

良いAPI設計の原則

1. 明確で一貫性のある命名

- エンドポイント、メソッド、パラメータは一貫した命名規則に従うこと。
- 名称は直感的で理解しやすいものにする。

例:

- `/users` (リソース名は複数形)
- `/users/{id}` (詳細なリソースはIDを使用)

2. HTTPメソッドの適切な使用

- **GET**: データの取得
- **POST**: 新しいリソースの作成
- **PUT**: 既存リソースの完全な更新
- **PATCH**: 既存リソースの部分的な更新
- **DELETE**: リソースの削除

3. ステータスコードの適切な使用

- 200 OK: 成功
- 201 Created: リソース作成成功
- 204 No Content: 削除成功
- 400 Bad Request: クライアントエラー
- 401 Unauthorized: 認証エラー
- 404 Not Found: リソース未検出
- 500 Internal Server Error: サーバーエラー

4. 適切なエラーメッセージ

- エラー発生時には、詳細で理解しやすいメッセージを提供。
- エラーコードと併せて返す。

```
{  
  "error": {  
    "code": "invalid_parameter",  
    "message": "The 'email' parameter is required."  
  }  
}
```