The citybike Wien Importer

Step one: fetch and transform API data

Your task is to fetch data from citybike Wien, transform it into a different structure and to apply some filtering and sorting on it.

API Endpoint:
https://wegfinder.at/api/v1/stations

This endpoint returns a list of citybike stations.

Example of one station as provided by the API:

```
{
  "id": 108,
  "name": "Friedrich Schmidtplatz",
  "status": "aktiv",
  "description": "Ecke Lichtenfelsgasse U2 Station Rathaus",
  "boxes": 35,
  "free_boxes": 32,
  "free_bikes": 3,
  "longitude": 16.356100,
  "latitude": 48.210425,
  "internal_id": 1026
}
```

Transform this structure into the following:

```
{
  "id": 108,
  "name": "Friedrich Schmidtplatz",
  "active": // true if status == “aktiv“
  "description": "Ecke Lichtenfelsgasse U2 Station Rathaus",
  "boxes": 35,
  "free_boxes": 32,
  "free_bikes": 3,
  “free_ratio”: // free_boxes divided by boxes
  “coordinates”: // [longitude, latitude]
}
```

Filter out any stations that have no free bikes available.
Sort stations by the number of free bikes descending. If two stations have the same number of bikes, sort by name ascending.

At the end, output a list of the stations with all modifications applied.

Step Two: Fetch Addresses for coordinates

Add a property called “address” to each station. To fetch the address of a location, you need to call the following endpoint with the correct parameters:

```
https://api.i-mobility.at/routing/api/v1/nearby_address
```

This API endpoint expects latitude and longitude as GET parameters.

Example:

```
https://api.i-mobility.at/routing/api/v1/nearby_address?latitude=48.191&longitude=16.330
```

This will return something like this:

```
{
  "data": {
    "coordinate": {
      "latitude": 48.191466,
      "longitude": 16.330851
    },
    "type": "address",
    "name": "Mariahilfer Straße 189, 1150 Wien",
    "id": "bev:V2llbnx8MTE1MHx8TWFyaWFoaWxmZXIgU3RyYcOfZXx8MTg5"
  }
}
```

Use data.name as the value for the address property.