

# SPA мини прототипа админки

Требуется реализовать SPA (Single Page Application) мини прототипа админки, взаимодействующее с сервером только по REST API.

В качестве бекенда (REST API) использовать сервис QuickBlox:

<http://quickblox.com/developers/Overview>

**Application ID:** 48231

**Authorization key:** EeqhcgLW-qZPTqN

**Authorization secret:** yXP6ejeahQXqnPY

Для построения UI использовать один из готовых шаблонов администраторской панели для Bootstrap. Например, <https://github.com/puikinsh/gentelella>.

## Требуемый функционал

1. Авторизация пользователя админки.

UI (например): <https://colorlib.com/polygon/gentelella/login.html>

API: [http://quickblox.com/developers/Authentication and Authorization](http://quickblox.com/developers/Authentication%20and%20Authorization)

В бекенде создан пользователь:

**E-mail:** test@digitalsparta.com

**Username:** test\_admin

**Password:** password

2. Вывод таблицы элементов типа **Event**.

UI (например): [https://colorlib.com/polygon/gentelella/tables\\_dynamic.html](https://colorlib.com/polygon/gentelella/tables_dynamic.html)

API: [http://quickblox.com/developers/Custom Objects](http://quickblox.com/developers/Custom%20Objects)

В бекенде создана коллекция элементов типа Event со следующими полями:

- **BeginDate** – Date;
- **EndDate** – Date;
- **Title** – String;
- **Description** – String.

3. Вывод таблицы элементов типа **Speaker**.

Поля:

- **FirstName** – String;
- **LastName** – String;
- **Bio** – String;
- **Birthday** – Date;
- **Photo** – File.

## Дополнительные требования

1. Использование одного из фреймворков для построения SPA. Предпочтительно - Angular 2.
2. Язык разработки логики: JavaScript или TypeScript.
3. Работа приложения "без перезагрузки страницы" в браузере.
4. Задание со звездочкой (по желанию кандидата): реализация редактирования элементов коллекций.