

Peter Gustafson writing sample
ShipEngine
July 19, 2019

Introduction to REST

The glue that makes ShipEngine APIs deliver world-class shipping services is built with REST (REpresentational State Transfer). Don't worry, REST is easy to learn once you realize how it's already part of your daily life.

Below are some examples of REST APIs you probably come in contact with:

- Facebook's REST APIs syndicate your new posts to your friends and other users in your network.
- Uber notifies curbside customers with up-to-date information when their driver will arrive at their location using Twillio's SMS REST API.
- Flickr REST APIs help users embed images on their website and social media profiles.
- Pokémon's REST API lets developers build mobile applications to retrieve updates on all of the Pokémon characters such as their moves, abilities, and even egg groups.

REST is a way for one software system to communicate with another, but behind the scenes. REST is like using a website to get information except it doesn't require the website. Instead, REST commands are done using code requests and responses. A request asks to get something back from a resource.

Resources

With REST APIs, a resource is an object with a type. It contains data that's connected to other pieces of data. Resources can have relationships to other resources.

HTTPS

REST communicates using HyperText Transfer Protocol Secure (HTTPS) methods to keep your calls to ShipEngine APIs safe and secure. HTTPS is a combination of [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) + [TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security). ShipEngine requires you to use [TLS 1.1](https://en.wikipedia.org/wiki/Transport_Layer_Security#TLS_1.1) or higher.

Some API providers use HTTP (no S), however, it's not secure. If you try making calls to our APIs using HTTP, they won't work.

HTTPS sends data between the client and the server using an encrypted SSL certificate to keep all transmissions secure and safe.

![shipping 101](<http://www.pdgseo.com/wp-content/uploads/2019/07/>)

http.png)

Typing `https` into a browser's address bar tells the browser to make the HTTPS request to a resource on a given server.

| Entity | Description |
|------------|---|
| Client | A computer or cell phone making a request for a data resource. |
| User-agent | A tool to make requests such as a web browser like Chrome, Firefox, Safari, or Internet Explorer. |
| Server | One or many computers storing the resource. |

HTTP Methods

In the world of REST APIs, HTTP methods are also called verbs. They're used to communicate between your client (for example, your iPhone) and the servers. Below are the common methods ShipEngine supports:

| Method | Description |
|--------|--|
| GET | Requests access to information or a resource. |
| POST | Creates a new resource in a collection of resources. |
| PUT | Tells the server to replace a resource. |
| DELETE | Notifies the server to delete a resource. |
| PATCH | Updates a resource of collection of resources. |

Endpoints

To make a request, you need to know where it's being sent right? That's where endpoints come in. For example, when you send a letter with the United States Postal Service (USPS), the envelope is addressed to someone. Endpoints are the same thing but formatted with a Uniform Resource Locator (URL) like the following example:

```
https://shipengine.com/shipping
  ^   ^           ^
  |   |           |
scheme root-endpoint path
```

HTTP Headers

HTTP headers validate that the request is coming from an authorized user. They also provide metadata about the message body. The name of the header is separated from the value by a colon.

Common Headers

It can be confusing to understand headers without real-world examples. So let's look at some examples of common headers and what they do.

![enter image description here](http://www.pdgseo.com/wp-content/uploads/2019/07/rest-apis-search.png)

When you do a Google search for REST APIs, the browser makes a request to a Google server asking it to return the results your browser is looking for in the request header.

The `Accept` header tells the Google server what file formats (MIME-type) the browser wants back to display on the page. For example, the server can send plain text, HTML, JSON, etc. Below is an example of Firefox's Accept header:

```
GET /page/routing-in-recess-screencast HTTP/1.1
Host: RecessFramework.org
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

The `Content-Type` header asks for details about the media format of the resource like in the example below:

```
Content-Type: text/html; charset=utf-8
Content-Type: multipart/form-data; boundary=something
```

The `Authorization` request header contains the user credentials to authenticate a user agent with an API key like below:

```
GET /something HTTP/1.1
X-API-Key: abcdef12345
```

Requests

A request passes the client information to the server. It's a text record sent by a web browser to a server. It contains details about the information the browser wants returned in the server response. Learn about [The Anatomy of an HTTP Request](<https://www.shipengine.com/the-anatomy-of-an-http-request/>).

Here's an example request to get tracking information on a single shipment:

