Clarity Backend Skills Test Documentation

Getting Started using the API

For the HTTP requests to the API, it is highly recommended to use httpie, which is a command line tool similar to curl, but with a cleaner and simpler user interface. All examples shown in this documentation use httpie to send requests. Link to httpie website for guide to installing: https://httpie.org/doc#installation

This API was written in Python and deployed using Chalice, which then deploys onto AWS Lambda with API Gateway and DynamoDB. All calls to this API are made using HTTP requests.

API URL:

```
https://lhlyv3ihej.execute-api.us-west-1.amazonaws.com/api/
```

```
X-Api-Key:YOUR_API_KEY
```

Note: You must append the above line as a header to every http request to access any of the API methods.

Example HTTP requests using httpie

list all nodes:

```
http https://lhlyv3ihej.execute-api.us-west-1.amazonaws.com/api/nodes X-Api-
Key:YOUR_API_KEY
```

create a new node:

```
http POST https://lhlyv3ihej.execute-api.us-west-1.amazonaws.com/api/nodes X-Api-
Key:YOUR_API_KEY
```

modify an existing node:

```
http PATCH https://lhlyv3ihej.execute-api.us-west-1.amazonaws.com/api/nodes/1 X-Api-
Key:YOUR_API_KEY node_id=1 latitude=39.29483949 longitude='40.38238298'
```

modify an existing project:

```
http PATCH https://lhlyv3ihej.execute-api.us-west-1.amazonaws.com/api/projects/1 X-Api-
Key:YOUR_API_KEY project_id=1 project_name='Berkeley Network' customer_name='City of
Berkeley'
```

assign a node to a project:

```
http PATCH https://lhlyv3ihej.execute-api.us-west-1.amazonaws.com/api/nodes/1/assign/1
X-Api-Key:YOUR_API_KEY node_id=1 project_id=1
```

In httpie, the **request headers** go right after the URL, formatted as name-value pairs separated by a colon (:). For the **request body**, the name-value attribute pairs are separated by the equals sign (=). In the documentation in the following page, the required and optional fields refer to name-value attribute pairs that are written in the request body.

When defining values for the request json body, quotes are not necessary for single words. However for values with a space in the middle (e.g. project_name:'Berkeley Network'), quotes are required.

All actions taken by the API should be safe, and the database should remain consistent. However, please be aware that if another user is modifying the same nodes and/or projects as you are, your modifications to a node or project may be overwritten or may overwrite their changes.

Deploying your own version of the API

To deploy on your own AWS account, download the code base and set up your AWS credentials to link to your account. Run `python database_setup.py`. Then, from the same folder where the `app.py` file is stored, run `chalice deploy`.

The necessary Python package installations required to run the code are shown in `required_packages.txt`. Please note that any dependences that these listed packages require should also be installed as well. A package manager such as conda may be useful for this step.

API Documentation

`GET /`
`GET /nodes`

Lists all existing nodes as an array of JSON objects.

`POST /nodes`
[required fields]: None
[optional fields]: None

Creates a new node with the following default attributes:

node_id:	auto-incrementing integer id
shipping_status:	'pending'
config_status:	'unconfigured'
assigned_to:	‘None’

`GET /nodes/{node_id}`

Returns information about a single node with the given node_id

`PATCH /nodes/{node_id}`
[required fields]: at least one optional field
[optional fields]: latitude, longitude, shipping_status, config_status

Modifies node information of node with given node_id. When modifying a node, you must include one of the optional fields in the request body.

`GET /projects`

Lists all existing projects as an array of JSON objects.

`POST /projects`
[required fields]: None
[optional fields]: None

Creates a project with the following default attributes:

project_id:	auto-incrementing integer id

`GET /projects/{project_id}`

Returns information about a single project with the given project_id

`GET /projects/{project_id}/nodes`

Lists all existing nodes (as an array of JSON objects) assigned to a single project with the given project_id.

`PATCH /projects/{project_id}`

[required fields]: at least one optional field

[optional fields]: project_name, customer_name, start_date, end_date

Modifies project information of project with given project_id. When modifying a project, you must have one of the optional fields in the request body.

`PATCH /nodes/{node_id}/assign/{project_id}`

[required fields]: None

[optional fields]: None

Assigns node with given node_id to a project with given project_id. The 'assigned_to' field of a node is updated with the project_id of the project that the node is now assigned to.

`PATCH /nodes/{node_id}/unassign`

[required fields]: None

[optional fields]: None

Unassigns a node with given node_id from a project. The 'assigned_to' field of the node is changed to ‘None’.