

Representational State Transfer (REST)

Dr. Roy Fielding, dissertation in 2000:

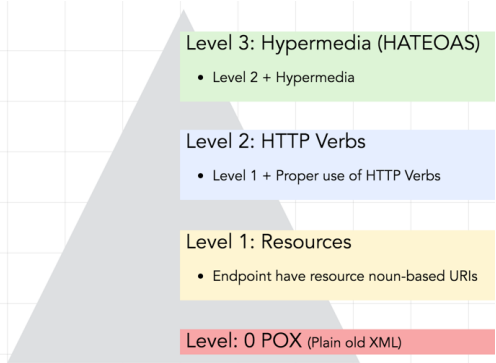
- **Client-Server:** client sends request, server sends response
- **Stateless:** communication stateless, no server state (session)
- **Cache:** responses labeled as cacheable or not
- **Uniform Interface:** resources, verbs, representations, hypermedia
- **Layered System:** components know about own layer
- **Code-on-demand:** Extend client functionality on demand

The Uniform Interface

REST is an architecture style:

- Identification of resources - Use the URI to identify a resource
- Representations of resources - Clients consume representations of resources (eg JSON) not the resource itself
- HATEOAS (Hypermedia as the engine of application state) - Use hyperlinks to decouple the client from application's URL structure

RESTful Richardson's Maturity Model



Distilled

**Resources** paths are nouns like `/products`. And by convention, paths are **plural**, lowercase, and multi-words paths use kebab-case  
**Actions** are defined by HTTP verbs: GET, POST, PUT, (PATCH) and DELETE

Safe and Idempotent Methods [2]

**Safe methods** never modify data on the server

**Idempotent methods** can modify data on the server the first time.

Here's a partial list:

Method	Action	Safe	Idempotent
GET	Read a resource	✓	✓
HEAD		✓	✓
PUT	Replace or update a resource	X	✓
PATCH	Partial update of resource	X	✓
DELETE	Delete resource	X	✓
POST	Create a resource	X	X

GET - Read all or a list of resources

Used to retrieve a list of resources. Request can often pass values to filter, sort and paginate results.  
Server responds with list of partial or whole resources.  
- If resources are relatively large, then return a partial representation  
- If resources are relatively small, then return the entire representation

Request

```
GET /todos?done=true HTTP/1.1
Host: api.example.com
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Connection: keep-alive
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 152
```

```
[{"_id": "59a10", "name": "Buy Milk", "done": "false"},
{"_id": "59a12", "name": "Buy Bread", "done": "false"},
{"_id": "59a15", "name": "Do Dishes", "done": "false"}]
```

GET - Read a single resource

Used to retrieve a single resource.  
Server responds with representation of resource

Request

```
GET /todos/35
Host: api.example.com
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 49
```

```
{"_id": "59a15", "name": "Do Dishes", "done": "false"}
```

DELETE - Delete a resource

Used to delete or remove a resource.  
Server responds an empty body, no content

Request

```
DELETE /v1/todos/59a15 HTTP/1.1
Host: api.example.com
Accept: application/json
cache-control: no-cache
```

Response

```
HTTP/1.1 204 No Content
Access-Control-Allow-Origin: *
Content-Length: 0
```

POST - Create a resource

Used to create a new resource, typically responds with newly created resource along with the new ID and Location header

Request

```
POST /todos HTTP/1.1
Host: api.example.com
Accept: application/json
Authorization: Bearer eyJhb...3ImCAkpXCmeNRVI91E
Content-type: application/json
Content-Length: 42
```

```
{ "name": "Buy Milk", "completed": "false" }
```

Response

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
Location: http://example.com/todos/59a10
Content-Type: application/json; charset=utf-8
Content-Length: 99
```

```
{"_id": "59a10", "name": "Buy Milk", "done": "false"}
```

PUT - Replace/Update a resource

Used to replace or update a resource.  
Server responds with representation of updated resource or no content

Request

```
PUT /v1/todos/59a15 HTTP/1.1
Host: api.example.com
Accept: application/json
Authorization: Bearer eyJhb...3ImCAkpXCmeNRVI91E
content-type: application/json
Content-Length: 37
```

```
{ "name": "Vacuum", "done": "false" }
```

Response

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 46
```

```
{"_id": "59a15", "name": "Vacuum", "done": "false"}
```

HTTP Statuses

- 200 OK - Request successful. Response processed and sent
- 201 Created - Resource successfully created
- 204 No content - Request successful, no content sent in response
- 301 Moved permanently - Resource moved, request at new URL
- 400 Bad Request - Malformed syntax, unsupported content-type
- 401 Unauthorized - Not authenticated. Bad username password combo
- 406 Forbidden - Authenticated but not authorized, no permissions
- 409 Conflict - Resource already exists
- 422 Unprocessable entity - Syntax OK, but values are incorrect