

# Тестовый квест «Интерфейс Github»

Необходимо создать одностраничное приложение для просмотра issues с выбраного репозитория на Github.

Ниже перечислен функционал приложения. Начните, пожалуйста, с первой; остальные — на ваше усмотрение. Количество реализованного функционала не важно, однако их необходимо сделать полностью. Каждый фунционал будет оцениваться по отдельности.

Спецификацию API можно найти здесь — <https://developer.github.com/v3>.

## Техническое задание

|  |  |
| --- | --- |
| **Функционал** | **Описание** |
| Поле ввода для указания имени пользователя и названия репозитория | Данные не нужно загружать, пока не будет нажата кнопка «Поиск». Каждое обращение должно отображаться с указанием номера, названия, даты открытия. |
| Пейджинг для обращений | Убедитесь, что загружаются только те данные, которые отображаются. Предусмотрите возможность выбора количества объектов отображаемых на странице. |
| Обработка и индикация ошибок в интерфейсе | Реализуйте отображение состояния загрузки обращений с сервера. |
| Автодополнение в поле ввода | Измените поле ввода таким образом, чтобы после указания имени пользователя появлялся список автодополнения из существующих репозиториев выбранного пользователя. Загружать обращения нужно сразу после выбора в списке автодополнения. |
| Роутинг | Сделайте страницу детального просмотра обращения, отображаемую информацию выберите сами. |
| Вывод автора | Для каждого обращения выведите автора не просто в виде имени, а с указанием его аватарки и со ссылкой на его Github-профиль. |
| Дизайн | Сделайте стилистически законченный вид приложения. |
| Тесты | Напишите тесты на базовую функциональность. |

## Требования

* React.js или AngularJS для построения интерфейса;
* Использование библиотек UI (jQuery и другие) запрещено;
* Использование препроцессоров и систем сборки приветствуется ([Jade](https://github.com/CSSSR/sputnik/blob/master/Jade.md), [Stylus](https://github.com/CSSSR/sputnik/blob/master/Stylus.md), [Gulp](https://github.com/CSSSR/csssr-project-template));
* Архитектура должна быть масштабируемой;
* Приложение должно работать в современных браузерах (IE11+) и на мобильных устройствах;
* На приложение можно потратить максимум 12 часов.