# Документация

Использование сервиса осуществляется посредством HTTP запросов к фиксированному URL сервера. Далее приведено описание возможных запросов.

## Данные

* 1. Формат данных

DataFormat задает формат данных, хранящихся в датасете. Также используется при создании задачи (Task) или имплементации (ProgramImplementation) для определения формата входных и выходных данных.

Работа с DataFormat осуществляется с помощью запросов по адресу /data/formats. Доступные запросы:

|  |  |  |
| --- | --- | --- |
| Запрос | Адрес | Комментарий |
| GET | /data/formats | Список всех доступных объектов DataFormat |
| GET | /data/formats/<id> | Объект DataFormat с Id=id |
| POST | /data/formats | Добавление нового объекта |
| DELETE | /data/formats/<id> | Удаление объекта DataFormat с Id=id |

Вывод запроса GET для одного формата:

{

"Id": 3,

"FormatName": "csv",

"FormatType": "File",

"FormatSchema": null,

"FormatExample": null,

"BaseEntityId": 3

}

При добавлении объекта необходимо задать следующие поля:

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Комментарий | Опции |
| FormatName | STRING | Формат хранения данных |  |
| FormatType | STRING | Enum[File, Table] |  |
| FormatSchema | STRING | Описание schema | Optional |
| Example | STRING | Пример формата данных | Optional |

Пример:

curl <server\_url>/data/formats -X POST -d “Format=CSV”

* 1. Датасет

DataSet определяет расположение файла с данными.

Работа с DataSet осуществляется с помощью запросов по адресу /data/sets. Доступные запросы:

|  |  |  |
| --- | --- | --- |
| Запрос | Адрес | Комментарий |
| GET | /data/sets | Список всех доступных объектов DataSet |
| GET | /data/sets/<id> | Объект DataSet с Id=id |
| POST | /data/sets | Добавление нового объекта |
| DELETE | /data/sets/<id> | Удаление объекта DataSet с Id=id |

Вывод запроса GET для одного датасета:

{

"Id": 10,

"DataFormatId": 3,

"Content": "<schema>://[host]/path",

"BaseEntityId": 26

}

При добавлении объекта необходимо задать следующие поля:

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Комментарий | Опции |
| DataFormatId | INT | Id объекта DataFormat, тип формата хранения данных |  |
| Content | STRING | Ссылка на данные. Connection string в случае таблицы, URL в случае файла. |  |

Пример:

curl <server\_url>/data/sets -X POST -d “DataFormatId=5&Content=<schema>://[host]/path”

### Эксперимент

* 1. Имплементация

Имплементация модели задается в объекте ProgramImplementation.

Работа с ProgramImplementation осуществляется с помощью запросов по адресу /implementations. Доступные запросы:

|  |  |  |
| --- | --- | --- |
| Запрос | Адрес | Комментарий |
| GET | /implementations | Список всех доступных объектов ProgramImplementation |
| GET | /implementations/<id> | Объект ProgramImplementation с Id=id |
| POST | /implementations | Добавление нового объекта ProgramImplementation |
| DELETE | /implementations/<id> | Удаление объекта ProgramImplementation с Id=id |

Вывод запроса GET для одной имплементации:

{

"Id": 4,

"Status": "Working",

"OS": "Win",

"EnvironmentId": 1,

"TaskId": 2,

"InputFormat": 3,

"OutputFormat": 3,

"CommandLineArgs": null,

"FileURL": "<schema>://<host>/<path>",

"FilePath": "<local\_path\_to\_file>",

"DataProcessing": 0,

"ProgramType": "Exec”,

"PythonRequirements": null,

"BaseEntityId": 34

"DownloadSuccess": True

}

При добавлении объекта необходимо задать следующие поля:

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Комментарий | Опции |
| OS | STRING | Enum[Win, Linux, MacOS] |  |
| EnvironmentId | INT | Id объекта ExperimentEnvironment |  |
| TaskId | INT | Id объекта Task |  |
| InputFormat | INT | Id объекта DataFormat, формат входных данных |  |
| OutputFormat | INT | Id объекта DataFormat, формат выходных данных |  |
| CommandLineArgs | STRING | аргументы командной строки для запуска модели | Optional |
| FileURL | STRING | URL файла имплементации |  |
| DataProcessing | INT | Способ скачивания входных данных. 0 – скачивание всех данных, число 1-99 – процент от общего количества данных, скачиваемых в одном батче |  |
| ProgramType | STRING | Enum[Python, Exec] – тип программы |  |
| PythonRequirements | STRING | URL файла requirements.txt, если тип программы является Python | Optional |

Пример:

curl <server\_url>/implementations -X POST -d “OS=Win&EnvironmentId=1&TaskId=5&InputFormat=2&OutputFormat=2%DataProcessing=0&ProgramType=Exec&Blob= file://[host]/path”

* 1. Требования к окружению

Требования к окружению для запуска эксперимента задаются в объекте ExperimentEnvironment. Они описывают минимальные условия, необходимые для успешного запуска эксперимента.

