

Описание

Реализовать приложение, отображающее погоду в городах с дополнительной визуализацией. На первом этапе местоположение пользователя должно определиться автоматически (возможно использование навигатора в браузере или сервиса, который определяет геопозицию по IP адресу). Поле ввода города должно быть предзаполнено. Интеграция должна быть сразу с двумя погодными сервисами(для того, чтобы была возможность увидеть погоду как по дням, так и по часам) и у пользователя должна быть возможность быстро переключаться между ними. Дополнительная визуализация подразумевает собой смену фона и основного изображения погоды (серый фон в дождь и изображение дождя, яркий фон в ясную погоду и изображение солнечной погоды и так далее).

Также должна быть реализована возможность подгрузки событий из гугл календаря и отображения их на UI(как это представлено на дизайне). Для этого будет необходимо осуществить интеграцию с гугл календарем(например, посредством следующей библиотеки - [react-google-calendar-api](#). При желании, можно использовать любую другую). Данный функционал подразумевает реализацию логики входа/выхода из аккаунта.

Приложение должно отлично выглядеть как на больших экранах, так и на экранах мобильных телефонов и планшетов. Запросы на погодные сервисы должны кэшироваться для избежания блокировки на бесплатных и публичных провайдерах и снижения трафика.

Дополнительные указания

Дизайн не содержит таких элементов, как селектор погодного сервиса и города. Это следует продумать самостоятельно, **так как дизайн является приблизительным и подлежит самостоятельному анализу**. Изображения погоды отсутствуют (но есть иконки), что тоже следует продумать и реализовать самостоятельно. На изображении с дизайном также присутствует блок событий, который необходимо наполнить данными из гугл календаря.

Обязательным в реализации приложения будет использование React, TypeScript, Redux, Redux-Saga, в Redux-Store необходимо хранить все данные с api и выбранным городом, данные должны кэшироваться и быть доступными после перезагрузки страницы, ESLint (конфиг airbnb). Приложение также должно быть покрыто различного вида тестами(Unit(Jest, enzyme), e2e(Cypress)). Структура кода не должна быть избыточной. Код также должен быть хорошо оптимизирован. В приложение необходимо применить все лучшие практики написания чистого кода.

Также особое внимание обращается на правильное построение архитектуры приложения.

Список API для использования(если представленные API не удовлетворяют каким-либо условиям задания, можно использовать любые другие открытые API для погоды) :

- <https://openweathermap.org/current>
- <https://docs.stormglass.io/#/>
- <https://home.openweathermap.org>

Приблизительный дизайн

