

Credits: http://www.bls.gov/bls/api\_feature\_article.htm

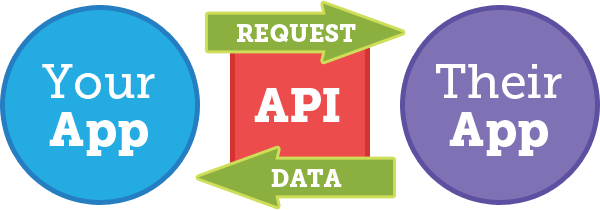
***API an Interface to Application's Programming.***

We are always overwhelmed by so many Candy Crush Request on Facebook. But how Candy Crush sends its invite to you through Facebook. How Candy Crush knows you are on Facebook. How can Facebook allow any app to access your data without your physical intervention? Here’s the answer:

Suppose I have a Candy Crush app and I want to put my latest scores on Facebook.

How do I do it? There are two ways:

1. I should hire a person who takes my latest score and posts it on facebook on my behalf. This is a ridiculous and a cumbersome way.
2. I should use a digital person. API is that digital person, who takes your score and uses Facebook API to posts your scores on Facebook on your behalf. So, now you know that Candy Crush takes your data (if authorized) using Facebook API and sends you an invite to your friends.

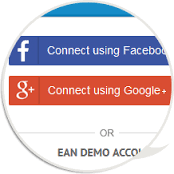


Credits: https://moz.com/blog/apis-for-datadriven-marketers

Another question arise is what will happen if there is no API and I want to show latest tweets of Apple Corporation on my web page:

1) I should parse twitter page of Apple Corporation. If twitter changes its look and DOM structure, my web page parser will stop working and I also need to change my parser. This will be over burden.

2) With an API, the exact structure of request and response is documented upfront by twitter.com, and is likely to remain constant, regardless of whether the website changes its look and feel for human visitors



When we use LOGIN VIA FACEBOOK OR GOOGLE, to log into different websites, we use their API'S

Now you all must have heard that API is stateless. This means there is no client context being stored on server side or in simple words there are no sessions.

Why API’s are stateless OR Why there are no Sessions ?

ST in REST means State Transfer, which means to transfer state from client to server in every request. This is the only way to scale to millions of concurrent users. It’s easier to distribute a stateless application across load-balanced servers. A stateless application is also easy to cache.

Credits: http://techcrunch.com/2014/05/06/google-maps-on-mobile-gets-uber-integration-and-more/

Examples of known API’s around us:

1. **UBER/OLA CAB SERVICE DISPLAYING INFO OF CARS ON GOOGLE MAPS**

Now google maps has an API, which is used by cab services to show cab on Google Maps. Without Google Maps API, both these companies have to make their own maps which is extremely difficult

2) YouTube APIs: Google's APIs lets developers integrate YouTube videos and functionality into websites or applications. YouTube APIs include the YouTube Analytics API, YouTube Data API, YouTube Live Streaming API, YouTube Player APIs and others

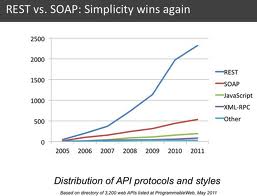
We are surrounded by API'S and I literally mean this

What happens when we want to print a document : Microsoft Word asks the active printer to return its status. Microsoft Word does not care what kind of printer is available. The API worries about that.

Even the simple copy and paste feature uses API. When we copy and paste anything in computer, the program uses API provided by O.S. to work

Two Most Famous Types of API

|  |  |
| --- | --- |
| REST | SOAP |
| REST stands for REpresentational State Transfer | SOAP stands for Simple Object Access Protocol |
| Made by microsoft, Soap is more rigid set of rules and regulations | Rest is flexible and not bound by so much rules. It is an architectural style rather than a protocol |
| Everything is mostly embed in URL, however there may be cases where addiditional information is required to pass.  URL is made of nouns and four verbs are used GET, POST, PUT, DELETE... On the basis of these verbs, program decides which function to perform  **GET:** It is used when we want to retrieve some data, for ex: Get the recipe of cake id 31.  https://www.cakerecipe.com/id/31  Optional Params can also be provided like headers  **POST:** It is used to insert some data in the database, for ex: POST https://www.cakerecipe.com/new/  POSTParams: Id: 1, Name: HotChocolate  **PUT:** It is used to update some data in the database  PUT https://www.cakerecipe.com/put/id/1  Params: Name: Chocolava  **DELETE:** It is used to delete some data in the database  DELETE https://www.cakerecipe.com/id/1 | You must make XML for even the simplest task.  With SOAP in JavaScript means writing a ton of code to perform extremely simple tasks because you must create the required XML structure absolutely every time. |
| REST requires use of HTTP | Language, platform, and transport independent |
| Efficient: REST can use smaller message formats | Inefficient: SOAP uses XML for all messages |



*Which API to use clearly depends on the use Case, but because of ease of use, REST is the most commonly used form of API all over the world*

