Adding new module on the AI4EOSC /iMagine marketplace

Valentin Kozlov
KIT

Co-funded by
the European Union

General steps

Check AI4OSdocs/develop-model#integrating-module-in-the-marketplace
Look into tips on AI4OSdocs/develop-model

(i) You may do only item "8."
But performing 1.-7. may help you to find problems **faster**

1. **check namings** (Jenkinsfile(s), metadata.json, README.md(s))
2. install **deep-app-schema-validator** and validate **metadata.json** (check: the JSON schema is OK)
3. **build** your **docker** image **locally** (check: it is built fine)
4. **start** that **docker** image and **enter inside**
5. **install tox** tool & **run** software **tests** (check: tests pass)
6. start **deepaas-run** (check: model loaded)
7. **access** the **API** and perform basic manual tests (check: app behaves as expected)
8. goto AI4OS catalog (deephdc/deep-oc) and edit MODULES.yml to add your module (PR)

=> platform operators proceed with checking and adding your module

1. Check namings

(i) Normally should be well set by the (cookiecutter) template

AI4 | eosc | Co-funded by the European Union

- (*currently*) two repositories:
  - <project_name> : repo with your AI model and API
  - UC-<original_github_repo>-deep-oc-<project_name> : docker and metadata
- both repositories have **Jenkinsfile** for the Jenkins CI/CD
- both repositories have **README.md** where <*CI/CD build status*> is referenced
- UC-<original_github_repo>-deep-oc-<project_name>/**Dockerfile** :
  - needs <*docker base name*>; git pulls <*project_name*>

BUT!

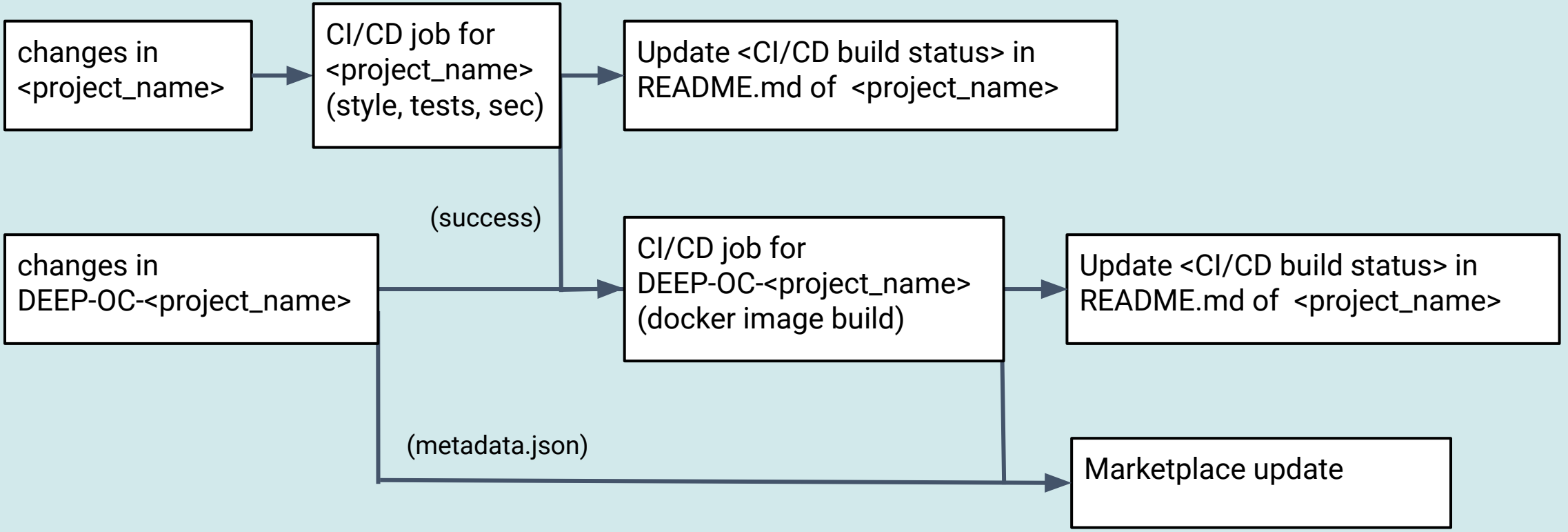
- UC-<original_github_repo>-deep-oc-<project_name>/**Jenkinsfile**
  - can **redefine** the <*docker base name*>
  - defines <*docker image name for the app* >

