**REST API Best Practices**

### **Best practices of json / Use JSON as the Format for Sending and Receiving Data**

In the past, accepting and responding to API requests were done mostly in XML and even HTML. But these days, JSON (JavaScript Object Notation) has largely become the de-facto format for sending and receiving API data.

This is because, with XML for example, it's often a bit of a hassle to decode and encode data – so XML isn’t widely supported by frameworks anymore.

JavaScript, for example, has an inbuilt method to parse JSON data through the fetch API because JSON was primarily made for it. But if you are using any other programming language such as Python or PHP, they now all have methods to parse and manipulate JSON data as well.

To ensure the client interprets JSON data correctly, you should set the **Content-Type** type in the response header to **application/json** while making the request.

For server-side frameworks, on the other hand, many of them set the **Content-Type** automatically. Express, for example, now has the **express.json()** middleware for this purpose. The **body-parser** NPM package still works for the same purpose, too.

* **Request response model**

→ In request/response communication mode.

→ One software module sends a request to a second software module and waits for a response.

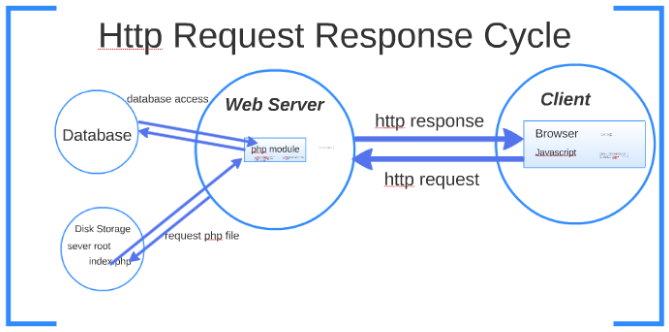
→ The First software module performs the role of the client.

→ The second, the role of the server,

→ This is called client/server interaction.

→ Rest API working flow

→ Rest API auth & security



* **Rest API working flow**

→ [Use nouns instead of verbs in endpoint paths](https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/#h-use-nouns-instead-of-verbs-in-endpoint-paths)

→ [Name collections with plural nouns](https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/#h-name-collections-with-plural-nouns)

→ [Nesting resources for hierarchical objects](https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/#h-nesting-resources-for-hierarchical-objects)

→ [Handle errors gracefully and return standard error codes](https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/#h-handle-errors-gracefully-and-return-standard-error-codes)

→ [Allow filtering, sorting, and pagination](https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/#h-allow-filtering-sorting-and-pagination)

→ [Maintain Good Security Practices](https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/#h-maintain-good-security-practices)

→ [Cache data to improve performance](https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/#h-cache-data-to-improve-performance)

→ [Versioning our APIs](https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/#h-versioning-our-apis)

* **Rest API auth & security**

### →Keep it Simple.

### →Always Use HTTPS

### →Use Password Hash

### →Never expose information on URLs

### →Consider OAuth

### → Consider Adding Timestamp in Request

### → Input Parameter Validation