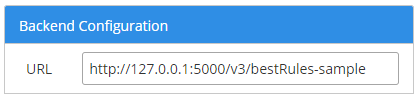


Figure 1: Where to enter the backend address

Note that there is an extension to the backend address. Currently, two extensions are provided

* **/v3/bestRules-sample** – The backend will return results immediately based on a sample JSON file. Used mainly for debugging.
* **/v3/bestRules** – The backend will use the provided inputs from the frontend, calculates needed values, and then returns it all as a JSON back to the frontend. This procedure may take up to a couple minutes long.

As of September 17, 2021, only **/v3/bestRules-samples** is confirmed to work due to updates in file naming conventions. This means **/v3/bestRules** may need to be updated so that the returned JSON file is valid.

## Using Tames.io Online

After running Tames.io locally, you can then tunnel the local addresses online using Ngrok <https://ngrok.com>). The application is free to use with a few limitations.

1. Once you download and extract the ngrok files
2. In a terminal navigate to where **ngrok.exe** is located.
3. Execute **ngrok http \*port number\* -host-header="localhost: \*port number\*".**

For example, if you wish to tunnel port 4200, you use **ngrok http 4200 -host-header="localhost: 4200"**. If successful, your terminal will look like the following.

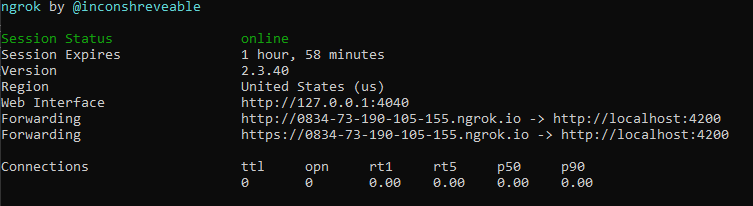


Figure 2: Ngrok tunnel for port 4200.

Once you establish a tunnel, you can then access your localhost port via the URLs highlighted above.

To fully run the backend and frontend online, you would need to run two separate instances if you’re only using the free version. One instance would tunnel the frontend, and the other would tunnel the backend. The general procedure is as followed.

1. Execute the Python backend
2. Run ngrok to tunnel the backend (default is port 5000)
3. Execute the frontend
4. Obtain the forwarded URL from the python backend and paste that URL into the Backend configuration panel on the front end.
5. Run a separate ngrok instance to tunnel the frontend (default is port 4200)
6. Obtain the forwarded URL from the frontend and distribute.

Please note that the free version of ngrok only opens a tunnel for up to two hours. Another limitation of the free version is that only one incoming connection is allowed at a time. This means that only one user at a time can access the forwarded URL, including yourself.

Also, please note that Government-issued computers may not allow access to Ngrok. If this is the case, then either you must run Tames.io fully locally on a personal device, or find an alternate tunneling solution.

# Interface Overview

The front end will look like the following image once executed



Figure 3: UI Layout

Currently, nine attributes are handled:

* Ascent Angle
* Descent Angle (Fast)
* Descent Angle (Slow)
* Cruise Speed (mph)
* Trip Distance (meters)
* Cruise Altitude (feet)
* Payload (lb)
* Wind Speed (m/s)
* Wind Direction

## Flight Chart

The flight chart is a display image to showcase different phases of flight ascent and descent. It is currently a static image and does not have any other functionality.

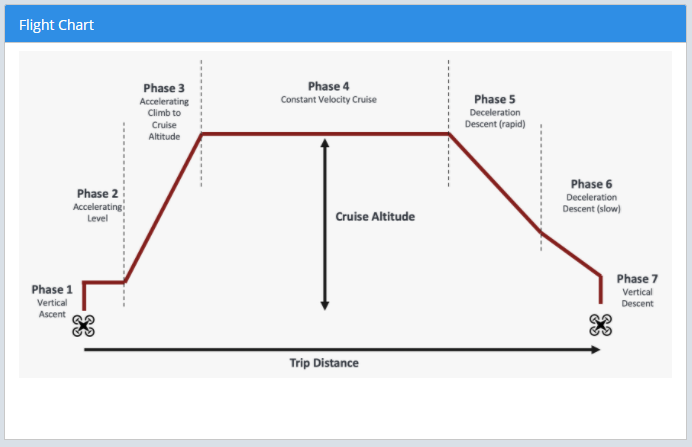


Figure : Flight Chart image

## Inputs

The parameters panel on the left side of the UI is used to set the attribute values that would be used for backend calculations. You can modify each attribute’s minimum (left) and maximum (right) values. Note that currently, there is no text filter in place so that the only accepted values are numeric values. The violation thresholds can also be modified by adjusting the single textbox values.

