REST API ??????????? ? ?????????

Архитектура REST API была создана, чтобы упростить и стандартизировать процедуры доступа к данным из различных служб.

Эта логическая архитектура часто использует структуру данных JSON для отправки и получения данных. JSON — это открытый стандартный формат файла и формат обмена данными, в котором для хранения и передачи объектов данных, состоящих из пар атрибут-значение и массивов, используется удобочитаемый текст.

Пример данных JSON:

```
{
"api": "v2.7",
    "apipath": "https://www.rackcorp.net/api/v2.7"
    }
```

Эти данные могут быть представлены в виде массива или объекта на многих языках, таких как PHP, PYTHON и C#.

Rackcorp перенесла многие из своих функций для работы с архитектурой REST API, чтобы модернизировать и упростить процесс получения данных по протоколу HTTP. Каждый месяц мы добавляем новые сервисы через REST API. Важно, чтобы вы подписались на нас, чтобы получать последние обновления и последнюю версию нашего API.

????????? ?????? ????????? REST API:

- **методы HTTP - GET, POST, PUT, DELETE**

  Веб-разработчики, вероятно, знакомы с GET и POST, а также с другими методами HTTP, также иногда называемыми HTTP-глаголами. Эти методы определяют тип запроса к REST API.

- **Методы по использованию:**
  - GET - запросить данные
  - POST - вставить новые данные
  - PUT - обновить данные
  - DELETE - удалить данные

- **Имена ресурсов:**

Resources are sometimes referred to as the nouns that the HTTP verbs act upon. Earlier web services were built around remote procedure calls, which saw APIs as extensions of the code that called them. By contrast, REST resources can be accessed with multiple HTTP methods.

Ресурсы иногда называют существительными, на которые воздействуют HTTP-глаголы. Ранее веб-службы были построены на основе удаленных (на дистанции) вызовов процедур, в которых API рассматривались как расширения вызывающего их кода. Напротив, к ресурсам REST можно получить доступ с помощью нескольких методов HTTP.

- `GET /api/animals:` получить список животных.
- `POST /api/animals:` добавить новое животное
- `GET /api/animals/dog:` получить одно животное по идентификатору ID.
- `PUT /api/animals/dog:` обновить одно животное по идентификатору ID.
- `DELETE /api/animals/dog:` удалить животное по идентификатору ID.

- **Форматы данных:**

  Большинство запросов API будут возвращать с сервера содержимое, которое клиент должен интерпретировать. Редко это содержимое представляет собой обычный текст — обычно оно использует формат структурированных данных. Хотя REST не определяет какие-либо форматы данных, чаще всего используются JSON и XML.

  JSON:

```
{
  "id": "dog",
  "name": "Pet dog",
  "genus": "Canis",
  "img": "https://cdn2.thedogapi.com/images/1MZ0YbOpS.jpg"
}
```

  XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <id>dog</id>
  <name>Domestic dog</name>
  <genus>Canis</genus>
  <img>https://cdn2.thedogapi.com/images/1MZ0YbOpS.jpg
</img>
</root>
```

- **Статусы HTTP:**

  Поскольку API-интерфейсы REST зависят от стандартов HTTP, статус каждого запроса используется для сообщения результата запроса, такого как успех или неудача. Каждый код состояния предоставляет машиночитаемый ответ, а также сообщение, понятное человеку. Веб-разработчики (и ряд пользователей) знакомы со многими из них.

  - **200**: Удачный (Success)
  - **201**: Создан (Created)
  - **401**: Неавторизирован (Unauthorized)
  - **403**: Запрещен (Forbidden)
  - **404**: Не найден (Not found)
  - **429**: Слишком много запросов (Too many requests)

Хотя REST не является стандартом, существует множество других стандартов, часто связанных с REST. Например, OAuth охватывает стороннюю авторизацию ресурсов, а JSON PATCH описывает стандартный подход к методу HTTP PATCH для формата данных JSON. Важным стандартом, о котором следует помнить при разработке собственных API, является спецификация OpenAPI.

??? ??????? ?????? REST API ??? ?????????, ??????? ? ?????????? ???????

Вы можете создать REST API, используя любой язык программирования, такой как Javascript, PHP, PYTHON и C#. В этой статье мы добавим несколько примеров кода, чтобы объяснить, как сделать запрос с помощью Javascript (JQUERY) и PHP. Мы выбрали эти два языка, потому что это они самые популярные для развертывания веб-сайтов.

> **Мы настоятельно рекомендуем выполнять вызовы REST API только для бэкэнда. Будьте осторожны при использовании кода javascript для вызова URL-адреса REST API. Никогда не добавляйте свои учетные данные во фронтъенд.**

Javscript (JQUERY):

Во-первых, Вам нужно создать небольшую логику для вызова URL-адреса API в качестве примера ниже:

