

# Тестовое задание на позицию Backend-разработчика (Middle)

Требуется разработать API сервис для отображения статей на PHP.

## Требования по сервису:

- Разработать следующие API методы:
  - Метод получения всех статей, строго с пагинацией (параметр page);
  - Метод получения всех категорий статей;
  - Метод получения статей по заданной категории (указывается id категории);
  - Метод получения статьи по slug;
- Разработать административную часть сервиса:
  - Форма авторизации, также реализовать редирект на эту форму, если неавторизованный пользователь открывает какой-то URL-адрес, относящийся к админпанели;
  - Добавление/редактирование пользователей для админпанели  
*Требуемые поля:*
    - Логин (email);
    - Пароль;
    - Статус активности пользователя;
  - Добавление/редактирования категорий статей  
*Требуемые поля:*
    - Наименование;
    - Статус активности категории;
    - Порядок категорий;
  - Добавление/редактирование статей  
*Требуемые поля:*
    - Наименование;
    - Слаг;
    - Категория (выпадающий список);
    - Изображение статьи (загрузка файла в формате PNG, JPEG или WEBP);
    - Содержимое статьи (использовать любой WYSIWYG-редактор с возможностью загрузки файлов внутри HTML-кода);
    - Статус активности статьи;
    - Порядок сортировки;
  - Во всех представлениях таблиц (список записей) значения должны быть отсортированы по умолчанию по порядку сортировки;
  - Во всех представлениях таблиц (список записей) значения добавить возможность изменения сортировки записи по любым полям, включая

ID, date создания и date последнего изменения;

- Во всех представлениях таблиц (список записей) использовать следующие плагины и реализовать их работу:
  - <https://abpetkov.github.io/switchery/> или аналогичный - активация/деактивация записи
  - <https://sortablejs.github.io/Sortable/> или аналогичный - для изменения порядка сортировки записей

Макет админпанели не принципиален. Можно взять Bootstrap или любой другой фронтенд-фреймворк для быстрого написания интерфейса.

Структура базы данных составляется на усмотрение кандидата. Структура должна быть описана в миграциях, для наполнения базы **обязательно** должны быть использованы сиды и библиотека Faker для добавления тестовых записей.

**Для выполнения этого задания требуется показать следующие навыки:**

- Чистый и понятный код;
- Уверенное владение Laravel (версия выбирается по желанию кандидата);
- Умение писать алгоритмы на языке программирования PHP, версии не меньше 7.4;
- Умение написания собственного REST API;
- Владение JavaScript;

**Для выполнения этого задания запрещено:**

- Подключать сторонние CDN, все JS-библиотеки должны быть подключены внутри проекта.
- Использовать готовые библиотеки для админпанели (например, SleepingOwl)
- Использовать JS-фреймворки
- Использовать Vite, при разворачивании проекта не должно быть ошибок, требующие его наличия

**Для приемки тестового задания необходимо подготовить:**

- Должна быть ссылка на репозиторий в любом месте (github/gitlab/bitbucket), с возможностью его скачать;
- В файле README.md должна быть написана инструкция по разворачиванию проекта.