Ccredits: http://www.slideshare.net/jmusser/j-musser-semtechjun2011

In Naukri.com we use API’s extensively, such as

* **Job details API** : It gives us details of a job given its job-id. An example is given below

|  |
| --- |
| *Mandatory Parameters*  JobId :  for example : *2019348237*  *Mandatory Headers*  clientId :  abc or xyz  *Optional Headers*  Accept: application/json OR application/xml  Authorization: NAUKRI id=<something>,authsource=<authsource type>(Eg. NAUKRI  zid=123123131231231231,authsource=mobileapp)  You can get full list of headers on [Headers-Wikipedia](https://en.wikipedia.org/wiki/List_of_HTTP_header_fields)  GET Request : http://www.naukri.com/jobapi/jobid/20137    **Sample Response**:  {  "Job": {  "JobId": "20137",  "FunctionalArea": "Banking ",  "Location": "Alexandria - Egypt",  "Description": "Sales Manager",  "Designation": "sales manager",  "LogoUrl": "",  "TELogoUrl": "http://logo.com/mylogo",  "Company": {  "Name": "ABC",  "Profile": "917"  },  }    } |

* **JobSearch/SRP API** : It returns job details given atleast a single keyword or location. Sample response is given below
* **Similar Jobs API** : Provides similar jobs, given a Job-Id. Sample response is given below

And many more such as **Job recommendations API, Job recommendations count API, Cluster API**.

*How to make a simple REST API In PHP*

Construct an Abstract Class

This class will act as a wrapper for all of the custom endpoints that our API will be using. It means

1. it must be able to take in our request
2. grab the endpoint from the URI string
3. detect the HTTP method (GET, POST, PUT, DELETE)
4. assemble any additional data provided in the header or in the URI

|  |
| --- |
| abstract class API {  /\*\*  \* Property: method  \* The HTTP method this request was made in, either GET, POST, PUT or DELETE  \*/  protected $method = '';  /\*\*  \* Property: endpoint  \* The Model requested in the URI. eg: /files  \*/  protected $endpoint = '';  /\*\*  \* Property: verb  \* An optional additional descriptor about the endpoint, used for things that can  \* not be handled by the basic methods. eg: /files/process  \*/  protected $verb = '';  /\*\*  \* Property: args  \* Any additional URI components after the endpoint and verb have been removed, in our  \* case, an integer ID for the resource. eg: /<endpoint>/<verb>/<arg0>/<arg1>  \* or /<endpoint>/<arg0>  \*/  protected $args = Array();  /\*\*  \* Property: file  \* Stores the input of the PUT request  \*/  protected $file = Null;     public function \_\_construct($request) {  $this->args = explode('/', rtrim($request, '/'));  $this->endpoint = array\_shift($this->args);  if (array\_key\_exists(0, $this->args) && !is\_numeric($this->args[0])) {  $this->verb = array\_shift($this->args);  }   $this->method = $\_SERVER['REQUEST\_METHOD'];  if ($this->method == 'POST' && array\_key\_exists('HTTP\_X\_HTTP\_METHOD', $\_SERVER)) {  if ($\_SERVER['HTTP\_X\_HTTP\_METHOD'] == 'DELETE') {  $this->method = 'DELETE';  } else if ($\_SERVER['HTTP\_X\_HTTP\_METHOD'] == 'PUT') {  $this->method = 'PUT';  } else {  throw new Exception("Unexpected Header");  }  }   switch($this->method) {  case 'DELETE':  case 'POST':  $this->request = $this->performSomething($\_POST);  break;  case 'GET':  $this->request = $this->performSomething($\_GET);  break;  case 'PUT':  $this->request = $this->performSomething($\_GET);  $this->file = file\_get\_contents("php://input");  break;  default:  $this->\_response('Invalid Method', 405);  break;  }  } } |

Once that's done, the abstract class will pass the request information on to a method in the concrete class to actually perform the work

### Creating a Concrete Class

|  |
| --- |
| class MyAPI extends API {  protected $User;   public function \_\_construct($request, $origin) {  parent::\_\_construct($request);   # Abstracted out for example...If you have keys enabled  $APIKey = new Models\APIKey();  $User = new Models\User();   if (!array\_key\_exists('apiKey', $this->request)) {  throw new Exception('No API Key provided');  } else if (!$APIKey->verifyKey($this->request['apiKey'], $origin)) {  throw new Exception('Invalid API Key');  } else if (array\_key\_exists('token', $this->request) &&  !$User->get('token', $this->request['token'])) {   throw new Exception('Invalid User Token');  }   $this->User = $User;  }   /\*\*  \* Example of an Endpoint  \*/  protected function example() {  if ($this->method == 'GET') {  return "Your name is " . $this->User->name;  } else {  return "Only accepts GET requests";  }  }  } |

### Creating a Concrete Class

|  |
| --- |
| try {  $API = new MyAPI($\_REQUEST['request'], $\_SERVER['HTTP\_ORIGIN']);  echo $API->processAPI(); } catch (Exception $e) {  echo json\_encode(Array('error' => $e->getMessage())); } |

If you want to know more about how we use APIs in our engineering projects, drop a comment below.