

Technical interview: Fullstack Developer

We ask you to solve a technical case before the interview. The goal of this task is to provide a base for you to showcase your talents. There are some basic requirements, we recommend that you complete them at best. Furthermore, you will have some bonus tasks. Those are more flexible; either you complete our ideas of improvement, either you may be creative and propose something new to showcase at best your abilities.

Here is our technology stack. **We suggest that you use the same.** If you want to use other technologies, can you check with us before making a definitive choice?

- Vue 3, Nuxt 3 for the components design
- Tailwind CSS for styles
- Python (FastAPI) or Java (Quarkus) for the backend

We expect that you use the best practices of each tool you use, and that you focus on **writing clean code**.

Once you have completed the task, do not forget to send us the code (email, GitHub,...) **at least 3 working days before the interview**. You can add some visuals in case the website is not available online (but we will still need to see the results during the interview, so don't forget your computer!). Good luck!

Interactive maps

As an aerospace company, we work with a lot of geospatial data, that we display on interactive maps. In this exercise, we ask you to build a small interface to display the live location of the International Space Station.

Backend

The goal of the exercise is to create REST endpoints exposing the status of the International Space Station.

The status of the ISS can be retrieved from <https://api.wheretheiss.at/v1/satellites/25544>.
See <https://wheretheiss.at/w/developer> for more information.

- The above API should be called not more than once every 20s, up to you to decide how to put that restriction in place.
- The project should expose a GET resource at /iss/sun that returns a list of time windows during which the ISS was exposed to the sun, until present time.
- The project should expose a GET resource at /iss/position that returns the latitude and longitude of the ISS at present time
- For persistence, in-memory storage is fine for now, but nothing stops you from using a proper database if you want to.
- The code should be **unit tested** and a **diagram** should be produced to document the architecture of the solution.

Frontend

- Display a full screen map on the browser. You can use the free tier of Mapbox, OpenLayers...
- On the map, show the live position of the International Space Station using the API that you created above.
- Display the time windows when the satellite was illuminated by the sun. You can choose how to display the information (popup on hover, list on the side...) but it must be neat and clear.

Ideas for bonus tasks

- Add a set of endpoints to save and retrieve polygons. Hint: have a look at the GeoJSON and WKT formats.
- Allow the user to draw polygons on the map and save them permanently.
- If you add functionalities, update your tests and diagram accordingly.