JSON: API documents are defined in JS object notation. Top level refers to the fact that all JSON: API requests and responses must have a JSON object at its’ root.

A document must have at least one top-level member:

* **Data** – primary, JSON data response
* **Errors** – Array of error objects
* **Meta** – Array of meta information

|  |  |
| --- | --- |
| https://kennbrodhagen.net/2016/03/09/how-to-return-a-custom-error-object-and-status-code-from-api-gateway-with-lambda/response-in-postman.jpg |  |
| **Errors Top-Level Member** | **Meta Top-Level Member** |

Optional top-level members:

* JSONAPI – describes server implementation
* Links - related to (JSON’s) primary data
* Links has its’ own members:
  + Self
  + Related
* Included – array of resource objects

Important Top-Level Member Combinations:

**Data & Errors:** Cannot exist in the same document.

**Data & Included:** The included member cannot exist in a document without the data member.

Primary data always returns as either a **single resource object** or an **array of resource objects**.

|  |
| --- |
| Single Resource Object |

{

"userId": 1,

"id": 1,

"title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",

"body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"

}

|  |
| --- |
| Array of Resource Objects |

[

{

"userId": 1,

"id": 1,

"title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",

"body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"

},

{

"userId": 1,

"id": 2,

"title": "qui est esse",

"body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"

}

]