

Как создать тестовое приложение для расчета атрибутов в системе DI

Версия документа 0.5 (21 Jun 2024)

1. Создаем job в БД

Для этого воспользуемся веб-интерфейсом (здесь и далее в примерах считается, что сервер доступен по адресу <http://localhost:9990>).

Открываем страницу swagger для API и находим запрос jobs/start: http://localhost:9990/docs#/jobs/start_job_jobs_start__post.

Заполняем поля:

- x-di-authorization (туда заносим токен из блокнота)
- Request body – здесь нужна структура с тремя ключами: program_name, job_description и project_id. job_description – это параметры запуска задания, соответствующие описанию в config.json для программы.

jobs

POST /jobs/start/ Start Job

Start new job. Valid token is required in x-di-authorization header.

Parameters

Name	Description
notify_app_starter	false
x-di-authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ

Request body required

application/json

```
{  "program_name": "SomeProgram",  "job_description": {"key2": 43},  "project_id": 60}
```

Execute **Clear**

Пример заполнения параметров запроса для приложения SomeProgram

После нажатия на кнопку Execute будет создано задание и вернется ID этого задания, который можно будет затем повторно использовать в параметрах командной строки запуска задания.

Следует учитывать, что если на сервере приложение уже развернуто, демон запуска заданий сделает попытку запуска задачи. Поэтому флажок `notify_app_starter` нужно выставить в положение `false`, при этом сервер не будет уведомлен о создании нового задания, и запуск задачи на сервере (в результате которого, возможно, будут созданы выходные данные) не произойдет.

Повторный запуск задания с одинаковыми параметрами приведет к попытке снова создать данные, эта операция закончится неудачей. Поэтому созданные данные нужно будет предварительно удалить (см. далее).

В результате выполнения запроса вернется идентификатор задания (в дальнейшем примере – 19636676).

2. Запуск задания из командной строки

Перейдите в каталог PyApps (в нем должен быть каталог отлаживаемого приложения) и запустите приложение с командной строки через скрипт запуска `app.sh`, например:

```
(venv) vladimirs:PyApps efremov$ ./Example03/app.sh --job 19636676
--project 3344276 --token eyJhbG....LpA2Q -u http://localhost --port
9990
```

Следует учесть, что библиотека `di_lib` будет искаться в текущем каталоге (т.е. в PyApps).

3. Чистка выходных данных

Поскольку данные с одинаковыми именами создавать запрещено (ограничение действует независимо для 2Д и 3Д данных), после тестового запуска для повторного запуска приходится удалять созданные данные.

Для этого можно воспользоваться тем, что (в текущей схеме именования данных счетного задания) поле `name2` начинается с имени программы (на

самом деле, с имени класса который унаследован от DiAppSeismic3D или DiAppSeismic3D2D), а поле name начинается с имени, заданного в параметрах задания. Можно найти идентификаторы таких данных и удалить данные при помощи метода delete_line_by_id или delete_cube_by_id объекта Session.

```
[11]: s.delete_line_by_id(p1.line_id)

[15]: tmp_l = s.list_lines()
      tmp_ids= [(i["id"], i["name"], i["name2"]) for i in tmp_l if i["name2"].startswith("Example03")]
      tmp_ids

[15]: [(19636685, 'new name (070124)', 'Example03 (Example03 1)'),
      (19636686, 'new name (070124)', 'Example03 (Example03 2)'),
      (19636684, 'new name (070122)', 'Example03 (Example03 2)'),
      (19636683, 'new name (070122)', 'Example03 (Example03 1)')]

[17]: tmp_l = s.list_cubes()
      tmp_ids= [(i["id"], i["name"], i["name2"]) for i in tmp_l if i["name2"].startswith("Example03")]
      tmp_ids

[17]: [(19636674, 'new name', 'Example03 (Example03 2)'),
      (19636673, 'new name', 'Example03 (Example03 1)')]
```

Пример запроса поиска идентификаторов данных, созданных программой.

NOTE! При удалении следует проявлять осторожность, поскольку отменить операцию удаления сейчас невозможно.