
What's an API?

API stand for Application Programmer Interface.

Application program interface (**API**) is a set of routines, protocols, and tools for building software applications. An **API** specifies how software components should interact and **APIs** are used when programming graphical user interface (GUI) components.

APIs allow developers to work with data in a simple way.

These days RESTful APIs are the coolest.

What's REST?

Representational State Transfer.

Not quite a standard. More of a philosophy.

RESTful APIs are the coolest.

From API Consumer

I provide my auth

I want a format that I <3

I need documentation

I need consistency

I need warning of changes

To API Provider

I issue auth creds to keep data secure

I provide JSON

I document my all endpoints

I respect the “API contract”

I use Semantic Versioning

I <3 my consumers

The “API Contract”

This is an unspoken commitment to consistency and usability of the resources you provide with respect to the developers that are building on your API.

This includes, but is not limited to, security, definition, http response codes, output changes, documentation, API uptime and valid formatting of returns.



REST API Philosophy

APIs are more about delivering a good Dev Experience than great coding.

→ **Design**

Deliver data needed that others easily consume

→ **Configure**

Make changes to deliver more or more easily

Definition and Documentation

Versioning allows me to know when your API Definition changes. Semantic versioning is the norm.

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes
2. MINOR version when you add functionality in a backwards-compatible manner
3. PATCH version when you make backwards-compatible bug fixes

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

Providing an API to the public requires Documentation. The more the better. Including a test console is the best.

There are plenty of API doc frameworks. Even auto gen. Your API code IS the documentation.

Swagger is my fav.



Live the API Contract

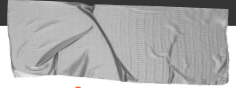
Your consumers need you to respect the API contract.



Tip

Those that provide the data are looking for others to build on it. So let's all play nice.

API barriers are anything that makes your API harder to consume



Remember

Every barrier I come across in discovering your API is another reason for me not to use your API.



Error Status Codes

HTTP Status Code	Reason
409	Limit greater than 100.
409	Limit invalid or below 1.
409	Invalid or unrecognized parameter.
409	Empty parameter.
409	Invalid or unrecognized ordering parameter.
409	Too many values sent to a multi-value list filter.
409	Invalid value passed to filter.

Try it out!

Marvel API Docs

One error status code???



Untappd to me

1/28/15

UNTAPPD
DRINK SOCIALLY

Hello Antonio,

As an Untappd API developer, we wanted to inform you that you will need to update your application to use our secure, HTTPS API endpoint (<https://api.untappd.com>). We will be discontinuing our standard **HTTP API endpoint on 2/1/15**. This will require you to update your code for your application to change the <http://> protocol to <https://>.

After 2/1/15, your API key will no longer work over **HTTP** and return a **500 HTTP error**. The security of the Untappd community is extremely important to us and this change will help keep things safer.

We recommend that you join our API Developer Group at <http://untpd.it/apigroup> for up the information about the API.

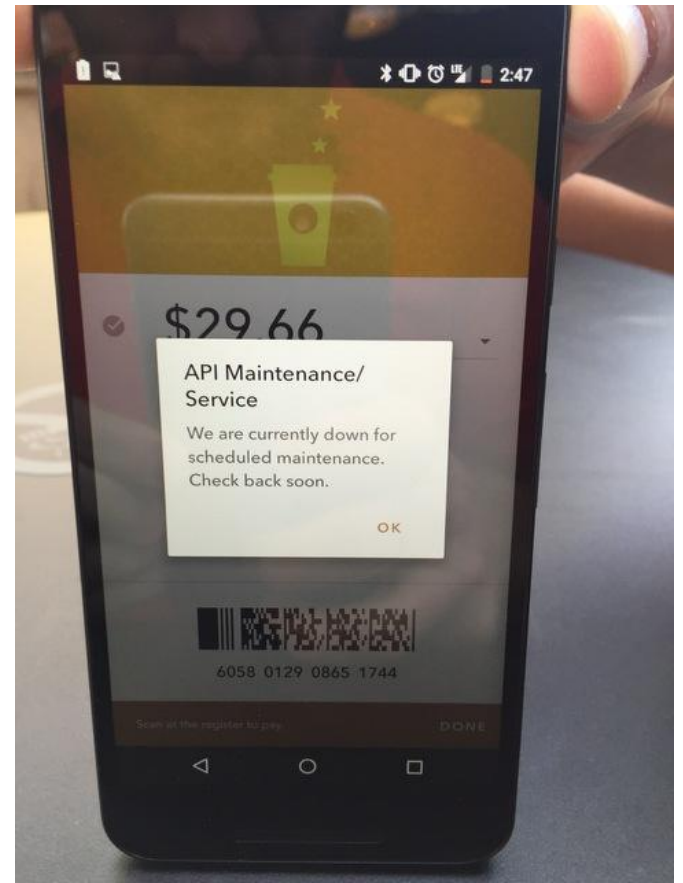
If you have any questions or concerns, please visit our API Developer group or <http://help.untappd.com>.

