REST API Best Practices: How to Use the Right HTTP Methods and Status Code

REST API development is very popular today, fulfilling rapid growing of cloud services and apps. You know, one of REST architectural constraints is *Uniform Interface* - stating that developers should use common, well-known HTTP methods and status codes in their APIs, in a way that ensures conformity across the web.
So what is the best practice widely used by the industry? In this article, I'd like to share with you guys how to use the right HTTP methods and status codes in your REST APIs.

1. How to use the Right HTTP Methods for REST APIs

Basically, for CRUD operations (Create, Retrieve, Update and Delete) you can use the HTTP methods as follows:
- POST: create resource or search operation
- GET: read resource operation
- PUT: update resource operation
- DELETE: remove resource operation
- PATCH: partial update resource operation

Let's go into detail of each HTTP method.

API end point with POST method:

Used mainly for Create resource operations, or Read operations in some certain cases:

- Read resource if URL / query string exceeds maximum allowed characters. Maximum length of URL and query string is 2,048 characters.
- Read resource if query parameters contain sensitive information

Return status code:

- ***201 Created*** for successful create operation
- ***200 OK***for successful read operation if the response contains data
- ***204 No Content*** for successful read operation if the response contains NO data

API end point with GET method:

Used primary for Read resource operations.
Return status code:

- ***200 OK*** if the response contains data
- ***204 No Content*** if the response contains no data

API end point with PUT method:

Used for Update resource operations.

Return status code:

- ***200 OK*** if the response contains data
- ***204 No Content*** if the response contains no data

API end point with DELETE method:

Used for Delete resource operations.
Return status code: ***204 No Content*** for successful delete operation.

API end point with PATCH method:

Used for Partial update resource operations.
Return status code: ***200 OK***for successful partial update operation.

2. How to use the Right Status Codes for REST APIs

Returning the right HTTP status codes in REST APIs is also important, to ensure uniform interface architectural constraint. Besides the status codes above, below is the guideline for common HTTP status codes:

- ***200 OK***: for successful requests
- ***201 Created***: for successful creation requests
- ***202 Accepted***: the server accepts the request, but the response cannot be sent immediately (e.g. in batch processing)
- ***204 No Content***: for successful operations that contain no data
- ***304 Not Modified***: used for caching, indicating the resource is not modified
- ***400 Bad Request***: for failed operation when input parameters are incorrect or missing, or the request itself is incomplete
- ***401 Unauthorized***: for failed operation due to unauthenticated requests
- ***403 Forbidden***: for failed operation when the client is not authorized to perform
- ***404 Not Found***: for failed operation when the resource doesn't exist
- ***405 Method Not Allowed***: for failed operation when the HTTP method is not allowed for the requested resource
- ***406 Not Acceptable***: for failed operation when the Accept header doesn't match. Also can be used to refuse request
- ***409 Conflict***: for failed operation when an attempt is made for a duplicate create operation
- ***429 Too Many Requests***: for failed operation when a user sends too many requests in a given amount of time (rate limiting)
- ***500 Internal Server Error***: for failed operation due to server error (generic)
- ***502 Bad Gateway***: for failed operation when the upstream server calls fail (e.g. call to a third-party service fails)
- ***503 Service Unavailable***: for a failed operation when something unexpected happened at the server (e.g. overload of service fails)

3. Notes

There’s no strict rule that specifies which method should be used for which operation. However, it’s better to follow some certain rules widely used by the industry, as mentioned above.
And be flexible, you can break the rule in certain cases. It’s not strict requirement.