

7.API Design & Response Standard

7.1. Mục tiêu chương

Chương này quy định chuẩn API và định dạng response nhằm:

- Đảm bảo tính nhất quán giữa các module
- Dễ tích hợp giữa FE và BE
- Hỗ trợ versioning API trong tương lai
- Giảm rủi ro khi thay đổi cấu trúc dữ liệu

7.2. Khái niệm / phạm vi áp dụng

Quy định này áp dụng cho:

- Tất cả API trong hệ thống
- Core Team và Partner Team
- Tất cả module trong thư mục modules/**

Mọi API phải tuân theo:

- Chuẩn base path
- Chuẩn định dạng response

7.3. Quy định chính

7.3.1. Chuẩn Base Path

Tất cả API phải có dạng:

`/api/v1/<module-name>`

vd: `http://localhost:8080/api/v1/auth/config`

Trong đó:

- `api`: tiền tố chung của hệ thống
- `v1`: version của API
- `<module-name>`: tên module nghiệp vụ

Ví dụ base path hợp lệ

Module	Base path
auth	<code>/api/v1/auth</code>
planning	<code>/api/v1/planning</code>
document	<code>/api/v1/document</code>
user	<code>/api/v1/user</code>

7.3.2. Chuẩn Response

Tất cả API phải trả về:

`ApiResponse<T>`

```
{
  "success": true,
  "data": [],
  "error": null,
  "meta": {
    "timestamp": "2026-02-09T08:02:58.310753500Z"
  }
}
```

Không được trả:

- Entity
- DTO
- List
- Map
- String
- Boolean

trực tiếp ra API.

Cấu trúc chuẩn của ApiResponse

Ví dụ:

```
{
  "success": true,
  "message": "OK",
  "data": {
    "id": 1,
    "name": "Test"
  }
}
```

```
}  
  
}
```

Hoặc với danh sách:

```
{  
  
  "success": true,  
  
  "message": "OK",  
  
  "data": [  
  
    { "id": 1, "name": "A" },  
  
    { "id": 2, "name": "B" }  
  
  ]  
  
}
```

7.4. Cách thực hiện / quy trình

Bước 1: Tạo controller theo base path

Ví dụ:

```
@RestController  
  
@RequestMapping("/api/v1/planning")  
  
public class PlanningController {  
  
}
```

Bước 2: Service trả DTO

```
public PlanningResponse getByld(Long id) {  
    // xử lý nghiệp vụ  
}
```

Bước 3: Controller trả ApiResponse

```
@GetMapping("/{id}")  
public ApiResponse<PlanningResponse> getByld(@PathVariable Long id) {  
    return ApiResponse.ok(planningService.getByld(id));  
}
```

7.5. Ví dụ minh họa

Trường hợp sai

```
@GetMapping("/planning")  
public List<PlanningEntity> getAll() {  
    return planningRepository.findAll();  
}
```

Sai vì:

- Trả entity trực tiếp
- Không dùng ApiResponse
- Bỏ qua service layer

Trường hợp đúng

@GetMapping

```
public ApiResponse<List<PlanningResponse>> getAll() {  
    return ApiResponse.ok(planningService.getAll());  
}
```

Trường hợp sai về base path

@RequestMapping("/planning")

→ Sai vì thiếu:

/api/v1/

7.6. Checklist áp dụng

Trước khi commit hoặc tạo PR:

- API có đúng base path /api/v1/<module> không?
- Controller có trả ApiResponse<T> không?
- Có trả raw object không?
 - Nếu có → phải sửa

- Có trả Entity trực tiếp không?
 - Nếu có → phải chuyển sang DTO
 - API có đúng version (v1) không?
-

Phiên bản #1

Được tạo 2026-02-23 10:22:21 UTC bởi admin_lifetex

Được cập nhật 2026-02-23 10:22:21 UTC bởi admin_lifetex