2-20数据采集

1. 课堂案例

课堂讲解

一、上节回顾

正则表达用法

二、学习目标

利用正则表达式采集新闻数据

三、教学过程描述

1、了解curl

PHP支持的由Daniel

Stenberg创建的libcurl库允许你与各种的服务器使用各种类型的协议进行连接和通讯。

libcurl目前支持http、https、ftp、gopher、telnet、dict、file和ldap协议。libcurl同时也支持HTTPS认证、HTTP POST、HTTP PUT、 FTP 上传(这个也能通过PHP的FTP扩展完成)、HTTP 基于表单的上传、代理、cookies和用户名+密码的认证。

PHP中使用cURL实现Get和Post请求的方法

1-1、curl_init

初始化一个cURL会话

说明

resource curl_init ([string $url = NULL])

初始化一个新的会话，返回一个cURL句柄，供curl_setopt(), curl_exec()和curl_close() 函数使用。

参数

url

如果提供了该参数，CURLOPT_URL

选项将会被设置成这个值。你也可以使用curl_setopt()函数手动地设置这个值。

返回值

如果成功，返回一个cURL句柄，出错返回 FALSE。

1-2、curl_setopt

设置一个cURL传输选项。

说明

bool curl_setopt (resource $ch , int $option , mixed $value)

