

What is DTO ( Data Transfer Object)? And Why to use in DTO Java Projects?

Data Transfer Object Design Pattern is a frequently used design pattern. It is basically used to pass data with multiple attributes in one shot from client to server, to avoid multiple calls to a remote server.

There are many reasons why we use DTO's and here are some reasons based on my experience:

1. Instead of sending single entity as REST API response to client, we can use DTO to combine and return multiple entities or multiple attributes in one shot

From server to client.

2. In Java applications - we use entity classes to represent tables in a relational database. Without DTOs, we'd have to expose the entire entities to a remote interface. This causes a strong coupling between an API and a persistence model.

3. Using DTOs on RESTful APIs written in Java (and on Spring Boot), is that they can help to hide implementation details of domain objects (JPA entities). Exposing entities through endpoints can become a security issue if we do not carefully handle what properties can be changed through what operations.

For Example: If the entity class contains sensitive information (ex: password, codes, photo etc) then if we return the entity directly then the client will get this sensitive information which causes security issues.