

# Описание продукта

Давайте реализуем упрощенную версию новостного каталога.  
Предполагаемая аудитория 1 000 000 пользователей в месяц.  
Размер базы новостей составляет порядка 100 000 записей.

Наш каталог будет содержать 3 вида объектов:

- Новость
- Рубрика
- Авторы

## Новость

Представляет собой объект новости и должен содержать следующую информацию:

- Заголовок
- Анонс
- Текст
- Может быть написана только 1 автором (Например Иванова Кристина)
- Может относиться к нескольким рубрикам (Например, “общество”, “день города” )

## Автор

Содержит в себе как минимум информацию о конкретном авторе, а именно:

- фио автора
- аватар автора
- подпись автора

## Рубрика

Рубрики позволяют классифицировать новостные материалы в каталоге. Имеют название и могут в древовидном виде вкладываться друг в друга. В простом случае реализации уровень вложенности будет 2-3 рубрики, в сложном - произвольный. Вот пример возможных рубрик и их иерархии:

- Общество
  - городская жизнь
  - выборы
- День города
  - салюты
  - детская площадка
    - 0-3 года
    - 3-7 года
- Спорт

.....

## Функции каталога

Взаимодействие с пользователем происходит посредством HTTP запросов к API серверу. Все ответы представляют собой JSON объекты.

Сервер реализует следующие методы:

- \* выдача всех новостей конкретного автора
- \* выдача списка всех новостей, которые относятся к указанной рубрике
- \* выдача списка авторов

- \* выдача информации о статьях по их идентификаторам
- искать новости по рубрике, включая дочерние
- искать новости по рубрике, не включая дочерние
- поиск новостей по названию
- рубрикатор каталога сделать с произвольным уровнем вложенности рубрик друг в друга

## Задание

Формат маршрутов для доступа к методам, а также формат ответа и запросов мы предоставляем Вам реализовать и выбрать самим.

Обязательными для выполнения являются пункты в задании отмеченные символом (\*), реализацию всего остального функционала вы оставляете на свое усмотрение.

1. Спроектировать базу данных + реализовать скрипт ее формирования
2. Выполнить конфигурацию веб-сервера (любого)
3. Реализовать API согласно ТЗ
4. Подготовить тестовые данные (дамп базы, скрипт для генерации тестового набора данных)
5. Написать краткую документацию по использованию сервиса:
  - a. список методов API, с кратким описанием за что они отвечают;
  - b. списка всех параметров для каждого метода;
  - c. краткое описание формата ответа каждого метода.
6. Опубликовать/развернуть приложение, предоставить нам ссылки по которым можно протестировать работу сервиса:
  - a. в песочнице, например <https://koding.com>;
  - b. выложить работающее приложение на любой хостинг, а код на публичный репозиторий.

Если у Вас есть желание продемонстрировать знание какой-то технологии или подхода, то можно реализовать произвольную дополнительную функциональность на Ваше усмотрение.

## Используемые технологии

При выполнении задания требуется использовать следующие технологии:

- операционная система Ubuntu (или любая другая \*nix система, но не Windows);
- вебсервер Nginx
- язык программирования PHP для бэкенда
- база данных MongoDB или Mysql
- желательно оформить код по стандартам PSR;
- запрещается использовать различные framework'и