Thanks,
The Untappd Team



Reply

Just now you add SSL??



Not living this API Contract can have pretty big impact

Delivering a good API will help the platform

The background of the slide shows the silhouettes of four people sitting at a table in a high-rise building, looking out at a city skyline. A large, prominent dome, likely St. Paul's Cathedral in London, is the central focus of the view. The scene is dimly lit, with the city lights providing a soft glow.

Making

What kind of data can you provide to the public?

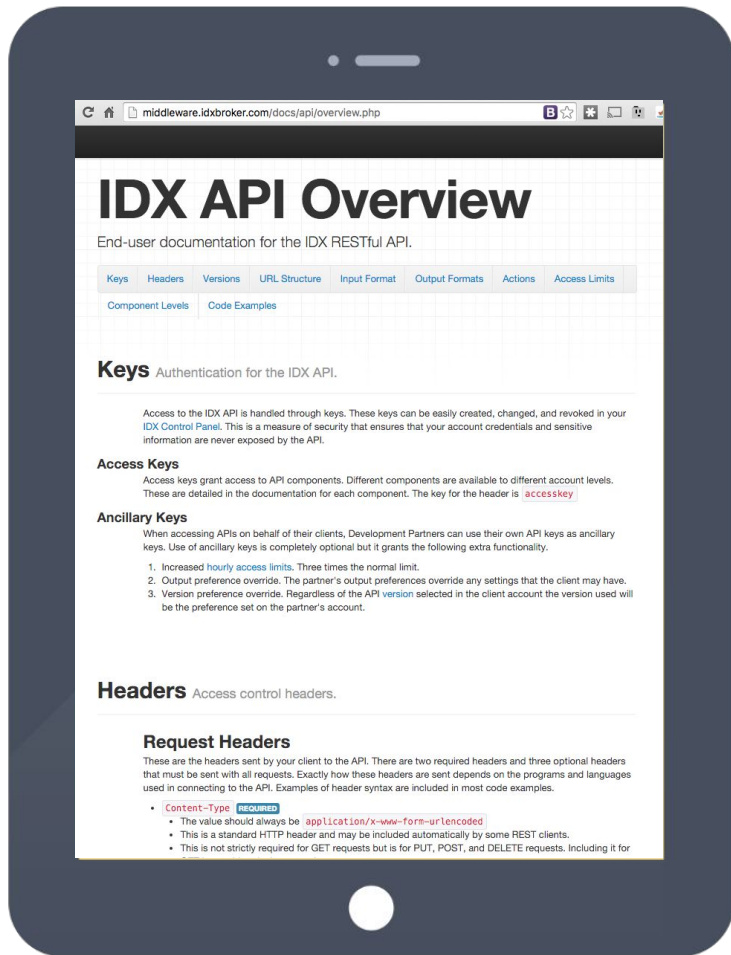
What kind of data can you provide in the future?

The API Documentation

[http://middleware.
idxbroker.
com/docs/api/overview.
php](http://middleware.idxbroker.com/docs/api/overview.php)

