

3.2 API routes and server validation

Name: _____ Class: _____ Date: _____ **Total: 12 marks**

KEY WORDS route 路由 • validate input 校验输入 • status code 状态码
• request method 请求方法

Objective

Design predictable API routes and validate every request on the server.

You must be able to:

- distinguish GET from POST
- explain routes, input validation, and status codes
- recommend clear responses to invalid requests

1 Worked examples

■ **Make the route match the action**

GET /courses asks for a course list. POST /enrolments asks the server to create an enrolment. A **route** 路由 connects the URL and method to the code that handles it.

The server must **validate input** 校验输入: check required values, types, and ranges. It can return 200 for success, 400 for a bad request, or 404 for something that does not exist.

2 Practice

2.1 State which method is normally used to retrieve a list of courses. [1]

2.2 State what status code 404 means. [1]

2.3 Explain why server validation must check that an age value is a number within an allowed range. [2]

3 Exam-style questions

3.1 A client sends POST /enrolments with no course identifier. Explain an appropriate server response. [4]

3.2 Explain why predictable route names and methods make an API easier for another developer to use. [4]

Solutions

2.1

- GET.

2.2

- The requested resource does not exist.

2.3

- The server needs a value it can safely use.
- It prevents impossible or unsuitable data entering the system.

3.1

- The server detects that the required course identifier is missing.
- It does not create a partial enrolment.
- It returns a clear 400 bad-request response.
- The response tells the client what must be corrected.

3.2

- The route and method communicate what action will happen.
- A developer can predict how to retrieve or create a resource.
- This reduces mistakes and unclear documentation.
- Consistent routes make the API easier to test and maintain.