

# php模拟各种请求、包括 post、get、浏览器、客户端、文件上传等操作

 unirest-php-master.zip

Unirest是一套轻量级HTTP库中可用 [多种语言](#) (源码包见附件)。

## 特性

- 实用程序方法调用 [得到](#) , [头](#) , [帖子](#) , [把](#) , [删除](#) , [连接](#) , [选项](#) , [跟踪](#) , [补丁](#) 请求
- 支持参数形式,文件上传和自定义实体
- 支持[gzip](#)
- 支持基础研究、消化,谈判,NTLM身份验证本身
- 可定制的超时
- 可定制每个请求的默认标题(干)
- 自动JSON解析到一个本地对象的JSON响应

## 需求

- [旋度](#)
- PHP 5.4 +

## 安装

使用 [作曲家](#)

安装与作曲家unirest-php,添加以下 [composer.json](#) 文件:

```
{
  "require-dev": {
    "mashape/unirest-php": "2.*"
  }
}
```

或通过运行下面的命令:

```
composer require mashape/unirest-php
```

这将让你最新版本的记者和安装它。 如果你想要主人,未加标签的,版本可以使用下面的命令:

```
composer require mashape/php-test-reporter:@dev-master
```

作曲家安装自动装卸机在 [/供应商/ autoloader.php](#) 。 库包含在您的脚本,添加:

```
require_once 'vendor/autoload.php';
```

如果你使用Symfony2,自动装卸机必须是自动检测。

你可以看到这个库 [Packagist](#) 。

## 从源代码安装

从Github下载PHP库,然后包括 [Unirest.php](#) 在你的脚本:

```
git clone git@github.com:Mashape/unirest-php.git
```

```
require_once '/path/to/unirest-php/src/Unirest.php';
```

## 使用

## 创建一个请求

所以您可能想知道如何在PHP中使用Unirest使得创建请求更容易,让我们看一个工作示例:

```
$headers = array("Accept" => "application/json");
$body = array("foo" => "hellow", "bar" => "world");

$response = Unirest\Request::post("http://mockbin.com/request", $headers, $body);

$response->code;      // HTTP Status code
$response->headers;   // Headers
$response->body;       // Parsed body
$response->raw_body;  // Unparsed body
```

## 文件上传

上传文件在一个多部分形式表示使用的返回值 `Unirest\文件::添加(路径)` 作为一个参数的值:

```
$headers = array("Accept" => "application/json");
$body = array("file" => Unirest\File::add("/tmp/file.txt"));

$response = Unirest\Request::post("http://mockbin.com/request", $headers, $body);
```

## 自定义实体的身体

发送一个自定义的身体如一个JSON对象,而不是一个字符串或形式风格参数我们使用`json_encode`身体:

```
$headers = array("Accept" => "application/json");
$body = json_encode(array("foo" => "hellow", "bar" => "world"));

$response = Unirest\Request::post("http://mockbin.com/request", $headers, $body);
```

## 身份验证

首先,如果您正在使用 [Mashape](#):

```
// Mashape auth
Unirest\Request::setMashapeKey('<mashape_key>');
```

否则,通过用户名、密码 (可选) 默认为基本身份验证:

```
// basic auth
Unirest\Request::auth('username', 'password');
```

第三个参数是一个位掩码,将Unirest HTTP身份验证方法(s)你想要使用你的代理身份验证。

如果已经设置了不止一位,Unirest (在PHP的libcurl级别) 将首先查询网站,看看它支持身份验证方法,然后挑选你允许它使用最好的。 对于一些方法,这将产生一个额外的网络往返。

### 支持方法

方法	描述
<code>CURLAUTH_BASIC</code>	HTTP基本身份验证。 这是默认的选择
<code>CURLAUTH_DIGEST</code>	HTTP摘要认证。 中定义的 <a href="#">RFC 2617</a>
<code>CURLAUTH_DIGEST_IE</code>	HTTP摘要认证即风味。 <i>IE味道只是libcurl将使用一个特殊的“怪癖”,即已知使用之前版本7,有些服务器要求客户端使用。</i>
<code>CURLAUTH_NEGOTIATE</code>	HTTP谈判(SPNEGO)身份验证。 中定义的 <a href="#">RFC 4559</a>
<code>CURLAUTH_NTLM</code>	HTTP NTLM身份验证。 一个专有协议由微软发明和使用。
<code>CURLAUTH_NTLM_WB</code>	NTLM委托给winbind帮手。 身份验证是由一个单独的二进制程序。 看到 <a href="#">libcurl文档</a> 更多信息
<code>CURLAUTH_ANY</code>	这是一个方便的宏,集所有比特,从而使libcurl挑选任何它找到合适的。 libcurl将自动选择它找到一个最安全的。

方法	描述
<b>CURLAUTH_ANYSAFE</b>	这是一个方便的宏,集所有位除了基本从而使libcurl挑选任何它找到合适的。libcurl将自动选择它找到一个最安全的。
<b>CURLAUTH_ONLY</b>	这是一个元符号。或这个值在一个特定的身份验证值强制libcurl探针un-restricted身份验证,如果没有,只有这一个身份验证算法是可以接受的。

```
// custom auth method
Unirest\Request::proxyAuth('username', 'password', CURLAUTH_DIGEST);
```

