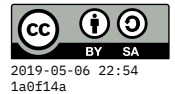


# ReST API



**API** = Application Programming Interface

**ReST** = Representational State Transfer (unhelpful name, but it’s basically an architectural style for web APIs that relies heavily on HTTP and (XML or JSON)).

Any data store must support 4 main operations:

- Create
- Retrieve
- Update
- Delete

These are known as “**CRUD**” operations.

“Resource” is the name for a data entity, identified by a URL/URI.

HTTP = HyperText Transfer Protocol: supports URL/URI and different “methods” on them.

- **GET** (whenever you retrieve a web page) = Retrieve
- **POST** (whenever you submit a form) ~= Create
- **PUT** (or **PATCH**) ~= Update
- **DELETE** = Delete

In ReST, resource URLs are usually structured with methods as:

- /api/RESOURCE/ID

A prefix like /api is optional, but may exist, especially if the same domain also serves regular web content.

Here are how the 5 basic ReST operations are represented:

- **POST** /api/user/ – Create a new user → returns ID
- **GET** /api/user/ – Retrieve a list of users and IDs
- **GET** /api/user/123 – Retrieve details about user 123
- **PATCH** /api/user/123 – (or **PUT**) To update details about user 123
- **DELETE** /api/user/123 – To remove user 123

Data formats are:

- **XML** = eXtensible Markup Language
- **JSON** = JavaScript Object Notation

HTTP status codes:

- **Success**
  - 200 = OK
  - 201 = Created
  - 204 = No Content
- **Redirection**
  - 301/302 = Moved permanently/temporarily
  - 304 = Not modified (client already has latest copy)
- **Client error**
  - 400 = Bad Request (generic)
  - 403 = Forbidden (perhaps needs a login)
  - 404 = Not Found
  - 405 = Method not allowed
  - 428 = Precondition Required (needs some extra data, or perhaps login)
- **Server error**
  - 500 = Internal server error (generic)
  - 502 = Bad gateway (usually misconfigured proxy or server down)

## Sample API

In class, we used the the live demo of the Python Eve framework, running at [eve-demo.herokuapp.com/people](https://eve-demo.herokuapp.com/people)<sup>1</sup>. Below are all the `curl` commands we used.

### Retrieve list

By default, `curl` with a URL uses the **GET** method. When the URL refers to the collective entity (`/people` or `/people/`), it means to retrieve a list.

```
curl -i -w "\n" https://eve-demo.herokuapp.com/people
```

Currently, that produces the following (JSON reformatted and some records omitted, for legibility):

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/json
Content-Length: 1912
X-Total-Count: 4
Cache-Control: max-age=20
```



<sup>1</sup>[eve-demo.herokuapp.com/people](https://eve-demo.herokuapp.com/people)

Expires: Tue, 02 Apr 2019 00:17:18 GMT  
 Last-Modified: Mon, 01 Apr 2019 19:33:29 GMT  
 Server: Eve/0.8 Werkzeug/0.14.1 Python/3.6.5  
 Date: Tue, 02 Apr 2019 00:16:58 GMT  
 Via: 1.1 vegur

```
{ "_items": [
  { "_id": "5c926fc393df890004bb25b8",
    "firstname": "yangping", "lastname": "Red",
    "role": ["copy"],
    "location": {"address": "98 Yatch Road", "city": "San Francisco"},
    "born": "Sun, 20 Jul 1980 11:00:00 GMT",
    "_updated": "Fri, 22 Mar 2019 08:50:22 GMT",
    "_created": "Wed, 20 Mar 2019 16:52:18 GMT",
    "_etag": "4a5b8fb948b355ca4bcf94c879dbb33ba4fb5d3c",
    "_links": {"self": {"title": "person", "href": "people/5c926fc393df890004bb25b8"}},
  },
  { "_id": "5c926fc393df890004bb25b9",
    "firstname": "Anne",
    "lastname": "White",
    "role": ["contributor", "copy"],
    "location": {"address": "32 Joseph Street", "city": "Ashfield"},
    "born": "Fri, 25 Sep 1970 10:00:00 GMT",
    "_updated": "Wed, 20 Mar 2019 16:52:18 GMT",
    "_created": "Wed, 20 Mar 2019 16:52:18 GMT",
    "_etag": "17dda0e8aa997061f98eb6cccb108e1334b15778",
    "_links": {"self": {"title": "person", "href": "people/5c926fc393df890004bb25b9"}},
  },
  ...MORE...
],
  "_links": {
    "parent": {"title": "home", "href": "/"},
    "self": {"title": "people", "href": "people"}
  },
  "_meta": {"page": 1, "max_results": 25, "total": 4}
}
```

### Retrieve individual item

This also uses a **GET**, so the only difference is that the URL contains an ID.

```
curl -i -w "\n" https://eve-demo.herokuapp.com/people/5c926fc393df890004bb25b8
```

```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/json
Content-Length: 527
Cache-Control: max-age=20
Expires: Tue, 02 Apr 2019 00:24:30 GMT
Etag: "4a5b8fb948b355ca4bcf94c879dbb33ba4fb5d3c"
Last-Modified: Fri, 22 Mar 2019 08:50:22 GMT
Server: Eve/0.8 Werkzeug/0.14.1 Python/3.6.5
Date: Tue, 02 Apr 2019 00:24:10 GMT
Via: 1.1 vegur

```

```

{"_id": "5c926fc393df890004bb25b8",
 "firstname": "yangping",
 "lastname": "Red",
 "role": ["copy"],
 "location": {"address": "98 Yatch Road", "city": "San Francisco"},
 "born": "Sun, 20 Jul 1980 11:00:00 GMT",
 "_updated": "Fri, 22 Mar 2019 08:50:22 GMT",
 "_created": "Wed, 20 Mar 2019 16:52:18 GMT",
 "_etag": "4a5b8fb948b355ca4bcf94c879dbb33ba4fb5d3c",
 "_links": {
   "parent": {"title": "home", "href": "/"},
   "self": {"title": "person", "href": "people/5c926fc393df890004bb25b8"},
   "collection": {"title": "people", "href": "people"}
 }
}

```

### Delete item

To delete, we must specify the method using the `-X` argument, and this service also requires that we provide the etag (versioning hash) of the latest version of the record. For that, we use the `-H` argument to specify an If-Match header.