为给定的cURL会话句柄设置一个选项。

参数

ch

由 curl_init() 返回的 cURL 句柄。

option

需要设置的CURLOPT_XXX选项。

value

将设置在option选项上的值。

对于下面的这些option的可选参数，value应该被设置一个bool类型的值：

选项	可选value值	备注
CURLOPT_AUTOREFERER	当根据*Location:*重定向时，自动设置header中的*Referer:*信息。	
CURLOPT_BINARYTRANSFER	在启用CURLOPT_RETURNTRANSFER的时候，返回原生的（Raw）输出。	
CURLOPT_COOKIESESSION	启用时curl会仅仅传递一个session cookie，忽略其他的cookie，默认状况下cURL会将所有的cookie返回给服务端。session cookie是指那些用来判断服务器端的session是否有效而存在的cookie。	
CURLOPT_CRLF	启用时将Unix的换行符转换成回车换行符。	
CURLOPT_DNS_USE_GLOBAL_CACHE	启用时会启用一个全局的DNS缓存，此项为线程安全的，并且默认启用。	
CURLOPT_FAILONERROR	显示HTTP状态码，默认行为是忽略编号小于等于400的HTTP信息。	
CURLOPT_FILETIME	启用时会尝试修改远程文档中的信息。结果信息会通过curl_getinfo()函数的*CURLINFO_FILETIME*选项返回。curl_getinfo().	
CURLOPT_FOLLOWLOCATION	启用时会将服务器服务器返回的"*Location:* "放在header中递归的返回给服务器，使用CURLOPT_MAXREDIRS可以限定递归返回的数量。	
CURLOPT_FORBID_REUSE	在完成交互以后强迫断开连接，不能重用。	
CURLOPT_FRESH_CONNECT	强制获取一个新的连接，替代缓存中的连接。	
CURLOPT_FTP_USE_EPRT	启用时当FTP下载时，使用EPRT (或 LPRT)命令。设置为FALSE时禁用EPRT和LPRT，使用PORT命令 only.	
CURLOPT_FTP_USE_EPSV	启用时，在FTP传输过程中回复到PASV模式前首先尝试EPSV命令。设置为FALSE时禁用EPSV命令。	
CURLOPT_FTPAPPEND	启用时追加写入文件而不是覆盖它。	
CURLOPT_FTPASCII	CURLOPT_TRANSFERTEXT的别名。	
CURLOPT_FTPLISTONLY	启用时只列出FTP目录的名字。	
CURLOPT_HEADER	启用时会将头文件的信息作为数据流输出。	
CURLINFO_HEADER_OUT	启用时追踪句柄的请求字符串。	从 PHP 5.1.3 开始可用。CURLINFO_前缀是故意的(intentional)。
CURLOPT_HTTPGET	启用时会设置HTTP的method为GET，因为GET是默认是，所以只在被修改的情况下使用。	
CURLOPT_HTTPPROXYTUNNEL	启用时会通过HTTP代理来传输。	
CURLOPT_MUTE	启用时将cURL函数中所有修改过的参数恢复默认值。	
CURLOPT_NETRC	在连接建立以后，访问*~/.netrc*文件获取用户名和密码信息连接远程站点。	
CURLOPT_NOBODY	启用时将不对HTML中的BODY部分进行输出。	
CURLOPT_NOPROGRESS	启用时关闭curl传输的进度条，此项的默认设置为启用。 ***Note***: *PHP自动地设置这个选项为**TRUE**，这个选项仅仅应当在以调试为目的时被改变。*	
CURLOPT_NOSIGNAL	启用时忽略所有的curl传递给php进行的信号。在SAPI多线程传输时此项被默认启用。	cURL 7.10时被加入。
CURLOPT_POST	启用时会发送一个常规的POST请求，类型为：*application/x-www-form-urlencoded*，就像表单提交的一样。	
CURLOPT_PUT	启用时允许HTTP发送文件，必须同时设置CURLOPT_INFILE和CURLOPT_INFILESIZE。	
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。	
CURLOPT_SSL_VERIFYPEER	禁用后cURL将终止从服务端进行验证。使用CURLOPT_CAINFO选项设置证书使用CURLOPT_CAPATH选项设置证书目录如果CURLOPT_SSL_VERIFYPEER(默认值为2)被启用，CURLOPT_SSL_VERIFYHOST需要被设置成TRUE否则设置为FALSE。	自cURL 7.10开始默认为TRUE。从cURL 7.10开始默认绑定安装。
CURLOPT_TRANSFERTEXT	启用后对FTP传输使用ASCII模式。对于LDAP，它检索纯文本信息而非HTML。在Windows系统上，系统不会把*STDOUT*设置成binary模式。	
CURLOPT_UNRESTRICTED_AUTH	在使用CURLOPT_FOLLOWLOCATION产生的header中的多个locations中持续追加用户名和密码信息，即使域名已发生改变。	
CURLOPT_UPLOAD	启用后允许文件上传。	
CURLOPT_VERBOSE	启用时会汇报所有的信息，存放在*STDERR*或指定的CURLOPT_STDERR中。	

