

Реализовать запросы для работы с категориями

Формат данных:

- json

Тип модели категории:

```
{
  id!: string,
  slug!: string, // Уникальное название на англ. в системе
  name!: string, // Название категории. Англ., кириллица
  description?:string, // Описание категории. Англ., кириллица
  createdAt!: Date, // Не управляется клиентом. Создается автом.
  active!: boolean, // Вкл, выкл
}
```

? - необязательное

! - обязательное

Реализовать следующие запросы:

- Создать категорию
- Изменить категорию. Добавить поддержку частичного обновления модели. Например возможность изменить только active без необходимости передавать всю модель. И так для любого поля модели.
- Удалить категорию
- Получить категорию по id или slug.
- Получить массив категорий по фильтру. См. фильтрация. Без фильтров по умолчанию отдаются первые две категории отсортированные по дате создания (поле createdAt). Новые категории идут первыми

Фильтрация

Принимаемая модель:

```
{
  // Поиск категорий по полю name
  // По вхождение переданного текста без учета регистра
  // Умеет искать названия с ё через переданное e
  // Например категория "Мёд". Запрос "?name=мед" найдет категорию
  // Мёд
  name?: string,

  // Все условия от поля name, но поиск идет по полю description
  description?: string,
```

// Поиск по полю active
// Поддерживаемые значения в параметрах: 0, false, 1, true.
// Например active=1 или active=true отдаем только активные категории
// active=0 или active=false отдаем только неактивные категории
active?: string,

// Все условия от поля name и description
// Поиск осуществляется по полю name и description через “или”
// Переданные фильтры по name и description должны игнорироваться
// Например ?name=тапочки&description=текст&search=мед
// При таком запросе фильтры name и description игнорируются
search?:string,

// Кол-во записей на страницу. Допустимы только цифры от 1-9
// Например pageSize=1. В ответе увидим только одну запись,
// т.е одну категория.
// При условии что есть записи в бд с переданными фильтрами
// По умолчанию 2
pageSize?: number,

// Номер страницы. Допустимы только цифры
// 0 и 1 являются первой страницей.
// В ответ приходит кол-во записей с учетом pageSize
// Например у нас в бд есть 4 записи(категории)
// Запрос: “?page=1&pageSize=2”.
// Ответ: первые две записи.
// Запрос: “?page=2&pageSize=2”.
// Ответ: следующие две записи.
page?: number,

// Сортировка категорий
// Принимает любое значение в виде названия поля модели категории
// и необязательного символа направления сортировки
// в виде - (дефис, тире).
// Символ означает направление сортировки как DESC
// Если переданного значения без учета “-” нет в модели категории,
// то работает сортировка по умолчанию.
// Например “?sort=active” ASC сортировка по полю active
// “?sort=-active” DESC сортировка по полю active
// “?sort=-name” DESC сортировка по полю name
// “?sort=-mike” игнорируем, так как нет такого поля в модели категории
// По умолчанию sort=-createdDate
sort?: string

}

? - необязательное

Фильтры комбинируются с сортировкой и пагинацией (page, pageSize).

Учитывать в данных фильтра вероятность инъекций и данные с пробелами.

Если в поле модели фильтрации передали только пробелы, то фильтрация не происходит. В таком случае работает запрос по умолчанию.

Требования к реализации:

- Nodejs (любые пакеты, фреймворки, библиотеки)
- Бд любая (любые пакеты, фреймворки, библиотеки)

Предлагаем разместить код на github или gitlab.

Реализация с docker будет плюсом.

Код в TypeScript - большим плюсом.