

1. successful GET /api/customers request in Postman. The API returns a 200 OK status and a JSON array containing all customers from the database.

The screenshot shows a Postman collection named "GET all customers". A GET request is made to `http://localhost:8080/api/customers`. The response status is 200 OK, time is 334 ms, size is 1.51 KB. The response body is a JSON array of two customer objects:

```

1 [
2   {
3     "id": 1,
4     "customerCode": "C001",
5     "fullName": "John Doe",
6     "email": "john.doe@example.com",
7     "phone": "+1-555-0101",
8     "address": "123 Main St, New York, NY 10001",
9     "status": "ACTIVE",
10    "createdAt": "2025-12-08T16:30:38"
11  },
12  {
13    "id": 2,
14    "customerCode": "C002",
15    "fullName": "Jane Smith",
16    "email": "jane.smith@example.com",
17    "phone": "+1-555-0102",
18    "address": "456 Oak Ave, Los Angeles, CA 90001",
19    "status": "ACTIVE",

```

2. This screenshot illustrates a failed POST /api/customers request with invalid input data. The server responds with a 400 Bad Request status and a structured error body that includes field-level validation messages.

The screenshot shows a POST request to `http://localhost:8080/api/customers`. The response status is 400 Bad Request, time is 156 ms, size is 529 B. The response body is a JSON object with validation errors:

```

1 {
2   "timestamp": "...",
3   "status": 400,
4   "error": "Validation Failed",
5   "message": "Invalid input data",
6   "path": "/api/customers",
7   "details": [
8     {"customerCode": "Customer code must be 3-20 characters"},
9     {"fullName": "Full name is required"},
10    {"email": "Invalid email format"}
11  ]

```

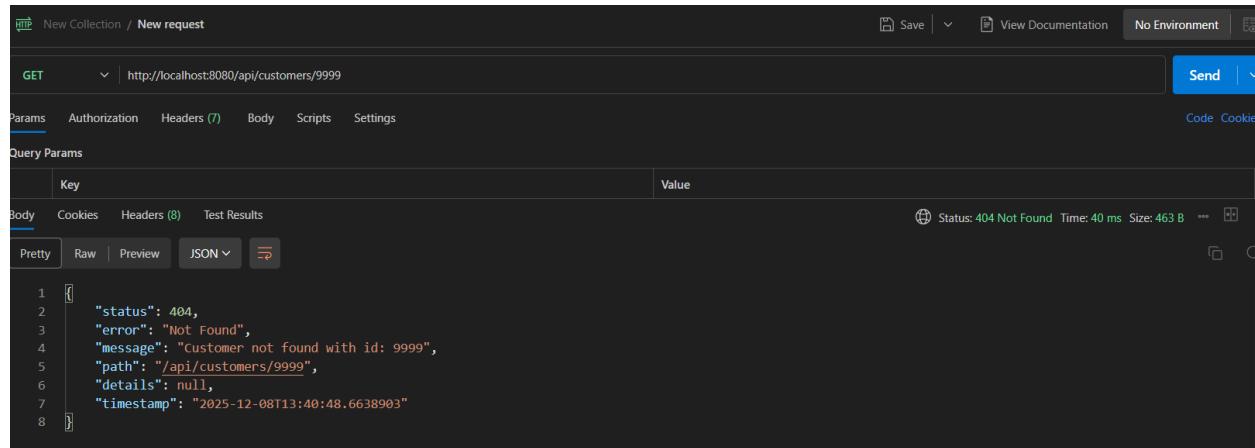
Below the error message, the raw JSON response is shown:

```

1 [
2   "status": 400,
3   "error": "Validation Failed",
4   "message": "Invalid input data",
5   "path": "/api/customers",
6   "details": [ "email: Email is required", "fullName: Full name is required", "customerCode: Customer code is required" ],
7   "timestamp": "2025-12-08T13:38:33.9764975"
8 ]

```

3. This screenshot shows a GET /api/customers/{id} request for a non-existent customer. The API returns a 404 Not Found status along with a JSON error response generated by the global exception handler.



The screenshot shows a Postman request interface. The URL is set to `http://localhost:8080/api/customers/9999`. The response status is `404 Not Found`, and the timestamp is `2025-12-08T13:40:48.6638903`. The response body is a JSON object:

```
1  {
2      "status": 404,
3      "error": "Not Found",
4      "message": "Customer not found with id: 9999",
5      "path": "/api/customers/9999",
6      "details": null,
7      "timestamp": "2025-12-08T13:40:48.6638903"
8 }
```