

Exercise: Building a CRUD API for Products

In this exercise, you'll implement **Create**, **Update**, and **Delete** operations for products. You'll extend the existing `/products` API to allow adding new products, updating existing ones, and deleting them.

1. Create a New Product

Implement an endpoint that allows clients to add a new product by sending a JSON payload in the request body.

Accept a `ProductDTO` containing:

- `name`
- `description`
- `price`
- `categoryId`

Return:

- 201 Created with the created product

2. Update an Existing Product

Implement an endpoint to update an existing product by ID.

- Accept a `ProductDTO` in the request body and update the product.
- Return 404 Not Found if product not found.
- Return 200 OK with the updated product.

Hint: When mapping a `ProductDto` to a `Product` entity, exclude the `id` field from mapping. This is because `id` is a primary key, and Hibernate does not allow primary key values to be changed once an entity is managed.

```
@Mapping(target = "id", ignore = true)
void update(ProductDto productDto, @MappingTarget Product product);
```

3. Delete an Existing Product

Implement an endpoint to delete a product by ID.

- Validate the product ID and return 404 Not Found if it doesn't exist.
- If successful, return 204 No Content with no response body.