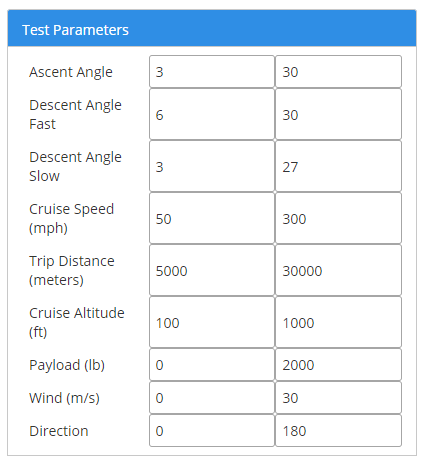


Figure : Modify test parameters

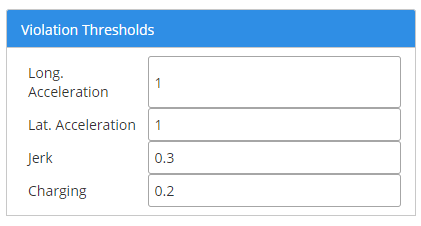


Figure : Modify violation thresholds

The Backend Configuration panel is used to designate the address used to access the backend. Either it can be a local address, or it can be a URL tunnel address. Please refer to the **Setup Process** section for more details. The Run button underneath will send the entered parameters to the backend when clicked.

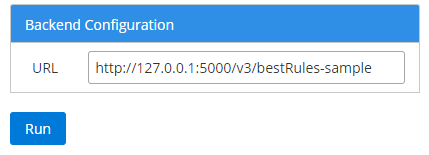


Figure : Backend configuration option

Currently, the Mode option is only a placeholder. It only has the Taxi option, and it doesn’t functionally affect anything.



Figure : Input mode option

## Charts

After submitting the inputs via Run, the backend will return the data needed to populate these charts. The attribute’s value makes up the X-axis, and P (non-violations) make up the Y-axis. There are currently nine charts to represent each different attribute used for the input.

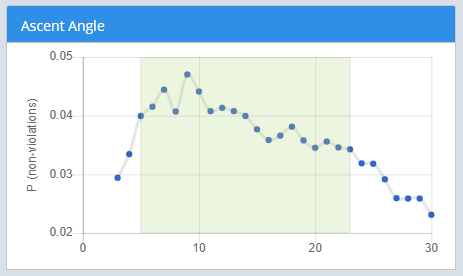


Figure : Example chart

The charts also include a highlighted region-of-interest based on the mode used. Changing the mode in Graph Display Options will change both the highlighted region and its color. So far, Warning Mode is red, Optimize Mode is green, and Suggestion Mode is yellow.

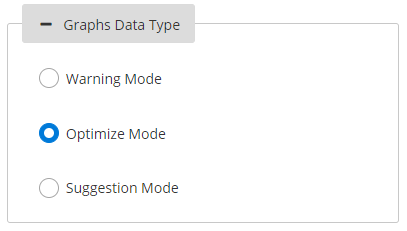


Figure : Changing Graph Modes

## Heatmap

The heatmap displays areas of interests based on the relationship between two attributes. Values are scored between 1.0 (green) and 0.0 (red). Highlighting your cursor over a cell will allow you to view its specific scoring value.

