

Test task

The task is designed for 16 working hours. You can complete the task using any interface and tools convenient for you.

Task:

To complete the test task you should design and develop a data model that implements the organizational structure of the list of cryptocurrencies using the OOP concept. You should implement Business Entities such as Cryptocurrency and Status

Implement a test application, in which the main screen will have a Tab Bar, consisting of 2 lists(UITableView): "Current cryptocurrencies" and "Favorites". After clicking on the element of the list on any tab new page opens with detailed data on the cryptocurrency.

- Current list of cryptocurrencies page.

The cell must consist of 2 elements: Name and code (e.g. Bitcoin BTC)

- Favorites page.

The cell view should be exactly the same as at the first tab.

Both lists should use pagination.

The current list of cryptocurrencies can be obtained from the CoinMarketCap API:

Documentation:

<https://coinmarketcap.com/api/documentation/v1/#operation/getV1CryptocurrencyMap>

Endpoint: / v1 / cryptocurrency / map

API Key can be used as follows: **2592e201-7cb0-41b4-81d5-abacc60ac4ee** , or create your own at <https://coinmarketcap.com/api/>

Functionality:

1. Upon entering, the user will immediately see a list of the current cryptocurrencies.
2. By clicking on an item in the list, opens a screen with a detailed view of the item (platform, first_historical_data, last_historical_data, etc.)
3. On the screen of detailed cryptocurrency there should be a button that allows you to add cryptocurrency to favorites
4. By clicking on the tab with favorites, the list of previously added cryptocurrencies opens
5. By clicking on an item from your favorites, opens a screen with a detailed view
6. The detailed view should have a button that allows you to remove cryptocurrency from favorites
7. Tie updating of the list on the main screen to "pull to refresh".

It Will be a plus:

Cache data downloaded from the network.

To implement your task, use please FRP(Functional Reactive Programming) paradigm.

Would be a huge plus:

To write Unit tests responsible for services and responsible for business logic.

Thank you, we are waiting for an answer.