

Coding Task Overview: Cryptocurrency Portfolio Management System

Introduction:

Develop a backend system for a Cryptocurrency Portfolio Management application. This system will allow users to manage cryptocurrency holdings by tracking different cryptocurrencies in their portfolio and viewing the current market value of these holdings. The system will retrieve real-time cryptocurrency prices via external API calls and perform CRUD operations on the user's portfolio.

Functional Requirements:

1. Add New Portfolio Item:

- Implement an endpoint to receive data for a new portfolio item: cryptocurrency type (Bitcoin, Ethereum, Ripple, etc.), amount of coins, and date of purchase. All dates should be in UTC.
- Upon adding a new item, fetch the current market price of the specified cryptocurrency using any suitable 3rd party API (e.g., [Bitfinex API](#)) and calculate the current market value **in EUR**.
- Respond with the newly added portfolio item, including the calculated market value.

2. List Portfolio Items:

- Implement an endpoint to retrieve and display a list of all added portfolio items, including their current market values.

3. Delete Portfolio Item:

- Implement an endpoint that enables users to remove an item from their portfolio.

Non-functional Requirements:

- **Backend:** Java or Kotlin with Spring Boot. The service should utilize REST APIs, which produce and accept JSON.
- **Data layer:** You can decide any data storage layer solution you find suitable (i.e., relational database, NoSQL database or an in-memory data store).
- **Testing:** Include unit and integration tests. Aim to cover key functionalities and any critical paths in your application.
- **Observability:** Include API documentation for the implemented endpoints and any essential documentation you deem necessary.

Task Expectations:

- **Production-ready MVP:** Your application should be a production-ready MVP, which you would be comfortable deploying in a real-life production environment.

Optional Enhancements:

These are entirely optional but welcomed if you wish to further showcase your technical capabilities and have the time for it:

- **Authentication:** Implement a basic authentication layer to ensure that only authorized users can interact with their portfolio items.

- **Caching:** Implement a simple caching mechanism to store the current market values of currencies for a specified period of time.

Requirements for Submission:

- Provide a link to your source code repository. A detailed git commit history is appreciated, as it showcases your development process and adherence to version control best practices.
- Your README file should include:
 - Instructions for setting up, running the application, and executing tests.
 - A brief explanation of the application architecture and any important development decisions.
 - Guidance on accessing the API documentation you have provided.