对于下面的这些option的可选参数，value应该被设置一个integer类型的值：

选项	可选value值	备注
CURLOPT_BUFFERSIZE	每次获取的数据中读入缓存的大小，但是不保证这个值每次都会被填满。	在cURL 7.10中被加入。
CURLOPT_CLOSEPOLICY	不是CURLCLOSEPOLICY_LEAST_RECENTLY_USED就是CURLCLOSEPOLICY_OLDEST，还存在另外三个CURLCLOSEPOLICY，但是cURL暂时还不支持。	
CURLOPT_CONNECTTIMEOUT	在发起连接前等待的时间，如果设置为0，则无限等待。	
CURLOPT_CONNECTTIMEOUT_MS	尝试连接等待的时间，以毫秒为单位。如果设置为0，则无限等待。	在cURL 7.16.2中被加入。从PHP 5.2.3开始可用。
CURLOPT_DNS_CACHE_TIMEOUT	设置在内存中保存DNS信息的时间，默认为120秒。	
CURLOPT_FTPSSLAUTH	FTP验证方式：*CURLFTPAUTH_SSL* (首先尝试SSL)，*CURLFTPAUTH_TLS* (首先尝试TLS)或*CURLFTPAUTH_DEFAULT* (让cURL自动决定)。	在cURL 7.12.2中被加入。
CURLOPT_HTTP_VERSION	*CURL_HTTP_VERSION_NONE* (默认值，让cURL自己判断使用哪个版本)，*CURL_HTTP_VERSION_1_0* (强制使用 HTTP/1.0)或*CURL_HTTP_VERSION_1_1* (强制使用 HTTP/1.1)。	
CURLOPT_INFILESIZE	设定上传文件的大小限制，字节(byte)为单位。	
CURLOPT_LOW_SPEED_LIMIT	当传输速度小于**CURLOPT_LOW_SPEED_LIMIT**时(bytes/sec)，PHP会根据**CURLOPT_LOW_SPEED_TIME**来判断是否因太慢而取消传输。	
CURLOPT_LOW_SPEED_TIME	当传输速度小于**CURLOPT_LOW_SPEED_LIMIT**时(bytes/sec)，PHP会根据**CURLOPT_LOW_SPEED_TIME**来判断是否因太慢而取消传输。	
CURLOPT_MAXCONNECTS	允许的最大连接数量，超过是会通过**CURLOPT_CLOSEPOLICY**决定应该停止哪些连接。	
CURLOPT_MAXREDIRS	指定最多的HTTP重定向的数量，这个选项是和**CURLOPT_FOLLOWLOCATION**一起使用的。	
CURLOPT_PORT	用来指定连接端口。（可选项）	
CURLOPT_PROTOCOLS	**CURLPROTO_***的位域指。如果被启用，位域值会限定libcurl在传输过程中有哪些可使用的协议。这将允许你在编译libcurl时支持众多协议，但是限制只是用它们中被允许使用的一个子集。默认libcurl将会使用全部它支持的协议。参见**CURLOPT_REDIR_PROTOCOLS** 可用的协议选项 为：CURLPROTO_HTTP、CURLPROTO_HTTPS、CURLPROTO_FTP、CURLPROTO_FTPS、CURLPROTO_SCP、CURLPROTO_SFTP、CURLPROTO_TELNET、CURLPROTO_LDAP、CURLPROTO_LDAPS、CURLPROTO_DICT、CURLPROTO_FILE、CURLPROTO_TFTP、CURLPROTO_ALL	在cURL 7.19.4中被加入。
