

Создание скрипта для отображения статистики о коронавирусе с помощью API

Стрельцова Марина Николаевна

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

Современный мир стремительно меняется и людям на сегодняшний день приходится адаптироваться под нынешние условия жизни. Одной из проблем на данный момент является непрекращающаяся пандемия коронавируса, которая охватила все континенты мира. Люди пытаются всевозможными способами противостоять эпидемии и одним из таких способов является круглосуточный мониторинг заболеваемости для своевременного принятия тех или иных мер. В данной статье рассмотрено создание скрипта для отображения статистики о коронавирусе с помощью API.

Ключевые слова: коронавирус, API, PHP, статистика.

Creating a script to display coronavirus statistics using the API

Streltsova Marina Nikolaevna

Sholom-Aleichem Priamursky State University

Student

Abstract

The modern world is changing rapidly and people today are faced with current living conditions. One of the problems at the moment is the ongoing influenza pandemic, which covers all continents of the world. The frequency of occurrence of epidemic cases is a consequence of monitoring the incidence. This article describes how to create a script to display coronavirus statistics using the API.

Keywords: coronavirus, API, PHP, statistics.

1. Введение

1.1 Актуальность исследования

Современный мир стремительно меняется и людям на сегодняшний день приходится адаптироваться под нынешние условия жизни. Одной из проблем на данный момент является непрекращающаяся пандемия коронавируса, которая охватила все континенты мира. Люди пытаются всевозможными способами противостоять эпидемии и одним из таких способов является круглосуточный мониторинг заболеваемости для своевременного принятия тех или иных мер. В данной статье рассмотрено создание скрипта для отображения статистики о коронавирусе с помощью API.

1.2 Обзор исследований

В своем исследовании Д. А. Сутер и А. А. Кабанов представили возможности виртуальных помощников в социальных мессенджерах, как они могут быть полезны в период распространения вируса [1]. Провели анализ и расчет последствий пандемии коронавируса для экономики, отобразили статистику потерь, просчитали возможные последствия для Российской экономики в своей статье М. Г. Масальский и другие [2]. Т. А. Макаровских и М. С. А. Аботалев исследовали применение модели ARIMA прогнозирования временных рядов для анализа открытых данных о распространении коронавирусной инфекции в ряде регионов Российской Федерации [3]. В статье Н. С. Крыжной описано внедрение стороннего сервиса в свое веб-приложение через API. Также подробно рассматривается API Яндекс.Метрики: как выглядит, в каком формате возвращаются данные, какую статистику можно вывести, как составляется запрос, какие разделы в себя включает [4]. В своем исследовании Г. С. Сейдаметов и Ш. М. Усеинов рассмотрели взаимодействие с API VK для загрузки медиа-контента на сервера VK (ВКонтакте - социальная сеть) [5].

1.3 Цель исследования

Целью исследования является создание скрипта для отображения статистики о коронавирусе с помощью API.

2. Методы исследования

Для проведения исследования были выбраны следующие языки и программные средства:

1. PHP — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов [6];
2. Open Server Panel — портативный локальный WAMP/WNMP сервер, имеющий многофункциональную управляющую программу и большой выбор подключаемых компонентов [7];
3. PhpStorm — это интегрированная среда разработки на PHP с интеллектуальным редактором, которая глубоко понимает код, поддерживает PHP 5.3-7.3 для современных и классических проектов, обеспечивает лучшее в индустрии автодополнение кода, рефакторинги, предотвращение ошибок налету и поддерживает смешивание языков [8];
4. Covid19api — Бесплатный API для данных о коронавирусе, позволяющий создавать информационные панели, мобильные приложения или интегрировать их в другие приложения [9].

3. Результаты исследования

Перейдем на сайт, предоставляющий бесплатный API для данных о коронавирусе, и откроем документацию (Рис. 1).

