Interaction design with HTTP

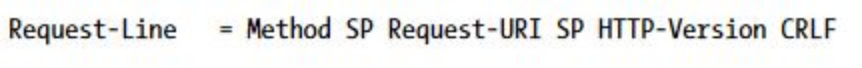
HTTP 1.1

REST APIs embrace all aspects of HTTP 1.1. This includes request methods, response codes and message headers.

Request methods

Request methods

Clients specify desired interaction method in request line part of request message. RFC 2616 defines request line syntax as:



```
Request-Line   = Method SP Request-URI SP HTTP-Version CRLF
```

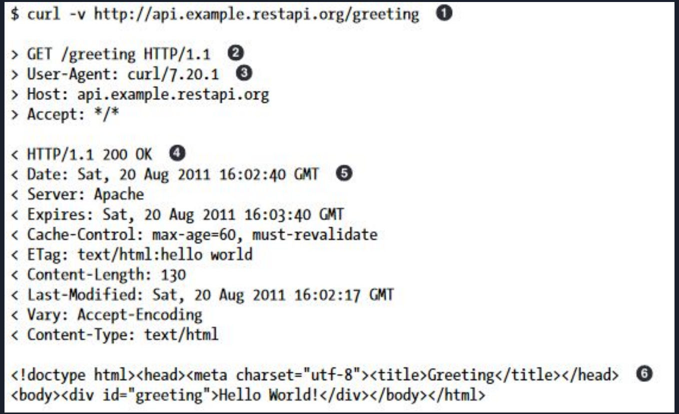
HTTP methods have well defined semantics in a REST API’s resource model.

GET and POST must not be used to tunnel other request methods

- Tunneling is misrepresenting a message’s intent.
- Don’t accommodate clients with limited HTTP vocabulary.
- Make proper use of HTTP methods as specified by this rules.

GET must be used to retrieve a representation of a resource

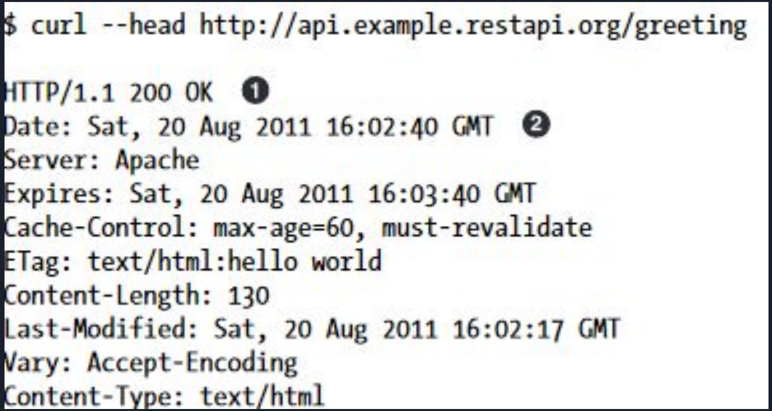
- A client uses the GET method to retrieve the state of a resource in some representational form.
- A GET request message may contain headers and no body.
- Clients count on being able to repeat GET requests without causing side effects.



```
$ curl -v http://api.example.restapi.org/greeting  ❶

> GET /greeting HTTP/1.1  ❷
> User-Agent: curl/7.20.1  ❸
> Host: api.example.restapi.org
> Accept: */*

< HTTP/1.1 200 OK  ❹
< Date: Sat, 20 Aug 2011 16:02:40 GMT  ❺
< Server: Apache
< Expires: Sat, 20 Aug 2011 16:03:40 GMT
< Cache-Control: max-age=60, must-revalidate
< ETag: text/html:hello world
< Content-Length: 130
< Last-Modified: Sat, 20 Aug 2011 16:02:17 GMT
< Vary: Accept-Encoding
< Content-Type: text/html

<!doctype html><head><meta charset="utf-8"><title>Greeting</title></head>  ❻
<body><div id="greeting">Hello World!</div></body></html>
```

