

Тестовое задание (Middle PHP Dev) revision 1.1

Разработать парсер АТС классификатора и интерфейс с навигацией:

- 1 Парсер АТС-классификатора должен сгребить все сущности со страницы <http://compendium.com.ua/atc/> (код группы, название группы, ID parent группы, <http://joxi.ru/v29WLKqS3b4Mer> название препарата, производитель, URL товара на сайте compendium.com.ua) в локальную БД.
- 2 Веб-интерфейс
 - 2.1 Повторить на VUE поведение оригинального интерфейса (раскрытие дерева, разные цвета для сущностей). Идеально копировать не нужно, просто показать функцию и цвета. <http://joxi.ru/EA4kQ5ycwRXQz2>
 - 2.2 После заголовка “АТС-классификация” разместить input поле, при вводе текста со второго символа должен происходить поиск по LIKE с подбросом по препаратам (метод API - search). При нажатии на элемент в подбросе под INPUT полем должен появиться найденный препарат в таком же виде как в дереве (productName, brandName) (метод API - fetch).

Технологические ограничения:

- 1) Использование ООП.
- 2) PHP7 + любые стандартные расширения
- 3) MySQL или MariaDB. Доступ из PHP **исключительно** через PDO
- 4) Для интерфейса — VUE JS. Не смешивать PHP и HTML разметку в одном файле. Или php шаблонизатор или интерфейс грузит все через api.
- 5) Для парсера — CURL
- 6) Формат обмена данными API — json

Допускается использование микрофреймворков, но только с целью увеличения скорости или удобства разработки. Не нужно делать MVC ради MVC или применять фреймворк просто так.

Необходимо корректно перехватывать все ошибки (нет доступа к БД, не загружена схема, ...) и сообщать о них.

Эстетика html представления не важна, но должно быть все читаемо и там где можно клацнуть использовать css cursor: pointer

Система должна поставляться в виде архива в котором будет помимо системы такие файлы:

- README.md с кратким описанием системы
- scheme.sql с дампом БД без рабочих данных
- config.php в который необходимо будет внести параметры доступа mysql
- parser.php — будет запускаться из консоли \$php parser.php
- папка var/ - для временных файлов, лог-файла

Парсер должен писать все ошибки в лог-файл (с датой сообщения), а при передаче ему через командную строку ключа -v также дублировать их на экран (без даты).

API, описанное в формате Swagger YAML OpenAPI 3.x, будет плюсом.

Порядок проверки работоспособности:

- архив распакован в любую папку на веб-сервере (ubuntu, php7)
- установлены параметры доступа к БД в config.php
- база загружена командой mysql -u username -p dbname < scheme.sql
- запущен parser.php
- проект открыт в браузере, выполнено несколько кликов по дереву, выполнен поиск.