CURLOPT_PROTOCOLS	**CURLPROTO_***的位域指。如果被启用，位域值会限定libcurl在传输过程中有哪些可使用的协议。这将允许你在编译libcurl时支持众多协议，但是限制只是用它们中被允许使用的一个子集。默认libcurl将会使用全部它支持的协议。参见**CURLOPT_REDIR_PROTOCOLS** 可用的协议选项 为：CURLPROTO_HTTP、CURLPROTO_HTTPS、CURLPROTO_FTP、CURLPROTO_FTPS、CURLPROTO_SCP、CURLPROTO_SFTP、CURLPROTO_TELNET、CURLPROTO_LDAP、CURLPROTO_LDAPS、CURLPROTO_DICT、CURLPROTO_FILE、CURLPROTO_TFTP、CURLPROTO_ALL	在cURL 7.19.4中被加入。
CURLOPT_PROXYAUTH	HTTP代理连接的验证方式。使用在**CURLOPT_HTTPAUTH**中的位域标志来设置相应选项。对于代理验证只有*CURLAUTH_BASIC*和*CURLAUTH_NTLM*当前被支持。	在cURL 7.10.7中被加入。
CURLOPT_PROXYPORT	代理服务器的端口。端口也可以在**CURLOPT_PROXY**中进行设置。	
CURLOPT_PROXYTYPE	不是*CURLPROXY_HTTP* (默认值) 就是*CURLPROXY_SOCKS5*。	在cURL 7.10中被加入。
CURLOPT_REDIR_PROTOCOLS	**CURLPROTO_***中的位域值。如果被启用，位域值将会限制传输线程在**CURLOPT_FOLLOWLOCATION**开启时跟随某个重定向时可使用的协议。这将使你对重定向时限制传输线程使用被允许的协议子集默认libcurl将会允许除FILE和SCP之外的全部协议。这个和7.19.4预发布版本种无条件地跟随所有支持的协议有一些不同。关于协议常量，请参照**CURLOPT_PROTOCOLS**。	在cURL 7.19.4中被加入。
CURLOPT_RESUME_FROM	在恢复传输时传递一个字节偏移量（用来断点续传）。	
CURLOPT_SSL_VERIFYHOST	1 检查服务器SSL证书中是否存在一个公用名(common name)。译者注：公用名(Common Name)一般来讲就是填写你将要申请SSL证书的域名(domain)或子域名(sub domain)。2 检查公用名是否存在，并且是否与提供的主机名匹配。	
CURLOPT_SSLVERSION	使用的SSL版本(2 或 3)。默认情况下PHP会自己检测这个值，尽管有些情况下需要手动地进行设置。	
CURLOPT_TIMECONDITION	如果在**CURLOPT_TIMEVALUE**指定的某个时间以后被编辑过，则使用*CURL_TIMECOND_IFMODSINCE*返回页面，如果没有被修改过，并且**CURLOPT_HEADER**为true，则返回一个"*304 Not Modified*"的header，**CURLOPT_HEADER**为false，则使用*CURL_TIMECOND_IFUNMODSINCE*，默认值为*CURL_TIMECOND_IFUNMODSINCE*。	
CURLOPT_TIMEOUT	设置cURL允许执行的最长秒数。	
CURLOPT_TIMEOUT_MS	设置cURL允许执行的最长毫秒数。	在cURL 7.16.2中被加入。从PHP 5.2.3起可使用。
CURLOPT_TIMEVALUE	设置一个**CURLOPT_TIMECONDITION**使用的时间戳，在默认状态下使用的是*CURL_TIMECOND_IFMODSINCE*。	