HEAD should be used to retrieve response headers

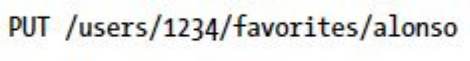
- Clients use HEAD to retrieve the headers without a body.
- HEAD returns the same response as GET but with an empty body.
- Can be used to check if a resource exists or read its metadata.
- Request may contain headers but no body.



```
$ curl --head http://api.example.restapi.org/greeting

HTTP/1.1 200 OK  ❶
Date: Sat, 20 Aug 2011 16:02:40 GMT  ❷
Server: Apache
Expires: Sat, 20 Aug 2011 16:03:40 GMT
Cache-Control: max-age=60, must-revalidate
ETag: text/html:hello world
Content-Length: 130
Last-Modified: Sat, 20 Aug 2011 16:02:17 GMT
Vary: Accept-Encoding
Content-Type: text/html
```

PUT must be used to both insert and update a stored resource

- PUT must be used to add a new resource to a store, with a URI specified by the client.
- PUT must also be used to update or replace an already stored resource.
- Body of a PUT request could differ from the one that a client would receive from a GET request.



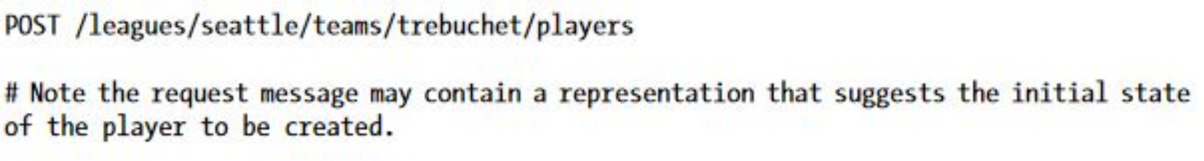
```
PUT /users/1234/favorites/alonso
```

PUT must be used to update mutable resources

- Clients must use the PUT request method to make changes to resources.
- The PUT request message may include a body that reflects the desired changes.

POST must be used to create a new resource in a collection

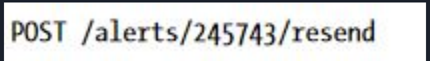
- Clients use POST when attempting to create a new resource within a collection.
- Request body contains suggested state representation of new resource to be added.



```
POST /leagues/seattle/teams/trebuchet/players

# Note the request message may contain a representation that suggests the initial state
of the player to be created.
```

POST must be used to execute controllers

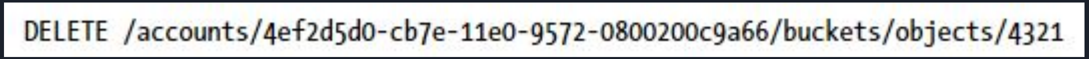
- Clients use the POST method to invoke the function-oriented controller resources.
- Request message may include headers and body as inputs to a controllers resource function.
- POST method can take any action, regardless of repeatability or side effects.
- POST is used to trigger all operations that cannot be intuitively mapped to other core http methods.
- HTTP calls POST unsafe. It’s outcome is unpredictable and not guaranteed to be repeatable.
- Controller resources trade some transparency and robustness for flexibility.



```
POST /alerts/245743/resend
```

DELETE must be used to remove a resource from its parent

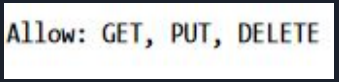
- A client uses DELETE to request that a resource be completely removed from its parent.
- After a DELETE request, resource can no longer be found by clients.
- DELETE method should not be overloaded or stretched.



```
DELETE /accounts/4ef2d5d0-cb7e-11e0-9572-0800200c9a66/buckets/objects/4321
```

OPTIONS should be used to retrieve metadata that describes a resource's available interactions

- Clients may use the OPTIONS request method to retrieve resource metadata that includes an Allow header value.
- May include a body with further details about each option.



```
Allow: GET, PUT, DELETE
```