Сервер хранения параметров пользователя

Параметры - именованные строки (str) и числа (int). Привязаны к юзеру. Имена уникальны в пределах типа и юзера.

Параметры хранятся в БД. В БД также должен храниться список юзеров (задается отдельно).

Сервер должен обеспечивать операции через REST-API:

* Установить параметр (имя юзера, имя параметра, тип, значение) – если параметр отсутствует в БД, то он добавляется, если есть – заменяется. Параметры устанавливаются только для существующих юзеров.

POST /api/parameters/<user>/<name>/<type> в данных – значение параметра. Если установить параметр не получилось (несоответствие типа или некорректный юзер) – ошибка.

* Получить параметр (имя юзера, имя параметра, тип) – если параметр не был задан, ошибка. Если не задан тип – выдаются параметры всех типов с данным именем.

GET /api/parameters/<user>/<имя параметра>/<тип> возвращает JSON [{"Name":"<имя параметра>", "Type":"<тип параметра>", "Value":"<значение параметра>"}] При некорректном юзере – ошибка. При отсутствии подходящего параметра – пустой список.

* Получить все параметры юзера (имя юзера)

GET /api/parameters/<user> возврат аналогичен предыдущему

* Установка параметров через JSON-API (имя юзера)

POST /api/<user> в данных – описание установки параметров, заданное в следующем формате:

{"Query":

[

{

"Operation":"SetParam",

"Name":"<имя параметра>",

"Type":"<тип параметра>",

"Value":"<значение параметра>"

}

]

}

Может быть установлено несколько параметров.

Ответ:

{"Result":

[

{

"Operation":"SetParam",

"Name":"<имя параметра>",

"Type":"<тип параметра>",

"Status":"OK|ERROR"

}

]

}

Рекомендуемый стек: Python (flask, sqlalchemy), SQLite