

Introduction to REST in PHP

Matt Williams

@technovangelist

m@technovangelist.com

j.mp/itrip-pres0

What Is REST ?

REST - Representational State Transfer

Term coined by Roy Fielding in 2000. One of principal authors of HTTP spec. Cofounder of Apache.

Resource-Oriented Architectural Style

Based on standard HTTP methods: GET, PUT, POST, DELETE

Components of a REST call

Method (verb) + Base URL + Resource (noun) + data

GET: http://api.placester.com/api/v2.1/listings?api_key=APIKEYHERE

GET: [https://app.datadoghq.com/api/v1/dash?
api_key=APIKEYHERE&application_key=APPKEYHERE](https://app.datadoghq.com/api/v1/dash?api_key=APIKEYHERE&application_key=APPKEYHERE)

Two Ways To Think About REST

Consuming

Creating

Exploring REST

Web Browser

Debug Apps

(j.mp/itrip-rested)

cURL

(j.mp/itrip-curl)

Web Browser Tools

Chrome: REST Sniffer, POSTMAN, Advanced REST Client, JSON Formatter

Firefox: REST Client

Consuming in PHP

Streams

(<http://j.mp/itrip-streams>)

Guzzle

(<http://j.mp/itrip-guzzle>)

cURL

(<http://j.mp/itrip-curlphp>)

HTTPFu

(<http://j.mp/itrip-httpful>)

PECL HTTP

About the Sample Code

```
$plsget = 'http://api.placester.com/api/v2.1/listings?api_key=' . $plsapi;  
$ddpost = 'https://app.datadoghq.com/api/v1/events?api_key=' . $ddapi;
```

```
function dd_output($title,$json){  
    echo "<h1>$title</h1>";  
    echo "Status: " . $json['status'] . "<br>";  
    echo "<a href='" . $json['event']['url'] . "' target='_blank'>View</a><br>";  
}
```

```
dd_output( 'POST Datadog Event with Stream', post_stream( $ddpost, $ddparams ) );  
pls_output( 'GET Placester Listings with Stream', get_stream( $plsget ) );
```

<https://github.com/technovangelist/consume-rest-sample>

Consuming with Streams - GET

```
function get_stream($url) {  
    $response = file_get_contents( $url );  
    return json_decode( $response,true );  
}
```

Consuming with cURL - GET

```
function get_curl($url){  
    $curl = curl_init( $url );  
    curl_setopt( $curl, CURLOPT_RETURNTRANSFER, true );  
    $response = curl_exec( $curl );  
  
    return json_decode( $response,true );  
}
```

Consuming with Guzzle - GET

```
function get_guzzle($url){  
    $client = new GuzzleHttp\Client();  
    $response = $client->get( $url );  
  
    return $response->json( );  
}
```

Consuming with Streams - POST

```
function post_stream($url,$params) {  
    $data = json_encode( $params );  
  
    $options = array(  
        'http' => array(  
            'method' => 'POST',  
            'header' => [ 'Content-Type: application/json' ],  
            'content' => $data  
        )  
    );  
    $context = stream_context_create( $options );  
    $response = file_get_contents( $url, false, $context );  
    return json_decode($response,true);  
}
```

Consuming with cURL - POST

```
function post_curl($url, $params){  
    $data = json_encode( $params );  
    $curl = curl_init( $url );  
    curl_setopt( $curl, CURLOPT_CUSTOMREQUEST, 'POST' );  
    curl_setopt( $curl, CURLOPT_POSTFIELDS, $data );  
    curl_setopt( $curl, CURLOPT_RETURNTRANSFER, true );  
    curl_setopt( $curl, CURLOPT_HTTPHEADER, array(  
        'Content-Type:application/json',  
        'Content-Length:' . strlen( $data )  
    ) );  
  
    $response = curl_exec( $curl );  
    return json_decode($response, true);  
}
```

Consuming with Guzzle - POST

```
function post_guzzle($url,$params) {  
    $data = json_encode( $params );  
    $client = new GuzzleHttp\Client();  
    $response = $client->post($url,[ 'body' => $data]);  
    return $response->json();  
}
```

Create a REST Service in PHP

Slim

(<http://j.mp/itrip-slim>)

Flight

(<http://j.mp/itrip-flight>)

Limonade

(<http://j.mp/itrip-limon>)

BulletPHP

Fat-Free Framework (f3)

Silex

Tonic

Wave

About Sample Code

```
function get_listings() {  
    echo 'get all listings';  
}
```

```
function get_listing($listing) {  
    echo 'get listing ' . $listing;  
}
```

```
function add_listing($listing) {  
    echo 'add listing ' . $listing;  
}
```

```
function update_listing($listing) {  
    echo 'update listing ' . $listing;  
}
```

<https://github.com/technovangelist/create-rest-server-sample>

Create a REST Service in Flight

```
require_once 'flight/Flight.php';
```

```
Flight::route( 'GET /flight/listings', 'get_listings' );  
Flight::route( 'GET /flight/listings/@listing', 'get_listing' );  
Flight::route( 'POST /flight/listings/@listing', 'add_listing' );  
Flight::route( 'PUT /flight/listings/@listing', 'update_listing' );
```

```
Flight::start( );
```

Create a REST Service in Slim

```
require_once 'vendor/autoload.php';
```

```
$slim = new \Slim\Slim();
```

```
$slim->get( '/slim/listings', 'get_listings' );
```

```
$slim->get( '/slim/listings/:listing', 'get_listing' );
```

```
$slim->post( '/slim/listings/:listing', 'add_listing' );
```

```
$slim->put( '/slim/listings/:listing', 'update_listing' );
```

```
$slim->run( );
```

Create REST Service with Limonade

```
require_once 'vendor/autoload.php';
```

```
dispatch( '/limon/listings', 'get_listings' );  
dispatch( '/limon/listings/:listing', 'get_listing' );  
dispatch_post( '/limon/listings/:listing', 'add_listing' );  
dispatch_put( '/limon/listings/:listing', 'update_listing' );  
  
run( );
```

What About WordPress and REST?

Coming in 4.1 (more likely 4.2)

Currently available as a plugin (wp-api)

Also easy install with Vagrant - <http://j.mp/itrip-wpapi>

What Resources Available?

POSTS - create, retrieve, edit, delete, create meta, retrieve revisions

MEDIA - create attachment, get attachments

USERS - create, retrieve, edit, delete

TAXONOMIES - retrieve, retrieve terms

<http://j.mp/itrip-wpapi2>

Questions?

Matt Williams

@technovangelist

m@technovangelist.com

j.mp/itrip-pres0