

edX-extensions

xqueue-watcher

package built by edX for establishing external grader servers. GitHub: [edx/xqueue-watcher](https://github.com/edx/xqueue-watcher).

Changes are made to build extra features such as start the grader with a config file, formatting the results, etc. New version Github: [ComeOnGetMe/xqueue-watcher](https://github.com/ComeOnGetMe/xqueue-watcher).

Usage: `python -m xqueue_watch -d [CONFIG_PATH]`

- This will look for a file that ends with `.json` in `[CONFIG_PATH/conf.d]`, which should contain queue name, authorization info, handler class and handler arguments. The queue name and authorization are used to log into the xqueue assigned to the course.
- To change course, only need to change the config file with corresponding xqueue info and course path.

container

Folder containing necessary files and requirements for the docker image.

`container/grade.py` Entrance for `docker_grader.py`. Usage: `python grade.py [SECTION_NAME] [PROBLEM_NAME] [SUBMISSION_URL]`

- This will invoke `container.simple_grader.SimpleGrader.grade([SECTION_NAME], [SUBMISSION_URL], [PROBLEM_NAME])` which does the grading.
- `SimpleGrader.grade()` work flow: [SimpleGrader.grade](#).
 - used Python package [BeautifulSoup](#) to get the feedback from the HTML file. (This could be done more elegantly using [nbgrader.api.Gradebook](#).)
- Building image: `docker build -t [REPO:TAG] .` Resulted Docker image: [zhipengyan:ucsdX](#).

DockerGrader

Python class defined in `edX-nbgrader/docker_grader.py` that links the Docker image and xqueue_watcher. It creates a temporary and (nearly) isolated environment for grading, runs the docker container, gathers the results and returns them to xqueue_watcher client. The place where most of the error handling happens.

- `DockerGrader(grader_root, course_dir, fork_per_item, logger_name)`
 - `grader_root`: the directory **inside the Docker image** where the copied course folder will be mounted. Default: `/app`, no need to change it. If it needs to be changed, make sure

that the Docker image uses the same directory to do the grading.

- `course_dir`: the path **in the grader server** that leads to the real course folder. For example, `edX-nbgrader/dummy-course`. The grader will copy the whole directory into some random-named folder in `/tmp` and mount the copy into the docker image.
- `fork_per_item` and `logger_name`: currently no need. Required by the parent class: `xqueue_watcher.grader`.
- `DockerGrader._handle_result(result_str, error_msg)`: post-process the message returned by the Docker.
 - If no error happend inside Docker, `result_str` should be a json string that could be assembled using either `eval(result_str)` or `json.loads(result_str)`.
 - If any error happened, the message will be printed.
 - The result dictionary that xqueue takes should contain `correct,score,tests,error` four keys. Or it will report wrong result on the submission page.