Data:

Example of a trade data JSON object:
```
{
    "id":1,
    "type": "buy",
    "user_id": 23,
    "symbol": "ABX",
    "shares": 30,
    "price": 134,
    "timestamp": 1531522701000
}
```

Project Specifications:

The task is to implement a model for the trade object and the REST service that exposes the "/trades" endpoint, which allows for managing the collection of trade records in the following way:

GET request to "/trades":

- Returns a collection of all trades.
- The response code is 200, and the response body is an array of all trades objects ordered by their IDs in increasing order.
- Optionally accepts query parameters type and user_id, for example `/trades?type=buy&&user_id=122`. All these parameters are optional. In case they are present, only objects matching the parameters must be returned.

GET request to "/trades/<id>":

- Returns a trade with the given ID.
- If the matching trade exists, the response code is 200 and the response body is the matching trade object.
- If there is no trade with the given ID in the collection, the response code is 404 with the body having the text 'ID not found'.

POST request to "/trades":

- Creates a new trade.
- Expects a JSON trade object without an ID property as a body payload. If the value of 'shares' is out of accepted range [1, 100], or the type value is invalid (i.e. not 'buy' or 'sell'), the API must return error code 400. Besides those cases, you can assume that the given payload is always valid.
- Adds the given trade object to the collection of trades and assigns a unique integer id
- The response code is 201, and the response body is the created trade object.

PUT request to "/trades/<id>":

- Edits the trade object with the given ID.
- Expects a JSON trade object as payload, with the same validations as POST request. Payload should include the ID of the trade object to be edited.
- The response code is 204, and the response body is the edited trade object.

DELETE request to "/trades/<id>":

- Deletes the trade object with the given ID.
- The response code is 202, and the response body is empty.

Note:
You are expected to choose the ORM you want to use out of the following:
1. Sequelize with SQLITE
2. Mongoose with MongoDB