```

curl -i -w "\n" -X DELETE \
  -H 'If-Match: "ecf93b8fdea9730a608c3414c3de15d1157b406c"' \
  https://eve-demo.herokuapp.com/people/5ca264b52eb8d500043810c7

```

If it returns a status code beginning with a 2, that indicates success. 204 is used because there's no other content for it to give us, besides the success acknowledgment.

```

HTTP/1.1 204 NO CONTENT
Connection: keep-alive
Content-Type: application/json
Content-Length: 2

```

```
Server: Eve/0.8 Werkzeug/0.14.1 Python/3.6.5
Date: Tue, 02 Apr 2019 00:28:37 GMT
Via: 1.1 vegur
```

### Create new item

For this one, I'm going to create a file `newuser.json` where I put the new user data, like the following. Be very careful about JSON syntax! If any quotes are missing or mismatched, or there are extra commas, the server will likely signal an error.

```
{
  "firstname": "Olivia",
  "lastname": "Chan",
  "role": ["copy"],
  "location": {"address": "111 E. 4th Ave", "city": "Gotham"},
  "born": "Mon, 31 Mar 1997 11:00:00 GMT"
}
```

Then I reference that file from a `-d` option to the `curl` command. The `-d` automatically switches the default method from **GET** to **POST**. We also need the `-H` option to tell the server that the content is JSON format (not XML or something else).

```
curl -i -w "\n" -d @newuser.json -H "Content-Type: application/json" \
  https://eve-demo.herokuapp.com/people/
```

The reply from the server includes the meta-data fields (created, updated, etag, id, ...).

```
HTTP/1.1 201 CREATED
Connection: keep-alive
Content-Type: application/json
Content-Length: 279
Location: http://eve-demo.herokuapp.com/people/5ca2af346f54900004402d91
Server: Eve/0.8 Werkzeug/0.14.1 Python/3.6.5
Date: Tue, 02 Apr 2019 00:39:16 GMT
Via: 1.1 vegur
```

```
{
  "_updated": "Tue, 02 Apr 2019 00:39:16 GMT",
  "_created": "Tue, 02 Apr 2019 00:39:16 GMT",
  "_etag": "220a2a32ff4a962fdfa10b9953a4bd20626c4acb",
  "_id": "5ca2af346f54900004402d91",
  "_links": {
    "self": {"title": "person", "href": "people/5ca2af346f54900004402d91"}
  },
  "_status": "OK"
}
```

**Update item**

Finally, to update selected fields in an existing user, you can specify just those. Let's say Olivia wants to include her middle name, and she moved from 4th Avenue to 5th Avenue. Put those changes into a file `moduser.json`:

```
{ "firstname": "Olivia Lynn",
  "location": { "address": "111 E. 5th Ave" }
}
```

Pass them using `-d` as before, but this time specify the ID in the URL, the etag using the `If-Match` header, and ask it to use **PATCH** instead of **POST**:

```
curl -i -w "\n" -X PATCH -d @moduser.json \
  -H "Content-Type: application/json" \
  -H 'If-Match: "220a2a32ff4a962fdfa10b9953a4bd20626c4acb"' \
  https://eve-demo.herokuapp.com/people/5ca2af346f54900004402d91
```

Here's the reply:

```
HTTP/1.1 422 UNPROCESSABLE ENTITY
Connection: keep-alive
Content-Type: application/json
Content-Length: 64
Server: Eve/0.8 Werkzeug/0.14.1 Python/3.6.5
Date: Tue, 02 Apr 2019 00:44:52 GMT
Via: 1.1 vegur
```

```
{ "_issues": { "firstname": "max length is 10"}, "_status": "ERR" }
```

Interesting! There's a validation requirement in the API that the `firstname` has a max length of 10 characters! (That's why I receive so much junk mail addressed to Christophe rather than Christopher.) So let's change `moduser.json` to just use the middle initial:

```
{ "firstname": "Olivia L.",
  "location": { "address": "111 E. 5th Ave" }
}
```

Now the reply will be:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/json
Content-Length: 279
Etag: "b7a57d2780b3ace05448298e418f0352a8037d7b"
Server: Eve/0.8 Werkzeug/0.14.1 Python/3.6.5
Date: Tue, 02 Apr 2019 00:47:24 GMT
```

Via: 1.1 vegur

```
{ "_id": "5ca2af346f54900004402d91",
  "_updated": "Tue, 02 Apr 2019 00:47:24 GMT",
  "_created": "Tue, 02 Apr 2019 00:39:16 GMT",
  "_etag": "b7a57d2780b3ace05448298e418f0352a8037d7b",
  "_links": {
    "self": {"title": "person", "href": "people/5ca2af346f54900004402d91"}
  },
  "_status": "OK"
}
```

Success! It gives us the new etag and the updated timestamp.