```
curl 'https://api.shipengine.com/v1/tracking?
carrier_code=stamps_com&tracking_number=9405511899223197428490' \
-X GET \
-H "api-key: ElJkhJuQIRoFq/kDEblco4LpZqRCdYNIoAVG7SywSXw"
```

Below are the definitions of the request elements in order:

| Object | Description |
|---------------------------|---|
| `curl` | A [command line tool](http://curl.haxx.se/) executes the HTTP request. |
| `v1` | The version number of the API. |
| `tracking` | The ShipEngine tracking API name. |
| `carrier_code=stamps_com` | The shipping carrier code (stamps.com). |

```
| `tracking_number` | The request to track a shipment. |
| `9405511899223197428490` | The tracking number of the shipment. |
| `-X` | Means you're passing a custom HTTP request method. |
| `-H` | The header parameter. |
| `api-key:` | The API key assigned to your ShipEngine account:
`ELJkhJuQIRoFq/kDEblco4LpZqRCdYNIoAVG7SywSXw` |
```

Responses

ShipEngine responses are returned using JavaScript Object Notation (JSON). It's a lightweight data-interchange format. Below is the response example from the previous request:

```
```\n{\n  "tracking_number": "9405511899223197428490",\n  "status_code": "DE",\n  "status_description": "Delivered",\n  "carrier_status_code": "01",\n  "carrier_status_description": "Your item was delivered in or at the mailbox at 9:10 am on March 2, 2017 in AUSTIN, TX 78756.",\n  "ship_date": "2019-03-01T15:57:50.830Z",\n  "estimated_delivery_date": null,\n  "actual_delivery_date": "2019-03-01T15:57:50.830Z",\n  "exception_description": null,\n  "events": [\n    {\n      "occurred_at": "2019-03-01T15:57:50.830Z",\n      "description": "Delivered, In/At Mailbox",\n      "city_locality": "AUSTIN",\n      "state_province": "TX",\n      "postal_code": "78756",\n      "country_code": "",\n      "company_name": "",\n      "signer": ""\n    }\n  ]\n}\n```\n
```

| Object                       | Description                                           |
|------------------------------|-------------------------------------------------------|
| --: --                       |                                                       |
| `tracking_number`            | The shipment tracking number.                         |
| `status_code`                | The status of the shipment delivery.                  |
| `status_description`         | The description of the delivery.                      |
| `carrier_status_code`        | The carrier status code (stamps.com).                 |
| `carrier_status_description` | The details of the carrier shipment.                  |
| `ship_date`                  | The date stamp of the delivery.                       |
| `estimated_delivery_date`    | The schedule estimated delivery date.                 |
| `actual_delivery_date`       | The date the shipment was delivered to the recipient. |
| `exception_description`      | Denotes exception details of the delivery.            |

|                  |  |                                             |  |
|------------------|--|---------------------------------------------|--|
| `occurred_at`    |  | The carrier delivery date & time.           |  |
| `description`    |  | The location where the delivery was made.   |  |
| `city_locality`  |  | The delivery address city.                  |  |
| `state_province` |  | The delivery address state or province.     |  |
| `postal_code`    |  | The delivery address city zip code.         |  |
| `country_code`   |  | The delivery address country code.          |  |
| `company_name`   |  | The company delivery address.               |  |
| `signer`         |  | The person's name signing for the delivery. |  |

**\*\*HTTP Status Codes\*\***

ShipEngine uses [HTTP response status codes](https://en.wikipedia.org/wiki/List\_of\_HTTP\_status\_codes) to define the success or failure of your API requests. If your request fails, ShipEngine returns an error using a status code.

Below are the common ShipEngine error codes:

| Status Code | Description           | Reason                                                                                                                                                            |
|-------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 200         | Success               | The HTTP request was successful.                                                                                                                                  |
| 400         | Bad Request           | There's an error with your request, read the `errors` to view the problem.                                                                                        |
| 404         | Not Found             | The resource you requested is not found. For example, `v1/shipments/se-123` is a valid endpoint but `se-123` may not exist.                                       |
| 405         | Not Allowed           | The method used to request the resource isn't supported by ShipEngine.                                                                                            |
| 429         | Too Many Requests     | You've sent too many requests to ShipEngine. <b>**Learn more about rate limits [LINK]**</b> .                                                                     |
| 500         | Internal Server Error | The server cannot process the request. If you receive persistent `500` errors, [submit a ticket](https://help.shipengine.com/hc/en-us/requests/new) with support. |

Need help with a status code? [Submit a ticket](https://help.shipengine.com/hc/en-us/requests/new) to our support team. We'll review the problem and reply with troubleshooting advice.