Работа с ExperimentEnvironment осуществляется с помощью запросов по адресу /experiment/enivornments. Доступные запросы:

|  |  |  |
| --- | --- | --- |
| Запрос | Адрес | Комментарий |
| GET | /experiment/environments | Список всех доступных объектов ExperimentEnvironment |
| GET | /experiment/environments /<id> | Объект ExperimentEnvironment с Id=id |
| POST | /experiment/environments | Добавление нового объекта ExperimentEnvironment |
| DELETE | /experiment/environments /<id> | Удаление объекта ExperimentEnvironment с Id=id |

Вывод запроса GET для одного объекта:

{

"Id": 1,

"OS": "Win",

"Processor": "x86",

"Memory": 500,

"HDD": 1000,

"BaseEntityId": 29

}

При добавлении объекта необходимо задать следующие поля:

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Комментарий | Опции |
| OS | STRING | Enum[Win, Linux, MacOS] |  |
| Processor | STRING | тип процессора |  |
| Memory | INT | необходимый объем оперативной памяти в мегабайтах |  |
| HDD | INT | необходимый объем свободного места на диске в мегабайтах |  |

Пример:

curl <server\_url>/experiment/environments -X POST -d “OS=Win&Processor=x86&Memory=200&HDD=500”

* 1. Задача (Task)

Задача, которую выполняет эксперимент, задается в объекте Task.

Работа с Task осуществляется с помощью запросов по адресу /tasks.

Доступные запросы:

|  |  |  |
| --- | --- | --- |
| Запрос | Адрес | Комментарий |
| GET | /tasks | Список всех доступных объектов Task |
| GET | /tasks/<id> | Объект Task с Id=id |
| POST | /tasks | Добавление нового объекта Task |
| DELETE | /tasks/<id> | Удаление объекта Task с Id=id |

Вывод запроса GET для одного объекта:

{

"Id": 2,

"Title": "Test task",

"Input": "File",

"DefaultInputFormat": 3,

"Output": "File",

"DefaultOutputFormat": 3,

"ResultQuality": "none",

"BaseEntityId": 20

}

При добавлении объекта необходимо задать следующие поля:

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Комментарий | Опции |
| Title | STRING | название, описывающее задачу |  |
| Input | STRING | Вид входных данных, Enum[Table, File] |  |
| DefaultInputFormat | INT | Id объекта DataFormat, формат входных данных |  |
| Output | STRING | Вид выходных данных, Enum[Table, File] |  |
| DefaultOutputFormat | INT | Id объекта DataFormat, формат выходных данных |  |
| ResultQuality | STRING | метрика, используемая для определения качества |  |

* 1. Эксперимент

Эксперимент задается в объекте Experiment.

Работа с Experiment осуществляется с помощью запросов по адресу /experiment/experiments.

Доступные запросы:

|  |  |  |
| --- | --- | --- |
| Запрос | Адрес | Комментарий |
| GET | /experiment/experiments | Список всех доступных объектов Experiment |
| GET | /experiment/experiments/<id> | Объект Experiment с Id=id |
| POST | /experiment/experiments | Добавление нового объекта Experiment |

Вывод запроса GET для одного объекта:

{

"Id": 1,

"Title": "Test Experiment",

"TitleShort": "Test Exp",

"Comment": null,

"BaseLine": null,

"RefRes": null,

"EnvironmentId": 1,

"TaskId": 2,

"BaseEntityId": 35,

"DataSets": [

10,

11

]

}

При добавлении объекта необходимо задать следующие поля:

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Комментарий | Опции |
| Title | STRING | название эксперимента, описывающее его |  |
| TitleShort | STRING | краткое название эксперимента |  |
| Comment | STRING | комментарий |  |
| DataSets | STRING  (INT,INT,…) | Список чисел, разделенных запятыми, список Id датасетов |  |
| BaseLine | STRING | результат бейзлайна | Optional |
| RefRes | STRING | референсный результат | Optional |
| EnvironmentId | INT | Id объекта ExperimentEnvironment |  |
| TaskId | INT | Id объекта Task |  |

## Запуск эксперимента

Перед запуском эксперимента необходимо создать объект Experiment и ProgramImplementation. Запуск производится с помощью POST запроса по адресу /experiment/run.

|  |  |  |
| --- | --- | --- |
| Запрос | Адрес | Комментарий |
| POST | /experiment/run | Запустить эксперимент, используя выбранную имплементацию |

Необходимо задать следующие поля:

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Комментарий | Опции |
| ExperimentId | INT | Id объекта Experiment |  |
| ProgramImplementationId | INT | Id объекта ProgramImplementation |  |

# Этапы создания и запуска эксперимента

### Загрузка данных

### Создание формата данных

Создать новый объект DataFormat, описывающий формат, в котором хранятся данные. Тип формата может быть файлом или таблицей (File или Table).

### Создание датасета

Создать новый объект DataSet, описывающий используемый датасет. В качестве DataFormatId используется Id созданного на прошлом шаге объекта DataFormat. Если тип формата – файл, то в поле Content необходимо указать URL для скачивания файла. Если тип формата – таблица, то в поле Content необходимо указать Connection String SQL таблицы.

### Загрузка файла имплементации

Создать объект ProgramImplementation, описывающий исполняемый файл программы. В качестве входного формата используется Id созданного объекта DataFormat.

### Аыва