Orbital Witness Technical Test

This document contains instructions for candidates to follow as part of the Orbital Witness Technical Assessment. We look forward to your submission.

Objective

At Orbital Witness, we deal with a variety of data. One of them is data on titles from the HM Land Registry. In this assessment, we have provided an analogue of what we usually deal with on a day-to-day basis.

User Story

As an API User, I want to filter and page through an extensive list of titles efficiently.

Your Task

1. You are provided with a JSON file containing a list of titles and their title number and title classification. This is the data for your database.
2. Create a list titles endpoint:
   `GET /api/titles`
   a. The API should only return the following fields:
      i. `id`
      ii. `title_number`
      iii. `title_class`
   b. The API should be filterable by `title_class`:
      `GET /api/titles?title_class=freehold`
   c. The API should be orderable by `id` and `title_number`:
      i. Sort by id in ascending order (default):
         `GET /api/titles?_sort=id`
      ii. Sort by id in descending order:
         `GET /api/titles?_sort=id&_order=desc`
      iii. For multiple fields, join the values using comma as the delimiter. For example, sort by `id` in descending order, then by `title_number` in ascending order:
         `GET /api/titles?_sort=id,title_number&_order=desc,asc`
   d. The API should be paginated:
      `GET /api/titles?_page=<page_id>&_limit=10`
3. Create an endpoint that returns a single title:
   `GET/api/titles/<id>`
   The API should return all available fields of a given title.
4. We will leave the API response structure up to you.
5. Optionally, create a basic frontend utilising the API that you have built.

Tips

Python and Web framework

We recommend using FastAPI running on the latest stable version of Python although not required.

Documentation

We would like to understand how you reached your solutions. We therefore highly appreciate good documentation practices like code comments, type hintings, well-structured readme, and proper commit messages in your Git history explaining each step.

Deployment

We would like to run and test your task locally. Please make sure all the dependencies are well-defined and installation instructions are available.

Timeline

We estimate this task should take no longer than 2-3 hours, but there is no limit so feel free to take it as far as you like. We don’t expect you to come up with a perfect solution. If you are not able to finish the whole test, please add a comment describing what you would have done given more time.

Submission

When you are ready, please publish your changes to a GitHub repository and share the link with us. Alternatively, send us a copy of the repository including the “.git” directory as a .zip file via e-mail.