(i) because we may want to automatically build different docker images/tags

- UC-<original_github_repo>-deep-oc-<project_name>/**metadata.json** lists <*project_name*>, <*docker image name for the app*>, <*dataset*>, <*CI/CD build status*> as shown on the Hub/Marketplace

1. Check namings, CI/CD example

AI4 | eosc
Co-funded by the European Union



- fasterrcnn_pytorch_api : repo with your AI model and API
- deep-oc-fasterrcnn_pytorch_api : docker and metadata

2. deep-app-schema-validator

AI4 | eosc Co-funded by the European Union

- Install deep-app-schema-validator:

  ```
  pip install git+https://github.com/deephdc/schema4apps
  ```

- run it in UC-<original_github_repo>-deep-oc-<project_name> :

  ```
  deep-app-schema-validator metadata.json
  ```

3. build your docker image locally

AI4 | eosc | Co-funded by the European Union

- Simple way: goto UC-<original_github_repo>-deep-oc-<project_name>

```
docker build -t myhub/UC-<original_github_repo>-deep-oc-<project_name> .
```

(i) don't forget "." (dot) in the end!

- Options:
  - redefine tag for <docker base image>

```
docker build -t myhub/UC-<original_github_repo>-deep-oc-<project_name>:<specific tag> \
--build-args tag=2.14.0-gpu .
```

  - redefine branch for <project_name>:

```
docker build -t myhub/UC-<original_github_repo>-deep-oc-<project_name> \
--build-args branch=dev .
```

  - you can combine the above options together

4. start docker image locally

AI4 | eosc | Co-funded by the European Union

- Simplest:
  `docker run -ti myhub/UC-<original_github_repo>-deep-oc-<project_name> /bin/bash`
- Options:
  - Add specific port(s):
    ```
    docker run -ti -p 5000:5000 -p 8888:8888 \
    myhub/UC-<original_github_repo>-deep-oc-<project_name> /bin/bash
    ```
  - Use docker's host network driver (all host ports are redirected to the container):
    ```
    docker run -ti --network host \
    myhub/UC-<original_github_repo>-deep-oc-<project_name> /bin/bash
    ```
    (i) can be most practical
  - Mount host directory inside the container, e.g. with your "host-version" of the code
    ```
    docker run -ti -v $PWD/<project_name>:/srv/<project_name> \
    myhub/UC-<original_github_repo>-deep-oc-<project_name> /bin/bash
    ```
    (i) in this case, you can modify the code on the host and run it inside the container
  - you can combine the above options together