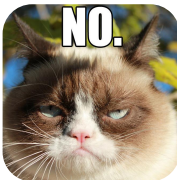
More at developers.
idxbroker.com

• <http://middleware.idxbroker.com/docs/api/overview.p>

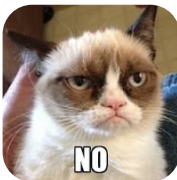


IDX Broker API limitations to keep in mind

The API will NOT



Will NOT return raw MLS data



Will NOT return search results



Will NOT allow you enter MLS data

Required When Calling the IDX Broker API



- A valid API key
- Ability to make requests via https
- Specify Content-Type and API key in your request headers
- Cheesy key graphic (not required)

```
// PHP Example. Featured Listings API call.

$url = 'https://api.idxbroker.com/clients/featured';

$method = 'GET';

// headers (required and optional)

$headers = array(

'Content-Type: application/x-www-form-urlencoded', // required

'accesskey: abcdefghijklmnopqrstuvwxyz', // required - replace with your own

'outputtype: json' // optional - overrides the preferences in our API control page );

// set up cURL

$handle = curl_init();

curl_setopt($handle, CURLOPT_URL, $url);

curl_setopt($handle, CURLOPT_HTTPHEADER, $headers);

curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);

curl_setopt($handle, CURLOPT_SSL_VERIFYHOST, false);

curl_setopt($handle, CURLOPT_SSL_VERIFYPEER, false);

// exec the cURL request and returned information. Store the returned HTTP code in $code for later reference

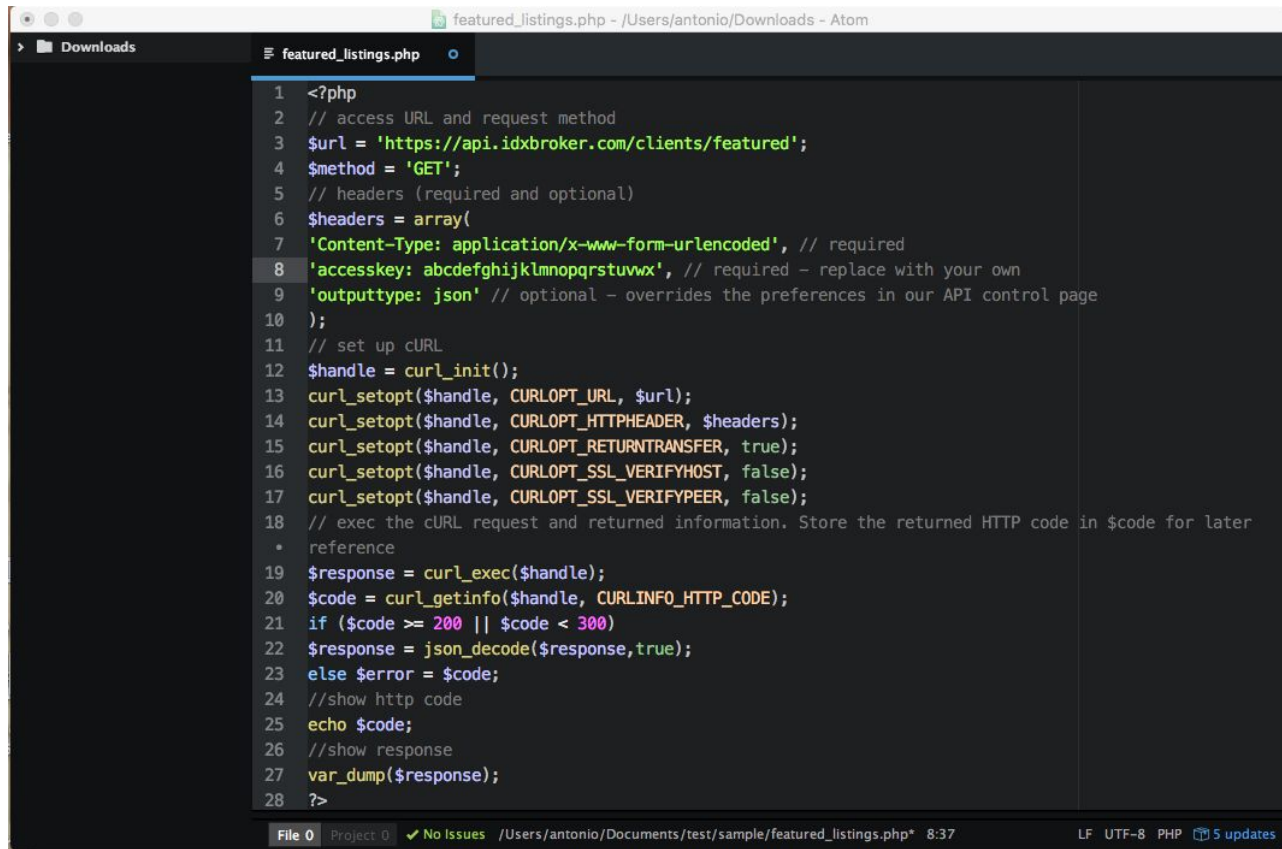
$response = curl_exec($handle);

$code = curl_getinfo($handle, CURLINFO_HTTP_CODE);

if ($code >= 200 || $code < 300)

$response = json_decode($response,true); else $error = $code;
```

