

Comprehension Questions

What is a controller in the context of an Express (or similar) web application?

What is the purpose of middleware in an Express app, and how does it relate to checkpoints in the bakery analogy?

What problems can arise if all route logic is kept in a single file without using controllers?

What should an API response look like when validation fails, and why is this important for users or clients?

How does storing orders in a digital list (array) improve the bakery's workflow compared to using sticky notes or paper?

What is one best practice to avoid duplicating logic across multiple controllers?

Why is the order in which middleware is registered in Express important?

Why should error-handling middleware always be registered last in your Express app?

How does chaining multiple middleware functions enforce the correct order of operations in a workflow?

What HTTP status code should be returned if insurance approval is missing, and why?

How does clear error messaging reduce support requests and improve user satisfaction?

Why is it a best practice to validate **all fields** (even optional ones) in a request?