**REST API: Path vs. Request Body Parameters**

Look at REST API path parameters versus request bodies.

[HTTP Verbs](https://www.signom.com/api/rest/docs/#overview-http-verbs)

RESTful Contracts tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP verbs.

|  |  |
| --- | --- |
| **GET** | Used to retrieve a resource |
| **POST** | Used to create a new resource |
| **PUT** | Used to update an existing resource |
| **DELETE** | Used to delete an existing resource |

[HTTP Status Codes](https://www.signom.com/api/rest/docs/#overview-http-status-codes)

RESTful Contracts tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP status codes.

| **Status code** | **Usage** |
| --- | --- |
| 200 OK | The request completed successfully. |
| 201 Created | A new resource has been created successfully. The resource’s URI is available from the response’s Location header. |
| 204 No Content | An update to an existing resource has been applied successfully. |
| 400 Bad Request | The request was malformed. The response body will include an error providing further information. |
| 404 Not Found | The requested resource did not exist. |
| 500 Internal Server Error | Unexpected condition that could not be handled by the application. |

Whenever we create a [REST API](https://restfulapi.net/), we have to decide which parameter should be present where.

For example, if we are creating a REST API to update student details using PUT ([HTTP Method](https://restfulapi.net/http-methods/)), then the request URI will be {server\_host}/students/{student\_id}, and the request body would be:

1

{

2

"id": student\_id,

3

"name": "student name",

4

"school\_name": "school name"

5

}

I have seen that many times, developers get confused about why we need to send the same parameter to multiple places. For instance, in the above example, we are sending the student\_id to the path parameter as well as the request body. It may seem that we are sending repetitive information via the API, but remember that the request body and path parameters have different meanings and should be used for the purpose that they are going to serve.

Below is the explanation of why we cannot remove student\_id from the path parameters and the request body.

**Path Parameters**

Path parameters are used to identify a resource uniquely. In the {server\_host}/students/{student\_id} example,  student\_id  is identifying a unique student\_id . If we remove student\_id from the path parameter, create {server\_host}/studentsAPI and use student\_id of the request body. Then, on the backend, we can write our logic perfectly fine, but that API will not follow the REST API principle. By looking at the {server\_host}/students API contract, no client would be able to identify that this API is to process the record of only one resource.

**Request Body**

The request body is used to send and receive data via the REST API. If we are using POST/PUT API, then based on the REST API contract, we should send the whole resource information because these methods work on the whole resource. In the above example, student\_id is also part of that resource, so it has to be present in the request body, else the request body would be able to represent the whole resource information.

After removing student\_id from the request body, we will have the below request body.

1

{

2

"name": "student name",

3

"school\_name": "school name"

4

}

Does this request body represent the whole resource information? No. So this is a violation of the REST contract.

So, to represent resource state, we need to send student\_id in the request body, and to identify the resource uniquely, we need to send the student\_id in path parameter.