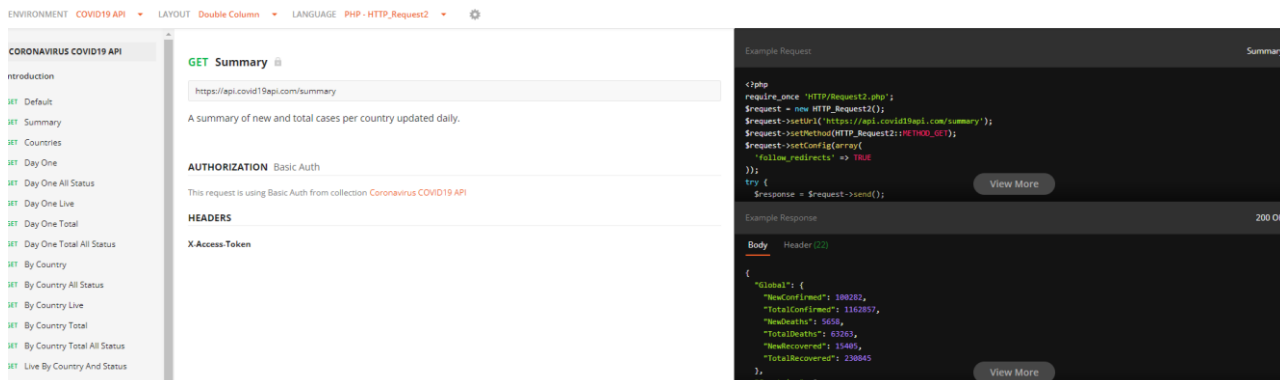


Рисунок 1 – Документация API

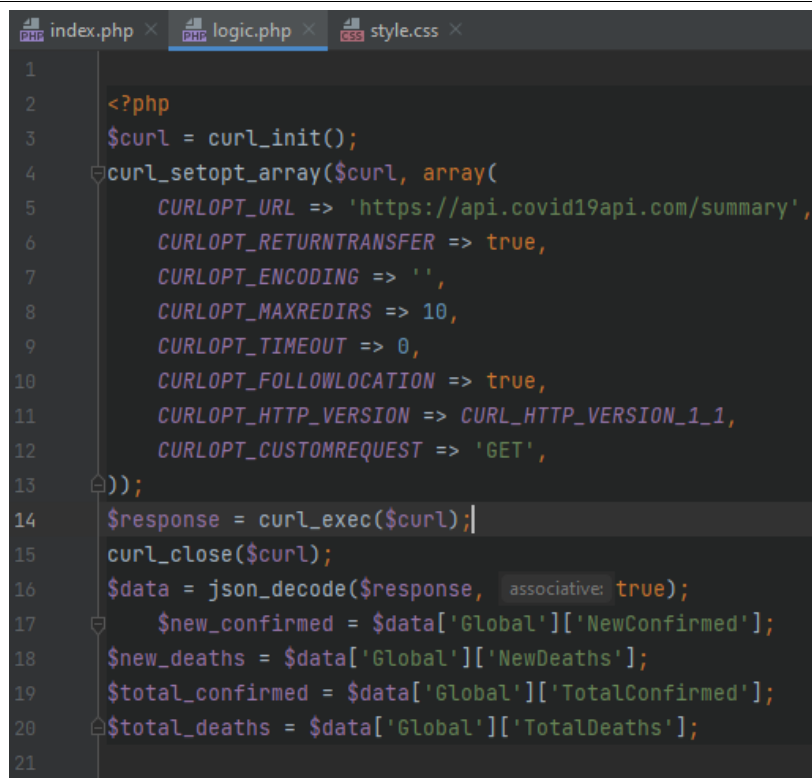
Для написания скрипта будем использовать метод `summary`, который позволяет получать данные о суммарном количестве больных и умерших в мире и по каждой стране в отдельности.

