

what is service container

At its core, a service container is a centralized registry or repository for storing and managing instances of classes, components, or services that your application relies on. The main purposes of a service container are:

1. **Singletons and Scoped Instances:** A container can manage the lifecycle of objects, ensuring that there's only one instance of a certain class (singleton) or creating new instances with a specific scope (e.g., per HTTP request). This is especially useful for managing resources efficiently.
2. **Lazy Loading:** Components are often not instantiated immediately but are created when they're actually needed. This helps improve application startup time and resource usage.
3. **Service Resolution:** When your code needs a specific service or component, it can request it from the container. The container is responsible for finding the appropriate class and instantiating it with its dependencies.

what is validations

Validations, in the context of software development, refer to the process of checking and verifying whether the data or input provided to a program or system conforms to certain rules, requirements, or constraints. The goal of validations is to ensure the accuracy, consistency, and integrity of the data that flows through the application. They help prevent errors, security vulnerabilities, and unexpected behavior by rejecting or correcting invalid data before it can cause issues.

1. **Data Type Validation:** This involves checking whether the input data matches the expected data type. For example, ensuring that a user's age is provided as a number, not a string.
2. **Format Validation:** Verifying that data adheres to a specific format, such as ensuring that an email address is correctly formatted (e.g., user@example.com).

3. Uniqueness Validation: Checking if data is unique within a certain context, such as ensuring that usernames or email addresses are not duplicated in a database.
4. Custom Validations: Implementing custom validation rules that are specific to your application's needs.