Atitit 发送post http 功能总结

目录

[1. 常见三种模式 Form-data Form-urlencode Raw 1](#_Toc29996)

[2. Php code 1](#_Toc25865)

[2.1. urlencode 方式 默认 1](#_Toc21641)

[2.2. Json模式 设置 Content-Type: application/json 2](#_Toc27125)

[2.3. 纯文本 'Content-Type: text/plain 3](#_Toc23131)

[3. Nodejs 'axios' lib 3](#_Toc8665)

[4. Java 4](#_Toc16704)

[5. Js web 4](#_Toc405)

[6. Ref 4](#_Toc9321)

# 常见三种模式 Form-data Form-urlencode Raw

* application/x-www-form-urlencoded
* multipart/form-data

urlencode 方式 默认

如果发送方没有添加 Content-Type:，服务方deflt user urlencode 来解析。。。

Json 和纯文本都要设置 'Content-Type: text/plain; charset=utf-8',

Content-Type: application/json

其实不指定 Content-Type:

## 应用程序/x-www-form-urlencoded

的application/x-www-form-urlencoded内容类型描述了在单个块中发送在HTTP消息体形式的数据。与 GET 请求中 URL 的查询部分不同，数据的长度不受限制。但是，媒体服务器拒绝超过配置参数指定大小的请求MaxFileUploadSize。

此内容类型对于发送包含非 ASCII 字符的大量二进制数据或文本效率低下，并且不允许您上传文件。出于这些目的，Micro Focus建议将数据发送为multipart/form-data（请参阅[Multipart/form-data](https://www.microfocus.com/documentation/idol/IDOL_12_0/MediaServer/Guides/html/English/Content/Shared_Admin/_ADM_POST_requests.htm" \l "Multipar)）。

在请求中：

* 使用等号 ( =)将每个参数与其值分开。
* 用逗号 ( ,)分隔多个值。
* 用与号 ( &)分隔每个参数-值对。
* Base-64 编码任何二进制数据。
* URL 对所有非字母数字字符进行编码，包括 base-64 编码数据中的字符。

## 多部分/表单数据

在multipart/form-data内容类型中，HTTP 消息体分为多个部分，每个部分包含一个离散的数据部分。

每个消息部分都需要一个标头，其中包含有关该部分数据的信息。每个部分可以包含不同的内容类型；例如，text/plain，image/png，image/gif，或multipart/mixed。如果参数指定多个文件，则必须**multipart/mixed**在部分标题中指定内容类型。

每个消息部分的编码是可选的。消息部分标头必须指定除默认 (7BIT) 之外的任何编码。

Multipart/form-data 非常适合发送非 ASCII 或二进制数据，并且是唯一允许您上传文件的内容类型。有关表单数据的更多信息，请参阅<http://www.w3.org/TR/html401/interact/forms.html>。

# Php code

## urlencode 方式 默认

如果发送方没有添加 Content-Type:，服务方deflt user urlencode 来解析。。。

## Json模式 设置 Content-Type: application/json

Rest一般默认使用raw模式

|  |  |
| --- | --- |
| CURLOPT\_RETURNTRANSFER | 将 curl\_exec()获取的信息以文件流的形式返回，而不是直接输出 |

使用PHP curl获取页面内容或提交数据，有时候希望返回的内容作为变量储存，而不是直接输出。这个时候就必需设置curl的CURLOPT\_RETURNTRANSFER选项为1或true。

$response = curl\_exec($ch); // 已经获取到内容，没有输出到页面上。

curl\_exec($ch); 直接输出到页面

*/\*\*  
 \* PHP发送Json对象数据  
 \*  
 \** ***@param*** *$url 请求url  
 \** ***@param*** *$jsonStr 发送的json字符串  
 \** ***@return*** *array  
 \*/*function http\_post\_json($url, $jsonStr)  
{  
 $ch = curl\_init();  
 curl\_setopt($ch, *CURLOPT\_POST*, 1);  
 curl\_setopt($ch, *CURLOPT\_URL*, $url);  
 curl\_setopt($ch, *CURLOPT\_POSTFIELDS*, $jsonStr);  
 curl\_setopt($ch, *CURLOPT\_RETURNTRANSFER*, 1);//for setinto var not echo out  
 curl\_setopt($ch, *CURLOPT\_HTTPHEADER*, array(  
 'Content-Type: application/json; charset=utf-8',  
 'Content-Length: ' . strlen($jsonStr)  
 )  
 );  
 $response\_returnContent = curl\_exec($ch); //context ret  
 $curl\_getinfo = curl\_getinfo($ch);  
 echo 'curl\_getinfo'.*PHP\_EOL*;  
 echo json\_encode($curl\_getinfo);  
 $httpCode = curl\_getinfo($ch, *CURLINFO\_HTTP\_CODE*); //  
  
 curl\_close($ch);  
  
 return array($httpCode, $response\_returnContent);  
}

## 纯文本 'Content-Type: text/plain

//send is txt ,,let server use txt parser..  
curl\_setopt($ch, *CURLOPT\_HTTPHEADER*, array(  
 'Content-Type: text/plain; charset=utf-8',  
 'Content-Length: ' . strlen($str)  
 )  
);

# Nodejs 'axios' lib

const axios = require('axios')

axios

.post('https://whatever.com/todos', {

todo: 'Buy the milk'

})

.then(res => {

console.log(`statusCode: ${res.statusCode}`)

console.log(res)

})

.catch(error => {

console.error(error)

})

# Java

# Js web

# Ref

CURL中curl\_setopt的中文含义 - 天行侠 - 博客园