Далее создадим проект в RHPStorm и 3 файла: `index.php`, `logic.php` и `style.css`. Теперь переходим по адресу `https://api.covid19api.com/summary` для понимания структуры возвращаемых данных и посмотрим ответ сервиса в формате JSON (Рис. 2).

```
{
  "ID": "2f0b6adf-d7ea-46ee-aeed-c8b439ee0b5b",
  "Message": "",
  "Global": {
    "NewConfirmed": 2576433,
    "TotalConfirmed": 377756609,
    "NewDeaths": 569,
    "TotalDeaths": 6363,
    "NewRecovered": 15485,
    "TotalRecovered": 238845
  },
  "Countries": [
    {
      "ID": "4d35-ae17-b9d388a38b1f",
      "Country": "Afghanistan",
      "CountryCode": "AF",
      "Slug": "afghanistan",
      "NewConfirmed": 0,
      "TotalConfirmed": 437,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "43fb-43a3-bbf0-596dc6dfcd17",
      "Country": "Albania",
      "CountryCode": "AL",
      "Slug": "albania",
      "NewConfirmed": 0,
      "TotalConfirmed": 4118,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "4118-8f67-ab2fb6af17c4",
      "Country": "Algeria",
      "CountryCode": "DZ",
      "Slug": "algeria",
      "NewConfirmed": 0,
      "TotalConfirmed": 252,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "b1a1-d4be35480803",
      "Country": "Andorra",
      "CountryCode": "AD",
      "Slug": "andorra",
      "NewConfirmed": 0,
      "TotalConfirmed": 35958,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "c81611eab211",
      "Country": "Angola",
      "CountryCode": "AO",
      "Slug": "angola",
      "NewConfirmed": 0,
      "TotalConfirmed": 98116,
      "NewDeaths": 87,
      "TotalDeaths": 87,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "87fa4e46dda8",
      "Country": "Antigua and Barbuda",
      "CountryCode": "AG",
      "Slug": "antigua-and-barbuda",
      "NewConfirmed": 0,
      "TotalConfirmed": 0,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "b9386f4c-1e3b-4732-96b3-10d41b91ad95",
      "Country": "Argentina",
      "CountryCode": "AR",
      "Slug": "argentina",
      "NewConfirmed": 0,
      "TotalConfirmed": 0,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "91124df4-9cbd-4ce0-a436-64ba5aa96741",
      "Country": "Armenia",
      "CountryCode": "AM",
      "Slug": "armenia",
      "NewConfirmed": 0,
      "TotalConfirmed": 0,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "a6ae932a-1d2b-4687-b0b2-5a4bcd42c925",
      "Country": "Australia",
      "CountryCode": "AU",
      "Slug": "australia",
      "NewConfirmed": 0,
      "TotalConfirmed": 0,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "a613987f-850a-43e7-b40c-128f77938820",
      "Country": "Austria",
      "CountryCode": "AT",
      "Slug": "austria",
      "NewConfirmed": 0,
      "TotalConfirmed": 0,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "a9a10a5b-3b3d-4495-af56-f20d29bf42a4",
      "Country": "Azerbaijan",
      "CountryCode": "AZ",
      "Slug": "azerbaijan",
      "NewConfirmed": 0,
      "TotalConfirmed": 0,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "01T09:49:23.674Z",
      "Country": "Bahamas",
      "CountryCode": "BS",
      "Slug": "bahamas",
      "NewConfirmed": 0,
      "TotalConfirmed": 0,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "01T09:49:23.674Z",
      "Country": "Bahrain",
      "CountryCode": "BH",
      "Slug": "bahrain",
      "NewConfirmed": 0,
      "TotalConfirmed": 0,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    },
    {
      "ID": "58725a73-9f62-405b-81de-cbb3bc15555e",
      "Country": "Bangladesh",
      "CountryCode": "BD",
      "Slug": "bangladesh",
      "NewConfirmed": 0,
      "TotalConfirmed": 1798833,
      "NewDeaths": 0,
      "TotalDeaths": 0,
      "NewRecovered": 0,
      "TotalRecovered": 0
    }
  ]
}
```

