

Тестовое задание. Приложение на Backbone.js

Исходные данные

На сервере хранится информация о видеороликах двух типов: фильмы и музыка. Эту информацию можно получать через API, указывая номер страницы с данными. Одна страница может содержать ролики обоих типов и не более 10 штук.

Пример запроса:

http://rutube.ru/sample_feed/?page=3

Сервер возвращает данные в формате JSON со следующей структурой:

```
{
  "page": "3",
  "has_next": "true",
  "content": [
    {
      "id": "1",
      "type": "movie",
      "category": "трейлер",
      "page_url": "sample_url",
      "name": "Железный Человек 3 / Iron Man 3"
    },
    {
      "id": "2",
      "type": "music",
      "genre": "rock",
      "page_url": "sample_url",
      "name": "Limp Bizkit - Behind Blue Eyes"
    },
    ...
  ]
}
```

Описание полей страницы:

page - [integer] номер страницы

has_next - [boolean] наличие следующей страницы

content - [array] массив с данными, которые необходимо отобразить на странице

Описание полей ролика:

id - [integer] уникальный id ролика

type - [string] тип контента (movie/music)

category - [string] категория (поле уникально для роликов типа "movie")

genre - [string] жанр (поле уникально для роликов типа "music")

page_url - [string] адрес страницы ролика ([http://rutube.ru/\[page_url\]](http://rutube.ru/[page_url]))

name - [string] название ролика

Задача

Необходимо сделать страницу, на которой ролики выводятся в виде списка ссылок. Рядом с каждой ссылкой выводится поле `category/genre`, в зависимости от типа ролика.

При загрузке страницы запрашивается и показывается первая порция данных. Также на странице имеется кнопка "Загрузить еще", по клику на которой необходимо получать следующую страницу с данными и выводить результат. Если данные закончились, кнопка "Загрузить еще" должна быть скрыта.

Реализовать на фреймворке Backbone.js. Обратить особое внимание на то, что ролики могут быть разных типов. Для рендеринга данных необходимо использовать шаблонизатор (Underscore.js или любой другой).

Требований к внешнему виду нет, но аккуратность приветствуется.

Дополнительно

Будет большим плюсом, если Вы выполните задание на своем сервере/хостинге, и его можно будет посмотреть в работе. API в таком случае можно эмулировать статическими файлами, и вместо запросов с параметрами (`?page=3`) использовать ссылки на файлы с данными, например:

`http://example.org/testjob/3`