

Exercise walk-through

3.2

Back-end Programming

GAC Computing

You must be able to

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

1

The method

Method — 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

Practice 2.1 ■ 1 mark

State which method is normally used to retrieve a list of courses.

Solution 2.1

Method: Make the route match the action — p.4

Worked steps

- GET. **B1**

Practice 2.2 ■ 1 mark

State what status code 404 means.

Solution 2.2

Worked steps

- The requested resource does not exist. **B1**

Practice 2.3 ■ 2 marks

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

Solution 2.3

Method: Make the route match the action — p.4

Worked steps

- The server needs a value it can safely use. **B1**
- It prevents impossible or unsuitable data entering the system. **A1**

3

Exam-style

Exam-style 3.1 ■ 4 marks

A client sends `POST /enrolments` with no course identifier.
Explain an appropriate server response.

Solution 3.1

Method: Make the route match the action — p.4

Worked steps

- The server detects that the required course identifier is missing. **B1**
- It does not create a partial enrolment. **A1**
- It returns a clear 400 bad-request response. **A1**
- The response tells the client what must be corrected. **B1**

Exam-style 3.2 ■ 4 marks

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

Solution 3.2

Method: Make the route match the action — p.4

Worked steps

- The route and method communicate what action will happen. **B1**
- A developer can predict how to retrieve or create a resource. **A1**
- This reduces mistakes and unclear documentation. **A1**
- Consistent routes make the API easier to test and maintain. **B1**