# Техническое задание на разработку системы обновления каталога по недостаточному количеству данных

# Введение

## Предметная область

Сайт sto01.ru - это PHP-проект на CMS NetCat. В рамках данного проекта будет обсуждаться только генерация каталога запасных частей.

Каталог запасных частей - это, простая с точки зрения интерфейса, витрина, с несколькими метриками для классификации (производитель, марка и модель машины, категория запчасти).

В идеальном мире этот каталог должен обновляться из 1С, однако у Заказчика нет такой возможности и у него присутствует только прайс-лист с запчастями, по каждой из которых есть следующие, важные для импорта, поля:

- Наименование
- Номер по каталогу
- Производитель

## Задача

Написать сервис, генерирующий на регулярной основе каталог с требуемой структурой, основываясь на данных из прайс-листа и открытых источников.

## Источник данных

Так как в прайс-листе нет практически никакой информации, её нужно получать из дополнительных источников. Drom.ru имеет свою доску объявлений, в которой продаются детали. У него есть и структура, и производители, и артикулы. Проблема в том, что данные эти заполняют пользователи и они не являются 100% истиной. Однако, детали продаются часто и объявлений много, поэтому статистическим методом можно получить наиболее вероятные значения для всех интересующих метрик.

## Краткий алгоритм работы

- **Администратор загружает прайс-лист в систему**
  Заказчику удобно отправлять прайс-лист на специальную почту (система периодически опрашивает почту и забирает прайс - это уже реализовано)
- **Сервис проверяют каждую деталь и осуществляет генерацию заданий:**

    - Если деталь есть в каталоге, но отсутствует в прайс-листе - исключить её из каталога (выключить)
    - Если деталь есть в каталоге и есть в прайс-листе - обновить цены
    - Если детали нет в каталоге, но она есть в прайс-листе - запустить парсинг данных для неё
- **Парсинг данных**
    - Сервис запускает поисковый запрос (или несколько) в drom.ru и получает в результате список объявлений с краткой информации о каждом
    - Берет заголовки и делает нечеткое сравнение названия объявления с названием в прайс-листе. Если процент совпадения нас устраивает - запускаем парсинг отдельной страницы объявления
    - На каждой странице объявления сервис извлекает из HTML все необходимые данные:
        - Категорию товара (и её положение в структуре каталога) по хлебным крошкам
        - Все возможные артикулы товара
        - Производителей
        - Подходящие марки и модели
    - Собрав достаточное количество данных (например, 10) сервис проводит объединение и находит наиболее часто встречающиеся варианты.
        - Например, то, что деталь относится к категории “Шины” указано 5 раз, а “ходовая часть” - один. Следственно берем категорию “Шины”. И так по каждой метрике
    - После получения метрик приложение добавляет все эти данные в свою структуру (категории и бренды могут быть новыми)
- **Импорт данных в каталог**
  Так как NetCat имеет свою сложную структуру каталога - не имеет смысла внедрять данные в неё “в сыром” виде. Очевидным решением является генерация файла по формату импорта из 1С и запуск механизма импорта. Таким образом мы не будем засорять NetCat служебной информацией и импорт будет ожидаемо верным с точки зрения каталога CMS.

## Текущее состояние приложения

На данный момент реализованы следующие функции:

- Парсинг данных с Drom.ru (возможно появились изменения в HTML, нужно проверить)
- Обновление всех данных в рамках отдельной базы данных (добавление новых категорий, соблюдение их структуры, марки, бренды и т. д.). Но это по историческим причинам - SQLite
- Использование прокси-серверов для обхода защиты Drom.ru

# Этапы работ

- Использовать для работы с данными популярный фреймворк, ORM и панель управления (для отладки и контроля за заданиями)
- Мигрировать систему на MySQL
- Ловить событие на загрузку прайс-листа и запускать необходимые задания
- Использовать прокси для работы с Drom.ru (их добавление, обновление и т. д.)
- Генерировать файл импорта 1С
- Запускать импорт каталога
- Отобразить все данные в нужном формате

# Панель управления

Панель управления позволит производить отладку парсинга и сгенерированных данных в удобном режиме. Для этой цели можно (и даже нужно) использовать любой доступный фремворк для генерации CRUD интерфейса.

## Разделы панели управления

- **Прокси**
  LIST, CREATE, UPDATE, DELETE, IMPORT (from CSV)
- **Прайс-листы**
  LIST, SHOW, DELETE, IMPORT (manual)
- **Бренды**
  LIST, CREATE, UPDATE, DELETE
- **Модели**
  LIST, CREATE, UPDATE, DELETE
- **Производители**
  LIST, CREATE, UPDATE, DELETE
- **Категории**
  TREE_LIST, CREATE, UPDATE, DELETE
- **Детали**
  LIST, CREATE, UPDATE, DELETE