Рисунок 2 – JSON структура возвращаемого ответа

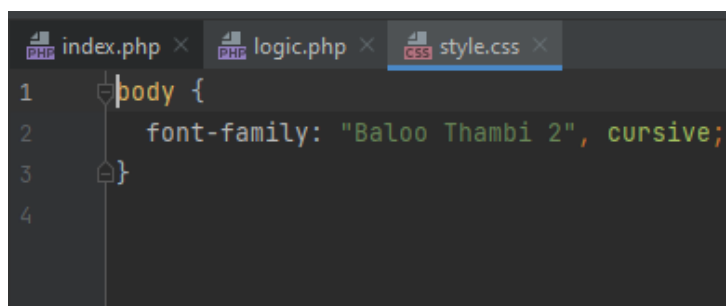
Скопируем код справа (Рис. 1) для подключения к сервису и вставим в файл `logic.php`, занося необходимые данные в переменные (Рис.3).



```
1
2  <?php
3  $curl = curl_init();
4  curl_setopt_array($curl, array(
5      CURLOPT_URL => 'https://api.covid19api.com/summary',
6      CURLOPT_RETURNTRANSFER => true,
7      CURLOPT_ENCODING => '',
8      CURLOPT_MAXREDIRS => 10,
9      CURLOPT_TIMEOUT => 0,
10     CURLOPT_FOLLOWLOCATION => true,
11     CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
12     CURLOPT_CUSTOMREQUEST => 'GET',
13 ));
14 $response = curl_exec($curl);
15 curl_close($curl);
16 $data = json_decode($response, associative: true);
17 $new_confirmed = $data['Global']['NewConfirmed'];
18 $new_deaths = $data['Global']['NewDeaths'];
19 $total_confirmed = $data['Global']['TotalConfirmed'];
20 $total_deaths = $data['Global']['TotalDeaths'];
21
```

Рисунок 3 – Код файла logic.php

В файле style.css пропишем параметры текста для отображения в браузере (Рис.4).



```
1  body {
2      font-family: "Baloo Thambi 2", cursive;
3  }
4
```

Рисунок 4 – Параметры текста

Далее прописываем html структуру в файле index.php, подключая необходимые для работы скрипты (Рис.5).

```

1 <?php
2     include 'logic.php';
3 >?
4 <!DOCTYPE html>
5 <html lang="en">
6 <head>
7     <meta charset="UTF-8">
8     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10    <!-- Bootstrap CSS -->
11    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
12
13    <!-- Bootstrap JS -->
14    <script src="https://code.jquery.com/jquery-3.4.1.js"></script>
15    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></script>
16    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
17
18    <!-- Font Awesome -->
19    <script src="https://kit.fontawesome.com/996973c893.js" crossorigin="anonymous"></script>
20
21    <!-- Google Fonts -->
22    <link href="https://fonts.googleapis.com/css2?family=Baloo+Thambi+2:wght@400;500;600;700;800&display=swap" rel="stylesheet">
23
24    <!-- My Stylesheet -->
25    <link rel="stylesheet" href="style.css">
26
27    <!-- My jQuery -->
28    <script src="main.js"></script>
29
30    <!-- Chart.js -->
31    <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.3/Chart.min.js"></script>
32
33 </head>

```

Рисунок 5 – Файл index.php

Затем следует написать «шапку» страницы, выводя общее число подтвержденных случаев заболевания коронавирусом и общее число умерших за время эпидемии (Рис. 6).

```

34 <body>
35     <div class="container-fluid bg-light p-5 text-center my-3">
36         <h1 class="">Статистика по Covid-19 </h1>
37         <h5 class="text-muted">Создано на основе сервиса с открытым исходным кодом для отслеживания всех случаев COVID-19 по всему миру.</h5>
38     </div>
39
40     <div class="container my-5">
41         <div class="row text-center">
42             <div class="col-4 text-warning">
43                 <h5>Общее число подтвержденных случаев</h5>
44                 <?php echo $total_confirmed; ?>
45                 <small class="text-danger pl-3"><i class="fas fa-arrow-up"></i><?php echo $new_confirmed; ?></small>
46             </div>
47
48             <div class="col-4 text-danger">
49                 <h5>Общее число умерших</h5>
50                 <?php echo $total_deaths; ?>
51                 <small class="text-danger pl-3"><i class="fas fa-arrow-up"></i><?php echo $new_deaths; ?></small>
52             </div>
53         </div>
54     </div>
55
56     <title>Статистика по Covid-19 </title>

