API Error Handling and Response Codes Reference

Standard Response Codes

HTTP Code	Name	Description
`200`	OK	Request succeeded, response contains requested data.
`201`	Created	Resource created successfully.
`400`	Bad Request	Malformed syntax or invalid request.
`401`	Unauthorized	Authentication failed or missing authentication token.
`403`	Forbidden	Authenticated user does not have permission.
`404`	Not Found	Requested resource does not exist.
`409`	Conflict	Request conflicts with current state of the server.
`422`	Unprocessable Entity	Validation or semantic error in request.
`429`	Too Many Requests	Rate limit exceeded.
`500`	Internal Server Error	Unexpected error on server.
`503`	Service Unavailable	API unavailable or overloaded.

Error Response Structure

- Errors return JSON objects with details.
- Common fields:
  - `error`: Error type or code
  - `message`: Human-readable description
  - `details` (optional): Additional error information

Example Error Response

```
{
  "error": "invalid_request",
  "message": "Parameter 'email' is required.",
  "details": {
    "field": "email"
  }
}
```

Best Practices

- Check HTTP status code to determine result.
- Use `message` for logging and debugging.
- Do not rely on error messages for program logic (use codes/fields).
- Handle unexpected errors gracefully (e.g., `500`).