对于下面的这些option的可选参数，value应该被设置一个string类型的值：

选项	可选value值	备注
CURLOPT_CAINFO	一个保存着1个或多个用来让服务端验证的证书的文件名。这个参数仅仅在和CURLOPT_SSL_VERIFYPEER一起使用时才有意义。	
CURLOPT_CAPATH	一个保存着多个CA证书的目录。这个选项是和CURLOPT_SSL_VERIFYPEER一起使用的。	
CURLOPT_COOKIE	设定HTTP请求中"*Cookie:* "部分的内容。多个cookie用分号分隔，分号后带一个空格(例如，*"fruit=apple; colour=red"*)。	
CURLOPT_COOKIEFILE	包含cookie数据的文件名，cookie文件的格式可以是Netscape格式，或者只是纯HTTP头部信息存入文件。	
CURLOPT_COOKIEJAR	连接结束后保存cookie信息的文件。	
CURLOPT_CUSTOMREQUEST	使用一个自定义的请求信息来代替"*GET*"或"*HEAD*"作为HTTP请求。这对于执行"*DELETE*"或者其他更隐蔽的HTTP请求。有效值如"*GET*"，"*POST*"，"*CONNECT*"等等。也就是说，不要在这里输入整个HTTP请求。例如输入"*GET /index.html HTTP/1.0\r\n\r\n*"是不正确的。 ***Note:*** *在确定服务器支持这个自定义请求的方法前不要使用。*	
CURLOPT_EGDSOCKET	类似CURLOPT_RANDOM_FILE，除了一个Entropy Gathering Daemon套接字。	
CURLOPT_ENCODING	HTTP请求头中"*Accept-Encoding:* "的值。支持的编码有"*identity*"，"*deflate*"和"*gzip*"。如果为空字符串""，请求头会发送所有支持的编码类型。	在cURL 7.10中被加入。
CURLOPT_FTPPORT	这个值将被用来获取供FTP"POST"指令所需要的IP地址。"POST"指令告诉远程服务器连接到我们指定的IP地址。这个字符串可以是纯文本的IP地址、主机名、一个网络接口名（UNIX下）或者只是一个'-'来使用默认的IP地址。	
CURLOPT_INTERFACE	网络发送接口名，可以是一个接口名、IP地址或者是一个主机名。	
CURLOPT_KRB4LEVEL	KRB4 (Kerberos 4) 安全级别。下面的任何值都是有效的(从低到高的顺序)："*clear*"、"*safe*"、"*confidential*"、"*private*"。如果字符串和这些都不匹配，将使用"*private*"。这个选项设置为NULL时将禁用KRB4 安全认证。目前KRB4 安全认证只能用于FTP传输。	
CURLOPT_POSTFIELDS	全部数据使用HTTP协议中的"POST"操作来发送。要发送文件，在文件名前面加上@前缀并使用完整路径。这个参数可以通过urlencoded后的字符串类似'*para1=val1&para2=val2&...*'或使用一个以字段名为键值，字段数据为值的数组。如果*value*是一个数组，*Content-Type*头将会被设置成*multipart/form-data*。	
CURLOPT_PROXY	HTTP代理通道。	
CURLOPT_PROXYUSERPWD	一个用来连接到代理的"*[username]:[password]*"格式的字符串。	
CURLOPT_RANDOM_FILE	一个被用来生成SSL随机数种子的文件名。	
CURLOPT_RANGE	以"*X-Y*"的形式，其中X和Y都是可选项获取数据的范围，以字节计。HTTP传输线程也支持几个这样的重复项中间用逗号分隔如"*X-Y,N-M*"。	
CURLOPT_REFERER	在HTTP请求头中"*Referer:* "的内容。	
CURLOPT_SSL_CIPHER_LIST	一个SSL的加密算法列表。例如*RC4-SHA*和*TLSv1*都是可用的加密列表。	
CURLOPT_SSLCERT	一个包含PEM格式证书的文件名。	
CURLOPT_SSLCERTPASSWD	使用CURLOPT_SSLCERT证书需要的密码。	
CURLOPT_SSLCERTTYPE	证书的类型。支持的格式有"*PEM*" (默认值)，"*DER*"和"*ENG*"。	在cURL 7.9.3中被加入。
CURLOPT_SSLENGINE	用来在CURLOPT_SSLKEY中指定的SSL私钥的加密引擎变量。	
CURLOPT_SSLENGINE_DEFAULT	用来做非对称加密操作的变量。	
CURLOPT_SSLKEY	包含SSL私钥的文件名。	
CURLOPT_SSLKEYPASSWD	在CURLOPT_SSLKEY中指定了的SSL私钥的密码。 ***Note:*** *由于这个选项包含了敏感的密码信息，记得保证这个PHP脚本的安全。*	
CURLOPT_SSLKEYTYPE	CURLOPT_SSLKEY中规定的私钥的加密类型，支持的密钥类型为"*PEM*"(默认值)、"*DER*"和"*ENG*"。	
CURLOPT_URL	需要获取的URL地址，也可以在curl_init()函数中设置。	
CURLOPT_USERAGENT	在HTTP请求中包含一个"*User-Agent:* "头的字符串。	
CURLOPT_USERPWD	传递一个连接中需要的用户名和密码，格式为："*[username]:[password]*"。	

对于下面的这些option的可选参数，value应该被设置一个数组：

选项	可选*value*值	备注
CURLOPT_HTTP200ALIASES	200响应码数组，数组中的响应吗被认为是正确的响应，否则被认为是错误的。	在cURL 7.10.3中被加入。
CURLOPT_HTTPHEADER	一个用来设置HTTP头字段的数组。使用如下的形式的数组进行设置：array('Content-type: text/plain', 'Content-length: 100')	
CURLOPT_POSTQUOTE	在FTP请求执行完成后，在服务器上执行的一组FTP命令。	
CURLOPT_QUOTE	一组先于FTP请求的在服务器上执行的FTP命令。	

对于下面的这些option的可选参数，value应该被设置一个流资源（例如使用fopen()）：

选项	可选*value*值
CURLOPT_FILE	设置输出文件的位置，值是一个资源类型，默认为*STDOUT* (浏览器)。
CURLOPT_INFILE	在上传文件的时候需要读取的文件地址，值是一个资源类型。
CURLOPT_STDERR	设置一个错误输出地址，值是一个资源类型，取代默认的*STDERR*。
CURLOPT_WRITEHEADER	设置header部分内容的写入的文件地址，值是一个资源类型。

