Surglogs Python Challenge

What to do?

- CRYPTO Trader API
- web API application
- CLI application
- feel free to use any third-party libraries and database of your choice (sharing your reasoning behind choosing them is welcome! ...for Python Flask is preferable ;)

What do we expect? & hints

- show us your best
- real-life production ready project
- don't forget to write tests! The scope and form is up to you ;)
- application's code should be kept in a public repository so that we can read it, pull it and build it ourselves
- don't forget to include README file or at least basic notes on application requirements/setup
- you have 2 weeks for implementation

Endpoint specification

ADD crypto exchange

- body should contain name (its presence should be validated) and currency in which total amount for exchange will be displayed
- response: new created object

POST /api/v1/crypto/exchanges

Deposit exchange

- body should contain only amount (Float) and currency (currency should be 3 letters shortcut, e.g. USD)
- response: success

POST /api/v1/crypto/exchanges/{exchange_id}

Update cryptocurrencies within exchange

- cryptocurrency object should contains minimal:
  name, 3 letter shortcut, symbol, favourites, exchange_id
- bulk request -> body should contain created/deleted or updated cryptocurrencies
  (when preffered new cryptocurrency and added/removed from favourites)
- name, 3 letter shortcut, symbol are all optional, but always one of them must be
  filled to complete others
- response: all cryptocurrencies belongs to exchange

PUT /api/v1/crypto/exchanges/{exchange_id}/currencie

Create trade

- body should contains amount, currency_in, currency_out after trade was made,
  converted amount should be automatically added to/subtracted from total
- exchange amount and actual exchange currency amount (depends on operation)

POST /api/v1/crypto/exchanges/{exchange_id}/trades

History of trades

- fetch list of all trades made within all exchanges
- allow filtering trades by date, specific exchange and search query (e.g. “ether”)
- paginated response

GET /api/v1/crypto/history?offset={offset}&limit={limit}&exchange_id={exchange_id}
&search={search}&date_from={date_from}&date_to={date_to}

Parameters

- amount: (Float) - amount, which we want to convert
- currency_in: (String) - 3 letters shortcut - e.g. USD or name/3 letters shortcut/symbol when cryptocurrency e.g. ethereum/ETH
- currency_out: (String) - 3 letters shortcut - e.g. USD or name/3 letters shortcut/symbol when cryptocurrency e.g. ethereum/ETH
- date_from: (String) - “2019-10-23T12:12:12Z”
- date_to: (String) - “2019-10-23T12:12:12Z”
- exchange_id: (Integer)
- search: (String)
- offset: (Integer)
- limit: (Integer)

Welcome Python Test

- you have 15 minutes on 7 questions, Good Luck!
- be prepared to code some small snippets, too
- **https://bit.ly/35kHNDt**

...if you have any questions, don’t hesitate to write a message on michal.korbel@surglogs.com!