Copy sample code to your machine

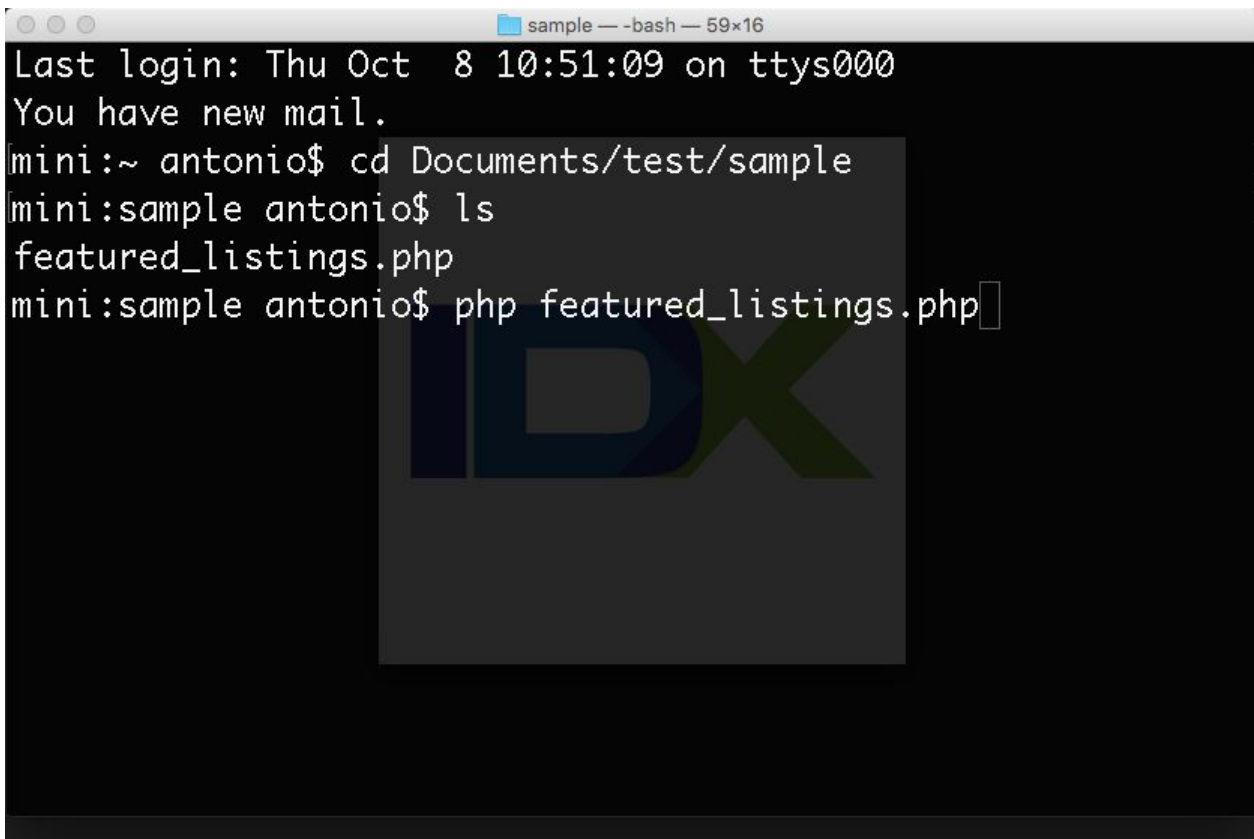


The image shows a screenshot of the Atom code editor. The title bar indicates the file is 'featured_listings.php' located in '/Users/antonio/Downloads'. The left sidebar shows a file explorer with 'Downloads' selected. The main editor area displays the following PHP code:

```
1 <?php
2 // access URL and request method
3 $url = 'https://api.idxbroker.com/clients/featured';
4 $method = 'GET';
5 // headers (required and optional)
6 $headers = array(
7     'Content-Type: application/x-www-form-urlencoded', // required
8     'accesskey: abcdefghijklmnopqrstuvwxyz', // required - replace with your own
9     'outputtype: json' // optional - overrides the preferences in our API control page
10 );
11 // set up cURL
12 $handle = curl_init();
13 curl_setopt($handle, CURLOPT_URL, $url);
14 curl_setopt($handle, CURLOPT_HTTPHEADER, $headers);
15 curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);
16 curl_setopt($handle, CURLOPT_SSL_VERIFYHOST, false);
17 curl_setopt($handle, CURLOPT_SSL_VERIFYPEER, false);
18 // exec the cURL request and returned information. Store the returned HTTP code in $code for later
   • reference
19 $response = curl_exec($handle);
20 $code = curl_getinfo($handle, CURLINFO_HTTP_CODE);
21 if ($code >= 200 || $code < 300)
22     $response = json_decode($response, true);
23 else $error = $code;
24 //show http code
25 echo $code;
26 //show response
27 var_dump($response);
28 ?>
```

The status bar at the bottom shows 'File 0', 'Project 0', 'No Issues', the file path '/Users/antonio/Documents/test/sample/featured_listings.php', the time '8:37', and encoding details 'LF UTF-8 PHP 5 updates'.

Run API Call

A terminal window titled 'sample — -bash — 59x16' with a dark background and light text. It shows a sequence of commands and their outputs. A large, semi-transparent 'IDX' watermark is centered over the terminal content.