```
<script>
let query = {};
const v = 'v2.7';
const URL = 'https://www.rackcorp.net/api/'+v+'/dcs';
```

```
$.ajax({
ur: URL,
    type: 'GET',
    data: query,
    success: function(res) {
        console.log(res);
        alert(res);
    },
    error: function(res) {
    console.log(res);
        alert(res);
    }
});
</script>
```

Вы можете видеть, что код для выполнения запросов с использованием REST API довольно прост.

У нас есть переменная с именем «data», которая представляет собой объект, который будет содержать пару «ключ-значение», которая будет преобразована в запрос в бэкэнде. Затем мы создаем строковую переменную, которая является URL-адресом для подключения к REST API HTTP. С помощью этих двух переменных мы можем создать скрипт AJAX для вызова REST API. Если код правильный, логика перенаправит в SUCCESS, если код неправильный, он перенаправит к ERROR. Любой другой вариант Вы можете редактировать по своему усмотрению.

Будьте осторожны, чтобы использовать javascript для выполнения запросов REST API. Это связано с тем, что javascript является интерфейсным (фронт-энд) языком программирования, это означает, что код запускается на рабочем столе пользователя (ноутбук, телефон) и этот код может быть перехвачен хакерами и любым другим злоумышленником. Запросы REST API в javascript рекомендуются только в том случае, если вам не нужно передавать конфиденциальные данные, такие как ключи, пароль или личную информацию. В приведенном выше примере мы вызвали URL-адрес, чтобы перечислить все центры обработки данных (датацентры или DC) в Rackcorp, которые не требуют аутентификации, они открыты для всего мира.

PHP:

В PHP обычно разработчики используют функцию CURL для подключения через HTTP. В примере ниже вы можете понять, как реализовать запрос REST API с вашего сервера на другой сервер.

```
<?php
```

```
$v = 'v2.7';
$url = "https://api.rackcorp.net/api/". $v . "/dcs";
$query = [];

$curl = curl_init($url);
curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($query));
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);

$response = curl_exec($curl);

if($response) {
return json_decode($response, true);
}

return false;
```

Как видите, мы создали одни и те же переменные «v» и «query» в PHP для вызова URL-адреса REST API. На них мы начинаем создавать начальную команду CURL. Мы устанавливаем некоторые параметры в функции CURL, такие как POSTFIELDS и RETURNTRANSFER. У вас есть много других опций, которые вы можете добавить в CURL PHP. Просмотрите документацию по PHP, чтобы найти наилучший подход к логике вашего кода.

?????:

Запросы REST API — это довольно простой способ получать, вставлять и обновлять данные через Интернет с небольшими усилиями. В настоящее время в большинстве сервисов через Интернет реализованы функции REST API, и RACKCORP также имеет свой собственный стандарт. Если вам нужно подключиться к нам с помощью REST API, узнайте больше в RACKCORP REST API.

Revision #4
Created 3 August 2022 11:39:23 by KonS
Updated 16 August 2022 01:18:26 by KonS