How to Make an API Call Using REST

# Overview

In this tutorial, you will learn how to make an API call using REST in the command terminal. This tutorial is aimed at programmers who want their system to communicate with others over the internet. This tutorial assumes you already know a little bit about APIs and REST.

For more introductory information about APIs, click [here](https://www.postman.com/what-is-an-api/), and for more information about REST, click [here](https://restfulapi.net/).

# Requirements

You need knowledge of command line commands, APIs, and REST prior to best utilize this information. You will also need a computer with a command terminal, a known API URL, and internet access to follow along with the example.

# Making an API Call

You will be using variations of the following command in order to retrieve, modify, or remove information from your desired API endpoint:

**curl -x {REQUEST\_TYPE} {API\_URL} --header {HEADER} --data {INFORMATION}**

* **{REQUEST\_TYPE}**- REST call type, can be get, post, put, or delete
* **{API\_URL}**- Desired API endpoint to communicate with
* **{HEADER}**- General server/host details, if any, format in JSON
* **{INFORMATION}**- New data being provided to the API, if any, format in JSON

Note that the header and data are not always necessary depending on the type of request.

If the request is successful, the output will show a Request Code of 200. Some requests will also output other information.

## 

## 

## Tutorial

Be sure to have usable API endpoints available to substitute into the example commands. Follow these steps to make a REST API call:

1. Open a command terminal window.
2. If necessary, retrieve an access token by using the following command:

**curl -x post {$AUTHENTICATION\_URL}**

* **{$AUTHENTICATION\_URL}** is the URL of the API endpoint that provides an access token. They usually look something like [api.example.com/authenticate](http://api.example.com/authenticate).

1. Use other call types to communicate with the API. Your command could look something like this:

**curl -x put** [**api.example.com**](http://api.example.com)**/v1 --header “key:value” --data “{json:string}”**

## Example

The following example is an ice cream shop website. It walks through checking existing flavor information, adding a new flavor, modifying a flavor’s availability, and removing a discontinued flavor.

### Get the Access Token

Command Terminal Input:

| **curl -x post api.icecream.shop/authenticate** |
| --- |

Expected Output:

| **Request Code: 200**  **token: stringofcharacters1a2b3c4d** |
| --- |

### Check Existing Flavor Info

Command Terminal Input:

| **curl -x pull api.icecream.shop/flavors --header “token: stringofcharacters1a2b3c4d”** |
| --- |

Expected Output:

| **Response Code: 200**  **{flavors: vanilla, strawberry, chocolate}** |
| --- |

### Add a New Flavor

Command Terminal Input:

| **curl -x post api.icecream.shop/flavors --header “token: stringofcharacters1a2b3c4d” --data “{pistachio: {in\_stock: true}}”** |
| --- |

Expected Output:

| **Response Code: 200** |
| --- |

### 

### Change Existing Flavor Info

Command Terminal Input:

| **curl -x put api.icecream.shop/flavors/chocolate --header “header: stringofcharacters1a2b3c4d” --data “{in\_stock: false}}”** |
| --- |

Expected Output:

| **Response Code: 200** |
| --- |

### Change Existing Flavor Info

Command Terminal Input:

| **curl -x delete api.icecream.shop/flavors/chocolate --header “header: stringofcharacters1a2b3c4d”** |
| --- |

Expected Output:

| **Response Code: 200** |
| --- |