之前版本的 **Unirest** 支持 基本身份验证 通过提供 用户名 和 密码 参数:

```
$response = Unirest\Request::get("http://mockbin.com/request", null, null, "username", "password");
```

这已经被废弃,将被完全移除 **v.3.0.0** 请使用 **Unirest::requireAuthentication()** 方法相反

## 饼干

设置cookie字符串指定cookie标题的内容。多个饼干用分号分隔空间(如紧随其后。“水果=苹果;颜色=红”)

```
Unirest\Request::cookie($cookie)
```

为使饼干饼干文件路径读取和存储饼干跨多个请求序列。

```
Unirest\Request::cookieFile($cookieFile)
```

**cookieFile**美元 必须正确的路径写权限。

## 请求对象

```
Unirest\Request::get($url, $headers = array(), $parameters = null)
Unirest\Request::post($url, $headers = array(), $body = null)
Unirest\Request::put($url, $headers = array(), $body = null)
Unirest\Request::patch($url, $headers = array(), $body = null)
Unirest\Request::delete($url, $headers = array(), $body = null)
```

- **url** —— 端点uri地址,或采取行动,要求信息。
- **头** —— 请求头作为关联数组或对象
- **身体** —— 请求主体关联数组或对象

你可以发送一个请求 [标准](#) 或自定义HTTP方法:

```
Unirest\Request::send(Unirest::Method::LINK, $url, $headers = array(), $body);
Unirest\Request::send('CHECKOUT', $url, $headers = array(), $body);
```

## 响应对象

在收到一个响应Unirest返回结果的形式一个对象,这个对象应该为每种语言有相同的密钥对响应的细节。

- **代码** HTTP响应状态码(例子 **200年** )
- **头** ----- HTTP响应头
- **身体** —— 解析响应主体适用,例如解析JSON响应对象/关联数组。
- **raw\_body** —— Un-parsed响应主体

## 高级配置

你可以设置一些高级配置优化Unirest-PHP:

### 自定义JSON解码旗帜

Unirest使用PHP的 [JSON扩展](#) 自动解码JSON响应。有时你可能想要返回关联数组,限制递归的深度,或者使用任何的 [自定义标记](#)。

为此,使用简单地设置所需的选项 **jsonOpts** 请求方法:

```
Unirest\Request::jsonOpts(true, 512, JSON_NUMERIC_CHECK & JSON_FORCE_OBJECT & JSON_UNESCAPED_SLASHES);
```

## 超时

你可以设置一个定制的超时值(秒):

```
Unirest\Request::timeout(5); // 5s timeout
```

## 代理

设置代理用于即将到来的请求。

你也可以设置代理类型之一 `CURLPROXY_HTTP`, `CURLPROXY_HTTP_1_0`, `CURLPROXY_SOCKS4`, `CURLPROXY_SOCKS5`, `CURLPROXY_SOCKS4A`, `CURLPROXY_SOCKS5_HOSTNAME`。

检查 [旋度文档](#) 更多信息。

```
// quick setup with default port: 1080
Unirest\Request::proxy('10.10.10.1');

// custom port and proxy type
Unirest\Request::proxy('10.10.10.1', 8080, CURLPROXY_HTTP);

// enable tunneling
Unirest\Request::proxy('10.10.10.1', 8080, CURLPROXY_HTTP, true);
```

## 代理 Authentication

通过用户名、密码 (可选) 默认为基本身份验证:

```
// basic auth
Unirest\Request::proxyAuth('username', 'password');
```

第三个参数是一个位掩码,将Unirest HTTP身份验证方法(s)你想要使用你的代理身份验证。

如果已经设置了不少于一位,Unirest (在PHP的libcurl级别) 将首先查询网站,看看它支持身份验证方法,然后挑选你允许它使用最好的。 对于一些方法,这将产生一个额外的网络往返。

看到 [身份验证](#) 更多细节的方法支持。

```
// basic auth
Unirest\Request::proxyAuth('username', 'password', CURLAUTH_DIGEST);
```

## 默认请求头

可以设置默认的标题,它将在每次请求:

```
Unirest\Request::defaultHeader("Header1", "Value1");
Unirest\Request::defaultHeader("Header2", "Value2");
```

你可以设置默认散装头通过数组:

```
Unirest\Request::defaultHeaders(array(
    "Header1" => "Value1",
    "Header2" => "Value2"
));
```

你可以随时清除默认的标题:

```
Unirest\Request::clearDefaultHeaders();
```

## 默认的旋度选项

你可以设置默认 [旋度的选择](#) 将在每个请求:

```
Unirest\Request::curlOpt(CURLOPT_COOKIE, "foo=bar");
```

你可以设置选项大部分通过数组:

```
Unirest\Request::curlOpts(array(  
  CURLOPT_COOKIE => "foo=bar"  
));
```

你可以随时清楚默认选项:

```
Unirest\Request::clearCurlOpts();
```

## SSL验证

您可以显式地启用或禁用SSL证书验证时使用SSL端点保护:

```
Unirest\Request::verifyPeer(false); // Disables SSL cert validation
```

默认情况下是 **真正的**。

## 实用方法

```
// alias for `curl_getinfo`  
Unirest\Request::getInfo()  
  
// returns internal cURL handle  
Unirest\Request::getCurlHandle()
```

---

的♥ [Mashape](#) 团队