

# MCO API

## To create an API key

In MCO under “Manage Users” there is an API key generator. Use this to generate or change the API key you are using in your software. You can create one API key per account.



The screenshot shows a web interface for generating an API key. At the top, there is a 'save' button. Below it is a section titled 'Logo Information' containing a 'Current Logo:' label, a 'Choose File' button, the text 'No file chosen', an 'Upload' button, and a 'Clear Logo' button. The next section is titled 'API key' and contains a text input field with the value '6YGTP6DinV6Q0F53586A3048D4D932FD5D6366D3E81C7c0Va3kT1O90'. Below the input field is a 'Generate NEW API key' button and a note: '(This will break any currently connected software that uses this key.)'

## Using the API

The MCO API can be called using either GET or POST and returns an XML string result.

In its most basic usage, it can be called with a web browser.

The API's URL is [http://mymarketingcalendaroffice.com/MCO\\_DEV2/services/](http://mymarketingcalendaroffice.com/MCO_DEV2/services/)

Which returns:

“<error>NO KEY</error>”

## Valid Parameters

*apikey* = The key generated on the MCO web site

*action* = The type of request being made [GetCalendars], [GetCampaigns], [GetProjects], [GetTasks], [GetEvents], [GetPressreleases]

*CalendarID*, *CampaignID*, *ProjectID*, *TaskID*, *EventID*, *PressreleaseID* = filter results by this item id

*filter* = an additional filter that can be used with any of the returned XML's elements. More than one *filter* parameter can be specified.

## Example

```
http://mymarketingcalendaroffice.com/MCO_DEV2/services/?  
apikey=[YourAPIkey]&action=GetCampaigns&CalendarID=3&filter=campaignname:test:PART&filter  
=campaigncreatede:2012:GT
```

This example retrieves all campaigns with calendarID 3 having “test” in their name and created after 2012.

## **Filter Rules**

The filter has two required parts: the xml element name and the comparison value.

*filter=elementname:value*

It has an optional third option that is a comparison type.

These are: *EQ*, *PART*, *GT*, *LT*.

*filter=elementname:value:comparisontype*

*EQ* – (default) is exactly equal to

*PART* – the value is a part of

*GT* – is greater than the value

*LT* – is less than the value

The filters can be stacked to create date ranges, etc.

## A POST example in PHP

```
<?php
$baseurl = "http://mymarketingcalendaroffice.com/MCO_DEV2/services/";
$key = "TheAPIKey";

$fields = array(
    'apikey' => urlencode($key)
    , 'action' => urlencode("getTasks")
    , 'filter' => urlencode("taskdescription:budget:part")
);
$fields_string = "";
foreach($fields as $key=>$value) { $fields_string .= $key.'='.$value.'&'; }
rtrim($fields_string, '&');

$ch = curl_init();
$fh = fopen("out.xml", "w");
//set the url, number of POST vars, POST data
curl_setopt($ch,CURLOPT_URL, $baseurl);
curl_setopt($ch,CURLOPT_POST, count($fields));
curl_setopt($ch,CURLOPT_POSTFIELDS, $fields_string);
curl_setopt($ch,CURLOPT_CONNECTTIMEOUT,0);
curl_setopt($ch, CURLOPT_FILE, $fh);
//execute post
curl_exec($ch);

//close connection
curl_close($ch);
fclose($fh);
?>
```