

Technical Interview Project

Goal

The goal is to create a modern web application or API that displays and interacts with data from the supplied data set. The time constraint for this project is 3 days.

We will be using CoinCap API for our API/data source (<https://docs.coincap.io/>). Using this data, you will need to create **either**:

- A web app that allows users to view, sort, and filter asset data using the supplied parameters as well as viewing values of your assets
- A wrapper API, that abstracts the ability to interact with the asset class data points, as well as managing your assets

The goal here is **not** to build both, but rather give you a choice if you want to focus more on back-end functionality or front-end. **It is also encouraged here to note ideas to enhance the deliverable for further discussion with the team that you did not have time to explore.**

Web Application - Requirements

The more fluid and interactive the functionality the better. Condensed logic, state management layers, etc. are encouraged, but be mindful of the time constraint. The majority of the interface design and functional details are left up to you, but there are a few set requirements:

Functionality

- Must be able to view different assets and their prices
 - List view with a summary/detail view on each
 - Assets must be searchable or filterable
- Can view detail of each asset
 - Current pricing, timescale-based information, etc.
 - Charting / visual representation
- Ability to maintain a wallet
 - Manage (add, remove) the amount owned of an asset (say, Bitcoin) (fractional, whole)

- See value in totality equated to \$USD (say, 3.1415 BTC, 6.0221 ETH, ... ~\$101,441.99), as said value has changed over time
- Persist this between sessions

Technical

- **Must be executable in delivery form**
- Use one of the latest/relevant JavaScript libraries/frameworks (e.g. Angular, React, Vue)
- Use at the very least one dataset API provided above (<https://docs.coincap.io/>)
 - That said, there are many more similar to tie in, if you so choose <https://github.com/public-apis/public-apis#cryptocurrency>

Testing

- Should have an executable testbed, with some testing code coverage

API (Backend) - Requirements

Focus here on main pieces of net-new functionality - the ability to manage an account/wallet of different assets, see their values (USD), and have them persist. Authentication and the case of multiple users here should be considered. Persistence means (to local filestore, via database, etc.) are up to you, but be mindful of the time constraint.

Again, the majority of the interface design and functional details are left up to you, but there are a few set requirements:

Functionality

- Must clone the basic API functionalities:
 - List assets
 - Pagination
 - Filter and Sort ability
 - View asset details (volume, pricing, etc)
 - Convert asset value (current) to USD
- Ability for each user to add and remove assets from their "Wallet/Account"
 - Similar to the Web problem, allow users to supply how much X they own to get out value \$Y in USD
 - Can return **current** value they have in their account at one point related to USD value
 - Can return **net gain/loss over period of holding** value they have in their account at one point related to USD value

- This should have a layer of persistence

Technical

- **Must be executable in delivery form**
- Use latest/relevant backend languages, frameworks, libraries (we largely use NodeJS + Express and Python at EI)
- Adhere to a common API standard (REST, Graph)
- Use and model further actions off of the API provided above (<https://docs.coincap.io/>)

Testing

- Should have an executable testbed, with some testing code coverage

Delivery

Upon completion, the project must be pushed to a remote Git repository (Github, Gitlab, etc) and publically accessible for us to review. Additionally, it must include a README doc explaining how to get the project up and running, and any relevant documentation.