```
sample — -bash — 59x16
Last login: Thu Oct  8 10:51:09 on ttys000
You have new mail.
mini:~ antonio$ cd Documents/test/sample
mini:sample antonio$ ls
featured_listings.php
mini:sample antonio$ php featured_listings.php
```

```
sample — -bash — 107x30
[mini:sample antonio$ php featured_listings.php
200array(81) {
  ["a000!%D1382663"]=>
  array(41) {
    ["address"]=>
    string(14) "3701 NE 201 ST"
    ["streetName"]=>
    string(6) "201 ST"
    ["streetNumber"]=>
    string(4) "3701"
    ["streetDirection"]=>
    string(2) "NE"
    ["unitNumber"]=>
    string(0) ""
    ["cityName"]=>
    string(8) "Aventura"
    ["countyName"]=>
    string(10) "Miami-Dade"
    ["state"]=>
    string(7) "FLORIDA"
    ["zipcode"]=>
    string(5) "33180"
    ["zip4"]=>
    string(4) "3066"
    ["listingPrice"]=>
    string(10) "$1,495,000"
    ["listingID"]=>
    string(8) "D1382663"
    ["listingAgentID"]=>
    string(13) "idxdemoagent4"
```

Status code
returned 200

1XX: Informational
2XX: Success
3XX: Redirection
4XX: Client Error
5XX: Server Error

success!

Data
in json key=>
value pairs

We made and API call...



Now what?

**Let's display some of these listings,
with a little more code.**

```
18 // exec the cURL request and returned information. Store the returned HTTP code in $code for
   • later reference
19 $response = curl_exec($handle);
20 $code = curl_getinfo($handle, CURLINFO_HTTP_CODE);
21 //save response
22 $file = 'cache.json';
23 file_put_contents($file, $response);
24
25 ?>
```

I've added some code to my featured_listings.php file to save the API return in a separate place named cache.json

You might use a database for larger projects

Caching the **API** return allows me to work with the returned data without having to make more **API** calls.

This helps load time and keeps me from hitting the **API** hourly limit.

