

Beginners Guide

So what's REST all about? Actually, it's just the way the Internet already works. Your web browser makes a request for a web page, and the web site responds.

However as you probably know, when you create a form in HTML, there are two method types:

- POST
- GET

You know this from HTML:

```
<form action="http://www.bbc.co.uk/dosomething"
method="GET">
<input type="text" name="page" value="1" />
</form>
```

With a GET request, the form variables are sent in the URL, e.g:

```
http://www.bbc.co.uk/dosomething?page=1
```

With a POST request, the form variables are 'hidden'. They are not included in the URL but are still sent in the request:

```
http://www.bbc.co.uk/dosomething
(page=1)
```

Actually there are more than 2 methods possible for HTTP requests. Of all the methods available, REST APIs often use 4 methods:

- POST
- GET
- PUT
- DELETE

All this means is that the request type (POST, GET, PUT or DELETE) is specified in the request, in the same way as POST or GET is specified when a web browser sends form data to a web server.

It's really the same process, which is partly why a REST API is said to use already-existing Internet technologies.

A REST API gives a meaning to each of these request types. The meanings are generally:

- POST - make some modification.
- GET - retrieve some information.
- PUT - create something.
- DELETE - delete something.

Therefore, they are sometimes referred to as 'verbs' (or 'actions') since they specify what to 'do':

- modify
- retrieve
- create
- delete

Do it to what?... The resource

We've seen that using a REST API we can specify an action: modify, retrieve, create or delete. So how do we specify what we want this action to be performed on?

That's where the 'resource' comes in. Actually this is just a unique URL which describes what to modify/retrieve/etc. In our case we are performing actions on a Text Marketer account... let's say we want to retrieve information about credits. This is the unique URL we use to identify our REST API credits resource:

```
http://api.textmarketer.co.uk/services/rest/credits
```

So the URL above specifies the credits resource in the Text Marketer REST API. In REST terminology it is referred to as a 'noun'. So you can see that we have 'verbs' (modify/retrieve/etc) and 'nouns' (credits, etc). This is just jargon to describe the action and the thing to do the action on.

So now we can combine the action/verb with the resource/thing/noun, for example 'retrieve the number of credits' is a GET request on the credits resource URL:

<http://api.textmarketer.co.uk/services/rest/credits> (GET)

This returns the number of credits on the account. The response will look something like this if you view it in **Firefox** (Chrome only displays the number):

```
<response
processed_date="2011-04-08T14:51:42+01:00">
<credits>461</credits>
</response>
```

What actions can be performed?

GET requests

We've seen the GET action on the credits resource returns the number of credits on the account. Different actions are available on different resources. Some resources may only accept a GET action, others may only accept a POST action, and yet others may accept both.

POST requests

The credits resource allows the POST (modify) action.

Calling a POST action on the credits resource (i.e. sending a HTTP request, of type POST, to the URL <http://api.textmarketer.co.uk/services/rest/credits>) allows you to transfer credits between accounts.

How does this work?

POST requests are for modifying a resource. But how do we specify exactly what needs to be modified? A POST request sent to the credits resource allows us to transfer credits between accounts, so we need some way of specifying how many credits to transfer and to which account.

Like an HTML form that uses a POST method, we will pass data within our POST request. In fact, if you were to build an HTML page with a form like this:

```
<form
action="http://api.textmarketer.co.uk/services/
rest/credits"
method="POST">
Quantity:<input name="quantity"
value="1"/><br/>
Target account:<input name="target"/><br/>
Target password:<input
name="target_password"/><br/>
<input type="submit" name="Test Credits POST"/>
</form>
```

...using input fields as above to specify the number of credits and the account to transfer to, you could successfully make a call to the API. However the API will respond with XML which isn't very friendly for a web browser! So you will instead do this programmatically, within your preferred programming language.

For example, this is what the same thing looks like in PHP:

```

<?php
$url =
'http://api.textmarketer.co.uk/services/rest/credits';
$data = array('quantity'=>'1', 'target'=>'961',
'target_password'=>'mypassword',
'username'=>'u', 'password'=>'u');
$data = http_build_query($data, '', '&');
// we're using the curl library to make the request
$curlHandle = curl_init();
curl_setopt($curlHandle, CURLOPT_URL, $url);
curl_setopt($curlHandle, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curlHandle, CURLOPT_POSTFIELDS, $data);
curl_setopt($curlHandle, CURLOPT_POST, 1);
$responseBody = curl_exec($curlHandle);
$responseInfo = curl_getinfo($curlHandle);
curl_close($curlHandle);
?>

```

If you're not familiar with PHP, it doesn't matter. All this illustrates is that we are making a call to the URL <http://api.textmarketer.co.uk/services/rest/credits> using a POST request ("CURLOPT_POST" in the example code), and as data we're passing:

- quantity
- target
- target_password
- username
- password

which is built into a string like this, just like you would see in a URL, before being sent to the server as POST fields:

```

quantity=1&target=961&target_password=mypassword&username=u&
password=p

```

The first three arguments are what we need to specify the number of credits to transfer, and which account to transfer them to. We'll cover the username/password arguments later.

PUT requests:

The group resource allows a PUT request, which lets you create a group. How to make a PUT request is not within the scope of this document.

DELETE requests:

Currently, the only resource allowing DELETE requests on Text Marketer REST API is the sms resource. This is used to delete a previously scheduled SMS via API.

- Next: [Quick Start Reference Sheet](#)
- Home: [Developers Documentation](#)