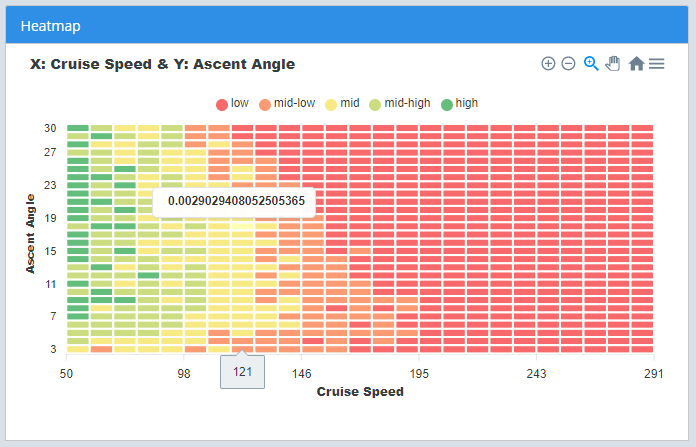


Figure : Heatmap with a highlighted value

To change the attribute used in the X or Y axis of the heatmap, use the dropdown menus located in the Graph Display Options panel.

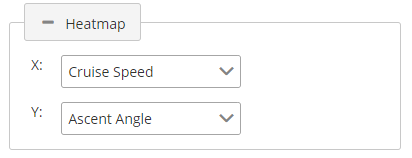


Figure : Heatmap axis dropdown menu

## Significant Attributes

The backend will also signify which attributes and values are the biggest contributors to the defined modes. These highlighted parameters are labelled as significant attributes and are visually marked in the UI after you hit Run. Currently, the significant attributes are marked with a (\*) in heatmap options, the corresponding graph’s title, and in the heatmap if present.

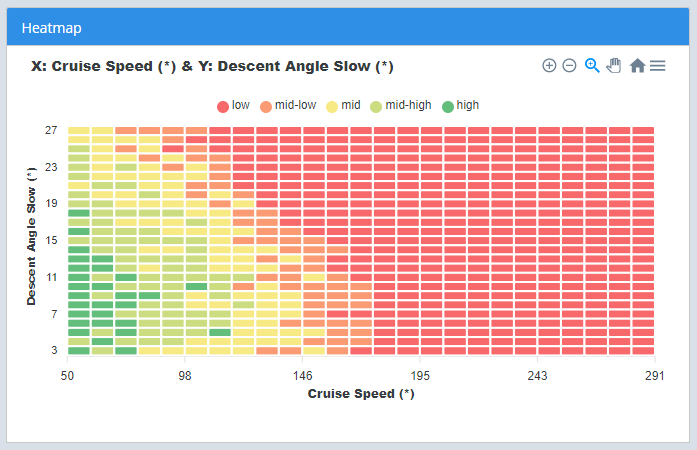


Figure : A heatmap with two significant attributes

A significant attributes panel will also display all returned significant attributes, as well as the returned min-max ranges for each one.

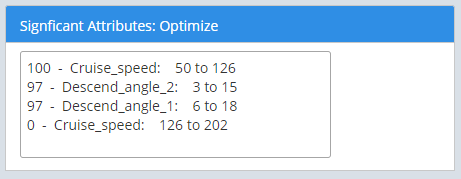


Figure : Example significant attributes in optimize mode

# Future Work

The following items are suggested as the next items to be worked on for the UI.

## Updating the source libraries

When the UI was developed, PrimeNG v7.2.6 was selected due to learning materials found. Unfortunately, this is not the latest version as of 2021. Ideally, we should strive to use the latest APIs if possible. Unfortunately, the newer PrimeNG versions, especially versions 10 or later, have updated changes that break older features. Simply using **npm-update** would break the current UI.

PrimeNG does have a GitHub page on the migration procedure, found at <https://github.com/primefaces/primeng/wiki/Migration-Guide>. Several elements are completely deprecated according to 10.0.0 notes. For those who continue work on the UI, it would be up to your decision whether to follow the migration procedure, or to switch to alternate APIs that may be better documented such as Bootstrap.

## Multi-Dimensional Data Display

Considering the large possible numbers of potential input parameters, multi-dimensional data display is planned next to help show potential relationships between them. One potential graph solution to investigate would be a parallel axis plot.

Unfortunately, it is recommended to not use HiPlot for this. Many resources for HiPlot are not in English and blocked for access on Government-issued computers. No potential APIs have been confirmed at this point.