5-6. inside container

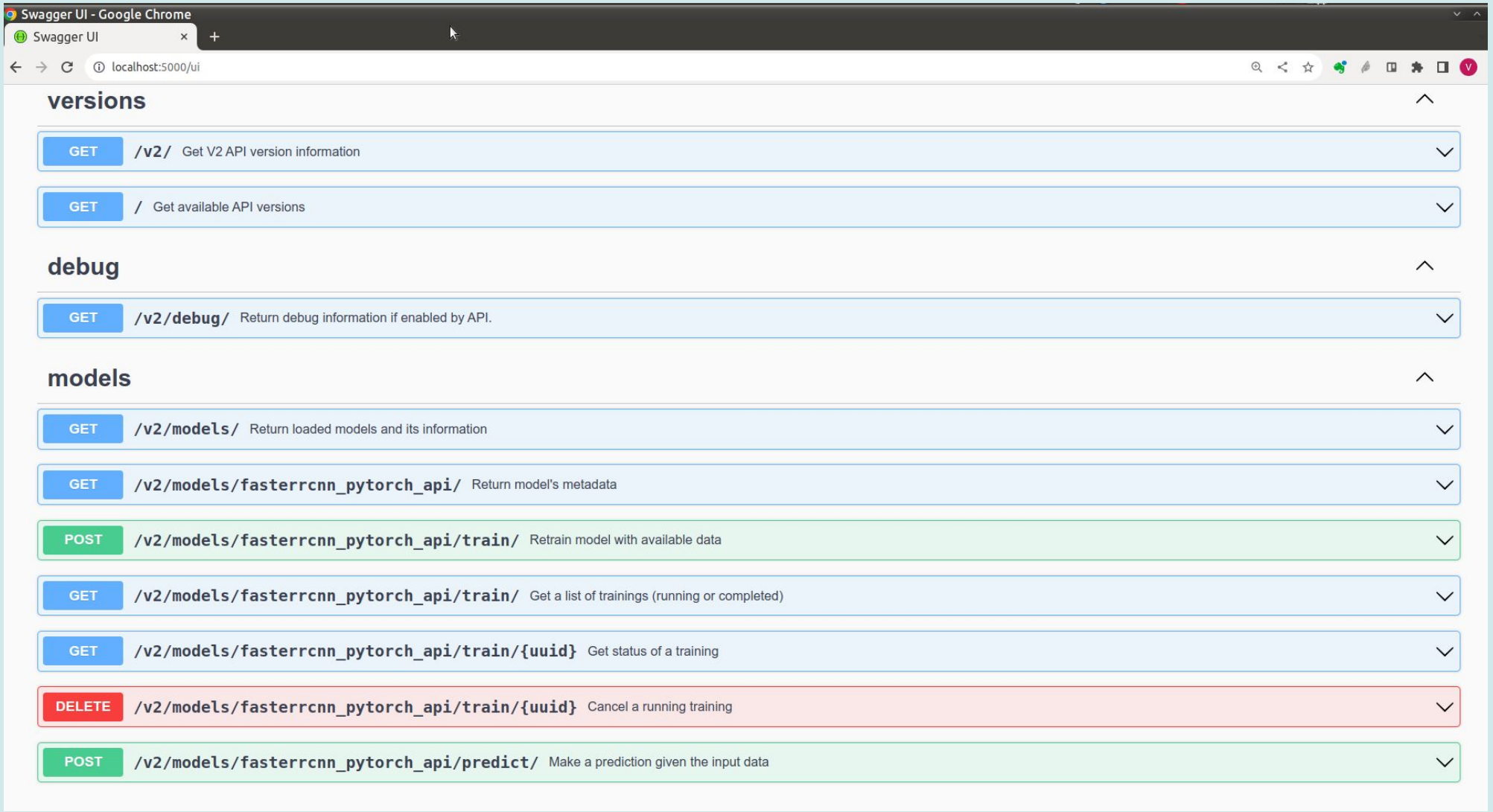
AI4 | eosc Co-funded by the European Union

5. Run tox:
   - Update Ubuntu: `apt-get update && apt-get upgrade -y`
   - (may need to) Upgrade pip: `pip3 install --upgrade pip`
   - Install tox : `pip3 install tox`
   - Goto directory **/srv/<project_name>** and run tox: `tox`
   - Options: you can also run a single test, like `tox -e qc.sty`

6. Run deepaas-run (if 1-5 is successful):
   - Full command: `deepaas-run --listen-ip 0.0.0.0 --listen-port 5000`
   - Or most of modern containers : `deep-start`
   - Check that the model is loaded and no Errors (Warnings may happen):
     `INFO deepaas.api [-] Serving loaded V2 models: ['**fasterrcnn_pytorch_api**']`

7. access swagger locally

AI4 | eosc
Co-funded by the European Union



8. Finally, PR in AI4OS hub

AI4 | eosc Co-funded by the European Union

Finally, when all is successful, go to AI4OS hub
(*moving* to github.com/ai4os-hub *from* deephdc/deep-oc)
and make PR (pull request) in order to add your module in MODULES.yml as :

- **module**: https://github.com/deephdc/**UC**-<github-user>-**DEEP-OC**-<project-name>

(see also this doc)

(i) with that *move*, some modifications will happen but general steps 1-8 will stay

AI4 | eosc

Co-funded by
the European Union


AI4EOSC

ai4eosc-po@listas.csic.es

ai4eosc.eu

Thank you! Any questions?

Author