对于下面的这些option的可选参数，value应该被设置为一个回调函数名：

选项	可选*value*值
CURLOPT_HEADERFUNCTION	设置一个回调函数，这个函数有两个参数，第一个是cURL的资源句柄，第二个是输出的header数据。header数据的输出必须依赖这个函数，返回已写入的数据大小。
CURLOPT_PASSWDFUNCTION	设置一个回调函数，有三个参数，第一个是cURL的资源句柄，第二个是一个密码提示符，第三个参数是密码长度允许的最大值。返回密码的值。
CURLOPT_PROGRESSFUNCTION	设置一个回调函数，有三个参数，第一个是cURL的资源句柄，第二个是一个文件描述符资源，第三个是长度。返回包含的数据。
CURLOPT_READFUNCTION	回调函数名。该函数应接受三个参数。第一个是 cURL resource；第二个是通过选项 CURLOPT_INFILE 传给 cURL 的 stream resource；第三个参数是最大可以读取的数据的数量。回调函数必须返回一个字符串，长度小于或等于请求的数据量（第三个参数）。一般从传入的 stream resource 读取。返回空字符串作为 *EOF*（文件结束）信号。
CURLOPT_WRITEFUNCTION	回调函数名。该函数应接受两个参数。第一个是 cURL resource；第二个是要写入的数据字符串。数据必须在函数中被保存。函数必须返回准确的传入的要写入数据的字节数，否则传输会被一个错误所中断。

返回值

成功时返回 TRUE，或者在失败时返回 FALSE。

1-3、curl_exec

执行一个cURL会话。

说明

mixed curl_exec (resource $ch)

执行给定的cURL会话。

这个函数应该在初始化一个cURL会话并且全部的选项都被设置后被调用。

参数

ch

由 curl_init() 返回的 cURL 句柄。

返回值

成功时返回 TRUE, 或者在失败时返回 FALSE。 然而，如果CURLOPT_RETURNTRANSFER选项被设置，函数执行成功时会返回执行的结果，失败时返回FALSE 。

1-4、curl_close

关闭一个cURL会话。

说明

void curl_close (resource $ch)

关闭一个cURL会话并且释放所有资源。cURL句柄ch 也会被释放。

参数

ch

由 curl_init() 返回的 cURL 句柄。

返回值

没有返回值。

2、设计采集规划表

3、添加采集规则

4、开始执行采集

4-1、set_time_limit

置脚本最大执行时间

如果超过了此设置，脚本返回一个致命的错误。默认值为30秒，或者是在php.ini的max_execution_time被定义的值。

4-2、 str_replace

以其他字符替换字符串中的一些字符(区分大小写)。

该函数必须遵循下列规则:

如果搜索的字符串是数组, 那么它将返回数组。

如果搜索的字符串是数组, 那么它将对数组中的每个元素进行查找和替换。

如果同时需要对数组进行查找和替换, 并且需要执行替换的元素少于查找到的元素的数量, 那么多余元素将用空字符串进行替换

如果查找的是数组, 而替换的是字符串, 那么替代字符串将对所有查找到的值起作用。

注释:该函数区分大小写。请使用 str_ireplace() 函数执行不区分大小写的搜索。

注释:该函数是二进制安全的。

语法

str_replace(find,replace,string,count)

参数 描述

find 必需。规定要查找的值。

replace 必需。规定替换 find 中的值的值。

string 必需。规定被搜索的字符串。

count可选。对替换数进行计数的变量。

四、小结

1、使用正则去匹配我们需要的数据

2、设置超时防止过长时间加载

2. 课堂练习

1、建好采集表

2、完成采集添加

3. 课后练习

完善新闻管理->新闻采集

4. 资料扩展

phpQuery