

## Задание для Волга - IT

Необходимо разработать скрипт импорта большого количества товаров ~2 000 000 шт. из csv-файла. Так же нужно написать сервис из одного метода [http://localhost/getProduct/<product\\_id>](http://localhost/getProduct/<product_id>), который будет возвращать продукт по идентификатору <product\_id>. Авторизацию на сервере реализовывать не нужно.

Ссылка на пример загружаемого файла: (ссылка будет добавлена позднее).

Входной файл имеет следующую структуру:

- Product id - идентификатор продукта в базе данных;
- Product code - код продукта;
- Language - язык, на котором описаны поля из этой строки;
- Category - категория товара. Категории товаров представлены в виде дерева. Если товар находится в категории "Электроника" > "Фото и видео" > "Цифровые фотоаппараты", то в этом поле будет следующая запись "Электроника///Фото и видео ///Цифровые фотоаппараты";
- List price - рекомендованная цена;
- Price - цена;
- Quantity - количество на складе;
- Product name - название товара;
- Description - описание товара;
- SEO name - имя для генерации ЧПУ;
- Short description - краткое описание;
- Status - статус продукта. 'A' - активный, 'D' - не активный, "
- Vendor - название продавца (магазина);
- Features - опции товара. Описываются в следующем формате: "<название\_опции>: <E|S|N|C>[<опция>]";. E,S,N,C - тип опции: E (enum) - объединение, S (string) - строка, N (number) - число, C (checkbox) - флажок (возможные значения Y или N). Опции разделяются символом ",";
- возможно присутствие других полей, которые можно игнорировать.

Сервис, который возвращает товар по ID, должен предоставлять данные в формате json. Пример:

```
{
  "product_id": 667779,
  "product_code": "01010101",
  "language": "ru",
  "category": [
    " ",
    " ",
    ""
  ],
  "list_price": "0.00",
  "price": 21.6,
  "quantity": 50877,
  "name": " «»",
  "description": "",
  "seo_name": "stakan",
  "short_description": "",
  "status": "A",
  "vendor": "",
  "product_features": {
    "": "",
    "": "",
    " ": "Y"
  }
}
```

## Общие требования

Работа должна сдаваться в виде ссылки на репозиторий в Github. В репозитории должен присутствовать файл [Readmy.md](#), в котором есть инструкция, как запускать скрипт и как запускать сервис.

Скрипт и сервис должны быть написаны на языке PHP версии не ниже 7.0. Можно использовать любые фреймвоки и библиотеки. Можно использовать любую базу данных. Важно, чтобы после перезагрузки базы данных они не терялись.

Скрипт должен информировать об успешном завершении работы сообщением в консоли.

Для реализации сервиса желательно использовать встроенный сервер PHP (`php -S localhost:8000`).

## Критерии оценки

Во время проверки будем запускать скрипты на файле 2 000 000 товаров. Каждый скрипт будет запускаться по 3 раза. Мы будем подсчитывать среднее время выполнения скриптов с каждым файлом. Скрипт останавливается, если время его выполнения превышает 15 минут, и это время будет считаться временем выполнения скрипта.

После выполнения скрипта будем перезагружать базу данных, запускать сервис и специально написанным скриптом выполнять 1000 запросов к сервису на получение товаров, а затем проверять правильность полученных данных.

Далее составим таблицу, в которой будет указано имя участника, количество правильно полученных товаров и время выполнения скрипта. Отсортировав таблицу сначала по количеству правильно полученных товаров, а затем по времени выполнения запросов, мы получим победителя.