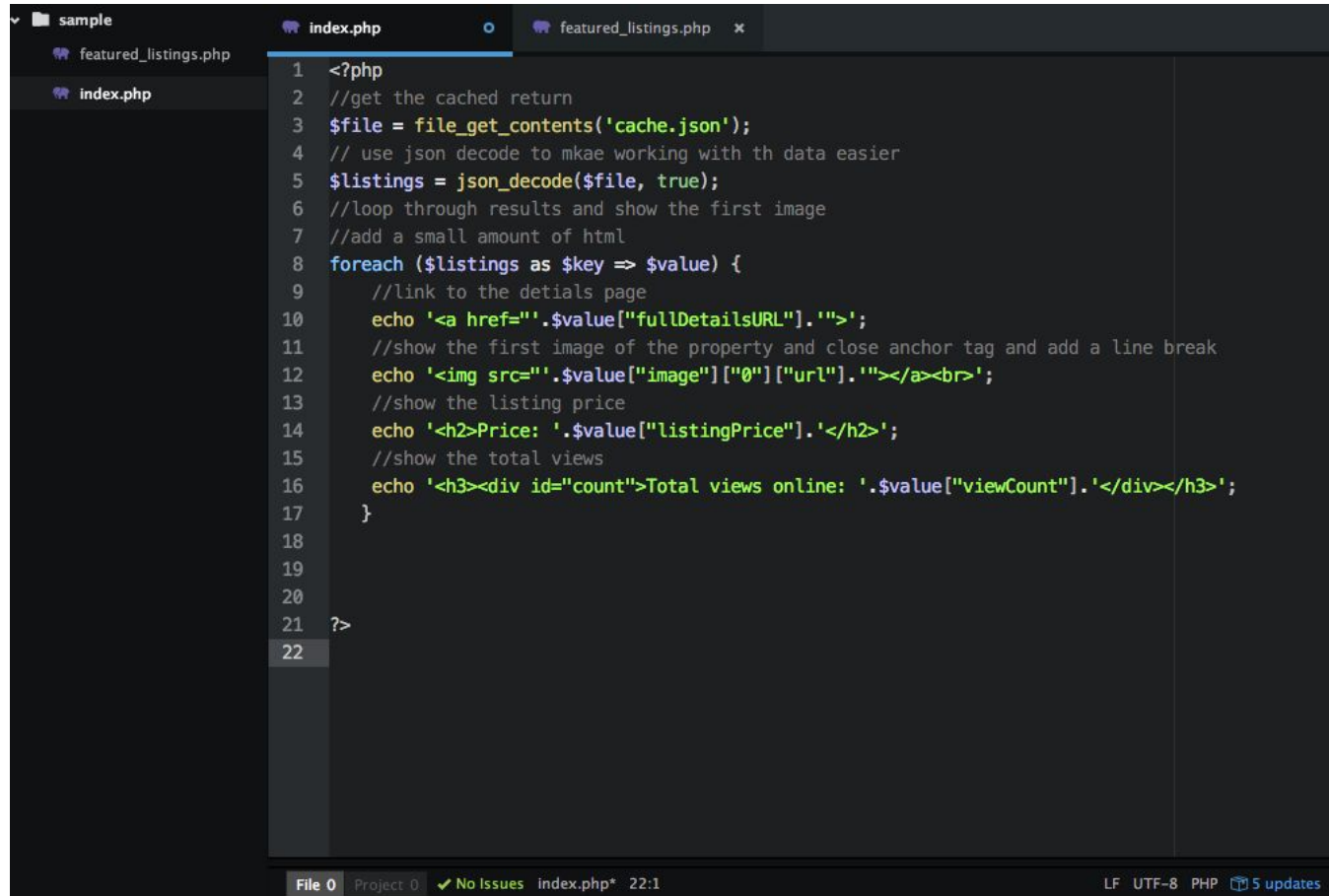


limit = 250 calls per hour



limit = 125 calls per hour

Another file
will get the
cached
return and
use the
data to
make super
simple web
page



```
1 <?php
2 //get the cached return
3 $file = file_get_contents('cache.json');
4 // use json decode to make working with the data easier
5 $listings = json_decode($file, true);
6 //loop through results and show the first image
7 //add a small amount of html
8 foreach ($listings as $key => $value) {
9     //link to the details page
10     echo '<a href="'. $value["fullDetailsURL"].' ">';
11     //show the first image of the property and close anchor tag and add a line break
12     echo '</a><br>';
13     //show the listing price
14     echo '<h2>Price: ' . $value["listingPrice"] . '</h2>';
15     //show the total views
16     echo '<h3><div id="count">Total views online: ' . $value["viewCount"] . '</div></h3>';
17 }
18
19
20
21 ?>
22
```

File 0 Project 0 ✓ No Issues index.php* 22:1 LF UTF-8 PHP 5 updates



Price: \$1,495,000

Total views online: 1113



**This renders out
like this.**

**Image, price, and
total views are
shown.**

Now that I am displaying the data I want I can add



and further customize this page.

Running the featured_listings.
php

daily will keep the data fresh

API Consumption

Should be consistent and easy to consume.

Returns data that is useful.

Now we can use data to make stuffs and things!



Source code for this example available on
GitHub so... Let's all give some code a try!

Visit

10.1.10.229

