

DISTRIBUTED SYSTEMS HW3

OGOKE FELICITY,
ENACHE IULIA

5-17-2022

FUNCTIONS

Def `create_response_worker_queue(message)`: Function that creates response message that is sent from the worker to the message queue

Def `create_response_queue_worker(message)`: Function that creates response message that is sent from message queue to the worker

Def `create_response_masterdata_queue(message)`: Function that creates response message that is sent from master data service to the message queue.

Def `create_response_queue_masterdata(message)`: Function that creates response message that is sent from message queue to master data service.

Def `validate_user_permission_queue(username,token)`: checks if token is valid and if user is an administrator.

Def `validate_user_permission_push_pull(username,token)`: checks if token is valid and if user is a manager or administrator.

Def `validate_push_job(data)`: A function that ensures all requirements for pushing a job such as job_id, username, timestamp, status, date_range, assets are met.

Def `validate_pull_job(data)`: A function that ensures all requirements for pulling a job such as username and token met.

Def `validate_queue(data)`: a function that ensures all data such as username and token needed for when creating, deleting or listing a queue is provided.

Def `validate_push_result(data)`: checking if all data needed for pushing a result such as job id, username,token,timestamp,assets was provided.

Def `pull_job(job)`: function for pulling a job from a queue and deleting it from database.

Def `push_job(data)`: function for pushing a job into a queue.

Def `pull_result(data)`: function for pulling a result from a queue and deleting it from database.

Def `push_result(result)`: function for pushing a result into the results queue.

Def `create_message_job_db(data)`: function for adding job description to the database.

Def `create_message_result_db(data)`: function for saving our jobs results to database

Def `delete_message_job_db(job)`: function for deleting job from database

Def `delete_message_result_db(result)`: function for deleting message result from database

CLASSES

Class `Message_Job(Resource)`: Class for pushing(PUT) and pulling(GET) jobs into and out of results queue

- Def `put(self)`
- Def `get(self)`

Class `Message_Results(Resource)`: Class for pushing(PUT) and pulling(GET) results into and out of results queue

- Def `put(self)`
- Def `get(self)`

Class `Jobs_Queue(Resources)`: Class for creating, deleting and listing jobs in queue

- Def `post(self)`

- Def get(self)
- Def delete(self)

`Class Results_Queue(Resource)`: A Class for creating, deleting and listing results in queue

- Def post(self)
- Def get(self)
- Def delete(self)

MODELS.py FILE

We used the Model as a declarative class of the SQLAlchemy object, db, to create the two models.

1. `class Jobs_Queue(db.Model)` – builds the Job model for the database, containing 6 columns: id, username, timestamp, status, date_range and assets.
2. `class Result_Queue(db.Model)` - build result model for the database, containing the 3 columns Job ID, timestamp, assets.

CONFIGFILE.py FILE

We used the configparser module which can help us with creation of config files (.ini format).

After creating a config file, we put as configuration data the length of the queue. We chose 100 as the maximum length.