html

Path Validation

- FastAPI validates that an item_id exists in the request path for GET and PUT requests.
- It ensures the item_id is of type int to prevent invalid operations.
- If item_id isn't provided, a useful error is shown to the client.

Optional Parameters

- The optional parameter must be present in both GET and PUT requests.
- Fully required parameters will cause an error without them being specified.
- 'None' is used for fully required parameters, while 'null' isn't valid.

Query Parameter Handling (GET)

- The query parameter 'q' must be present in GET requests.
- 'None' is used for fully required parameters, while 'null' isn't valid.
- Fully required parameters require their values to be provided.

JSON Parsing Checks

- The JSON object must have a string attribute called 'price' that is a float.
- 'None' (or 'null') is used for fully required attributes, while 'string' is needed.
- 'bool' is allowed and will be converted to 1 or 0 in the response.

Dead-simple Deep Structure

- FastAPI handles deeply nested JSON objects similarly.
- All validations apply recursively within nested structures.
- Trials and tests for the deep structure will be provided in future updates.

These validations ensure FastAPI works efficiently with a variety of request formats, from simple GET requests to deeply nested JSON data. For those less familiar with the intricacies, we've broken them down into digestible sections that highlight key points and potential pitfalls. Let us know how you fare in your API journey!