

# Курс по JavaScript. 2021

19 июн 2022, 18:08:09

старт: 27 сен 2021, 14:53:11

начало: 1 сен 2020, 00:00:00

Мы очень хотим, чтобы код вы написали сами, а не пользовались внешними библиотеками.

## Основное задание

Свадьба Аркадия сыграна, торт съеден, гости проводили молодоженов в их новую семейную жизнь. Что же дальше? А дальше наша молодая пара планирует свое свадебное путешествие. Прагматичный Аркадий посчитал важным заранее продумать свадебное путешествие, чтобы оно было максимально комфортным.

Аркадий едет в свадебное путешествие не только со своей возлюбленной супругой, но и с личным фотографом, поэтому перед нашим программистом-прагматиком стоит не менее прагматичная задача: составить расписание маршрута так, чтобы можно было сделать фотографии с разными погодными условиями в разных городах.

Для того чтобы достичь желаемого, Аркадий решил написать приложение, которое не уступит по своей красоте, простоте и удобству даже Кибертраку Илона Маска. Аркадий слышал, что у Яндекс.Погоды есть удобное API для получения прогноза погоды. Им он и решил воспользоваться в своем приложении.

Аркадию, как известному эстету, хотелось сделать свою библиотеку простой и удобной:

```
const availableGeoIds = [5, 10, 192, 20, 37];
```

```
const sunnyPlan = planTrip(availableGeoIds)
  .sunny(2)
  .cloudy(1)
  .sunny(2)
  .max(3)
  .build();
```

```
const cloudyPlan = planTrip(availableGeoIds)
  .cloudy(2)
  .sunny(1)
  .build();
```

Можно считать, что все города, которые будут передаваться в метод `planTrip`, расположены между собой на расстоянии, которое Аркадий может преодолеть меньше, чем за половину суток, поэтому про расстояния в рамках данного задания можно забыть. Путешествие начинается свой отсчет с одного из городов из списка.

## Методы-условия

Методы `cloudy`, `sunny`, `max` добавляют условия для будущего путешествия. Они принимают параметр — число дней, в течении которого должно соблюдаться заданное методом условие.

- `sunny(N)` — Солнечная погода N дней. Для API Яндекс.Погоды — это значения `condition` равные `clear` и `partly-cloud`
- `cloudy(N)` — Полная облачность N дней. Для API Яндекс.Погоды — это значения `condition` равные `cloudy` и `overcast`
- `max(M)` — Максимальное количество дней, в котором молодожены хотят быть в одном городе

Для соблюдения условий Аркадий может переместиться в другой город или же остаться в предыдущем городе путешествия.

Аркадий с женой очень любят изучать города, по которым путешествуют очень внимательно, поэтому строят маршрут так, чтобы максимальное количество дней побыть в одном городе.

## API планировщика

Метод `build()` иницирует построение маршрута и возвращает `Promise` с ответом.

Если маршрут по указанным условиям сформировать возможно, то ответом будет массив вида `[ { geoid: 1, day: 1 } ]`, где `day` — день поездки (начиная с 1).

Если по указанным условиям маршрут построить невозможно, метод `build()` возвращает `Promise`, который переходит в состояние `rejected` с ошибкой и сообщением: Не могу построить маршрут! .

## Библиотека для HTTP запросов

В браузер уже встроена глобально доступная функция `fetch` (она же `window.fetch`), которая позволяет удобно делать HTTP запросы.

Мы работаем в Node.js окружении и в неё встроены низкоуровневые модули `http`, `https`, `http2`. Они гораздо менее удобные, чем `fetch`. Предлагаем вам в решении домашнего задания воспользоваться библиотекой `node-fetch`, которая почти полностью повторяет браузерный аналог. Мы уже установили её для вас.

## API Яндекс.Погоды

Для получения прогноза погоды на ближайшие семь дней можно обратиться [к данному методу API](#), заменяя лишь параметр `geoid` на необходимый. За документацией ответа можно обратиться на [эту страницу](#). Для обращения к API вам понадобится получить ключ для тарифа "Тестовый". Положите полученный ключ в файл `key.json`, чтобы не отправлять ваш секретный ключ вместе с решением. Прогнозом на сутки для города следует считать прогноз в графе `day_short`.

## Дополнительные условия

- Запрос к серверу по каждому из городов не должен происходить более одного раза;
- Путешествие должно начинаться со дня выполнения запроса на построение маршрута;
- Порядок погодных условий нужно иметь в виду;
- Посетив город один раз — молодожены уже не захотят туда возвращаться;
- Обратите внимание, что на пути у наших молодожёнов не должно быть осадков! То есть погода может быть только `clear`, `partly-cloudy`, `cloudy` и `overcast`.

## Доступные команды

В этот раз задание нужно решать, используя [Node.js](#). Заготовку проекта вы найдете в файле, приложенном к условию задачи внизу (Ссылка "Скачать условие задачи"). Пишите свой код в файле `src/weather.js`

Перед началом работы выполните команду `npm install` для установки зависимостей.

Запускаются так: `npm run command`

Команда	Действие
<code>start</code>	Запуск <code>src/index.js</code>
<code>test</code>	Запуск тестов

## Отправка решения

При отправке решения отправляйте только файл `weather.js` или его содержимое.

## Полезные ссылки

- [Цепочка вызовов](#)
- [Promise](#)
- [Async/Await](#)

[Скачать условие задачи](#)

```
1 'use strict';
2
3 const fetch = require('node-fetch');
4
5 const API_KEY = require('./key.json');
6 const {json} = require("mocha/lib/reporters");
7
8 /**
9  * @typedef {object} TripItem Город, который является частью маршрута.
10  * @property {number} geoid Идентификатор города
11  * @property {number} day Порядковое число дня маршрута
12  */
13
14 class TripBuilder {
15
16   constructor(geoids) {
17     this.geoids = geoids;
18     this.maxDays = 7;
19     this.queueConditions = [];
20   }
21
22   addCondition(daysCount, condition) {
23     for (let i = 0; i < daysCount; i++) {
24       this.queueConditions.push(condition);
25     }
26   }
27
28   /**
29    * Метод, добавляющий условие наличия в маршруте
30    * указанного количества солнечных дней
31    * Согласно API Яндекс.Погоды, к солнечным дням
32    * можно приравнять следующие значения `condition`:
33    * * `clear`;
34    * * `partly-cloudy`.
35    * @param {number} daysCount количество дней
36    * @returns {object} Объект планировщика маршрута
37    */
38   sunny(daysCount) {
```