```

Рисунок 6 – Код «шапки» страницы

Последним шагом построим таблицу, занося в ячейки название страны, общее число подтвержденных случаев и смертей в стране. Рядом с общими цифрами будет выводить информацию о новых случаях за сутки (Рис. 7).

```

56 <title>Статистика по Covid-19 </title>
57
58 <canvas id="myChart"></canvas>
59
60 <div class="container-fluid">
61   <div class="table-responsive">
62     <table class="table">
63       <thead class="thead-dark">
64         <tr>
65           <th scope="col">Страны</th>
66           <th scope="col">Общее число подтвержденных случаев</th>
67           <th scope="col">Общее число умерших</th>
68         </tr>
69       </thead>
70       <tbody>
71         <?php
72         foreach($data['Countries'] as $key){
73
74           <?>
75
76           <tr>
77             <th scope="row"><?php echo $key['Country'];?></th>
78             <td>
79               <?php echo $key['TotalConfirmed'];?>
80               <?php if($key['NewConfirmed'] != 0){ ?>
81                 <small class="text-danger pl-3"><i class="fas fa-arrow-up"></i><?php echo $key['NewConfirmed'];?></small>
82               <?php } ?>
83             </td>
84             <td><?php echo $key['TotalDeaths'];?>
85               <?php if($key['NewDeaths'] != 0){ ?>
86                 <small class="text-danger pl-3"><i class="fas fa-arrow-up"></i><?php echo $key['NewDeaths'];?></small>
87               <?php } ?>
88             </td>
89           </tr>
90         <?php }?>
91       </tbody>
92     </table>
93   </div>
94 </div>
95 </body>

```

Рисунок 7 – Информация о коронавирусе по странам

Запустим Open Server и перейдем на страницу index.php в браузере для просмотра результата выполнения кода (Рис. 8-9).

Статистика по Covid-19

Создано на основе сервиса с открытым исходным кодом для отслеживания всех случаев COVID-19 по всему миру.

Общее число подтвержденных случаев

377769130 ↑2588954

Общее число умерших

5669746 ↑6397

Рисунок 8 – Результат выполнения скрипта

| | | |
|------------------------|---|--|
| Belgium | 3155239 ↑99314 | 29052 ↑95 |
| Belize | 50487 | 625 |
| Benin | 26450 | 163 |
| Bhutan | 4891 | 4 |
| Bolivia | 855705 | 20951 |
| Bosnia and Herzegovina | 347467 | 14447 |
| Botswana | 253036 | 2581 |
| Brazil | 25463530 ↑102883 | 627589 ↑439 |

Рисунок 9 – Статистика о коронавирусе по странам

4. Выводы

Таким образом в статье был написан скрипт для отображения статистики о коронавирусе с помощью сервиса Covid19api через технологию API. Статистика содержит информацию об общем числе подтвержденных случаев заболевания коронавирусом и общему числу смертей. Также приведены данные по каждой стране в отдельности.

Библиографический список

1. Сутер Д. А., Кабанов А. А. Система обработки информации о коронавирусной инфекции в виде виртуального помощника // Информационные технологии и автоматизация управления. 2020. С. 207-212.
2. Масальский М. Г. и др. Влияние эпидемии коронавируса на экономику // Форум молодых ученых. 2020. №. 10. С. 423-427.
3. Макаровских Т. А., Аботалеб М. С. А. Автоматический подбор параметров модели ARIMA для прогноза количества случаев заражения и смерти от Covid-19 // Вычислительная математика и информатика. 2021. Т. 10. №. 2. С. 20-37.
4. Крыжная Н. С. Интеграция сервиса статистики посещаемости через API // Актуальные проблемы современной науки в 21 веке. 2017. С. 29-32.
5. Сейдаметов Г. С., Усеинов Ш. М. Загрузка фотографий через API VK на языке программирования PHP // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2018. №. 2. С. 35-41.
6. PHP URL: <https://www.php.net/> (дата обращения: 31.01.2022).
7. OpenServer Panel URL: <https://ospanel.io/> (дата обращения: 31.01.2022).
8. PHPStorm URL: <https://www.jetbrains.com/ru-ru/phpstorm/> (дата обращения: 31.01.2022).
9. Covid19api URL: <https://covid19api.com/> (дата обращения: 31.01.2022).