

Day-1: Why Spring Framework Uses IoC (Inversion of Control)?

Master the foundational concept that makes Spring Framework powerful and elegant for building enterprise applications.

Understanding IoC: Control Moves to Spring

IoC means the control of object creation and management moves from your code to the Spring container. Instead of manually creating objects, Spring creates and injects them automatically.

This approach reduces boilerplate code significantly, makes testing easier, and helps beginners write cleaner, more maintainable applications without worrying about complex object creation logic.



Less Code

No manual object creation



Easy Testing

Mock dependencies simply



Cleaner Apps

Professional structure



DURGASOFT

Page 2

Traditional Java vs Spring Approach

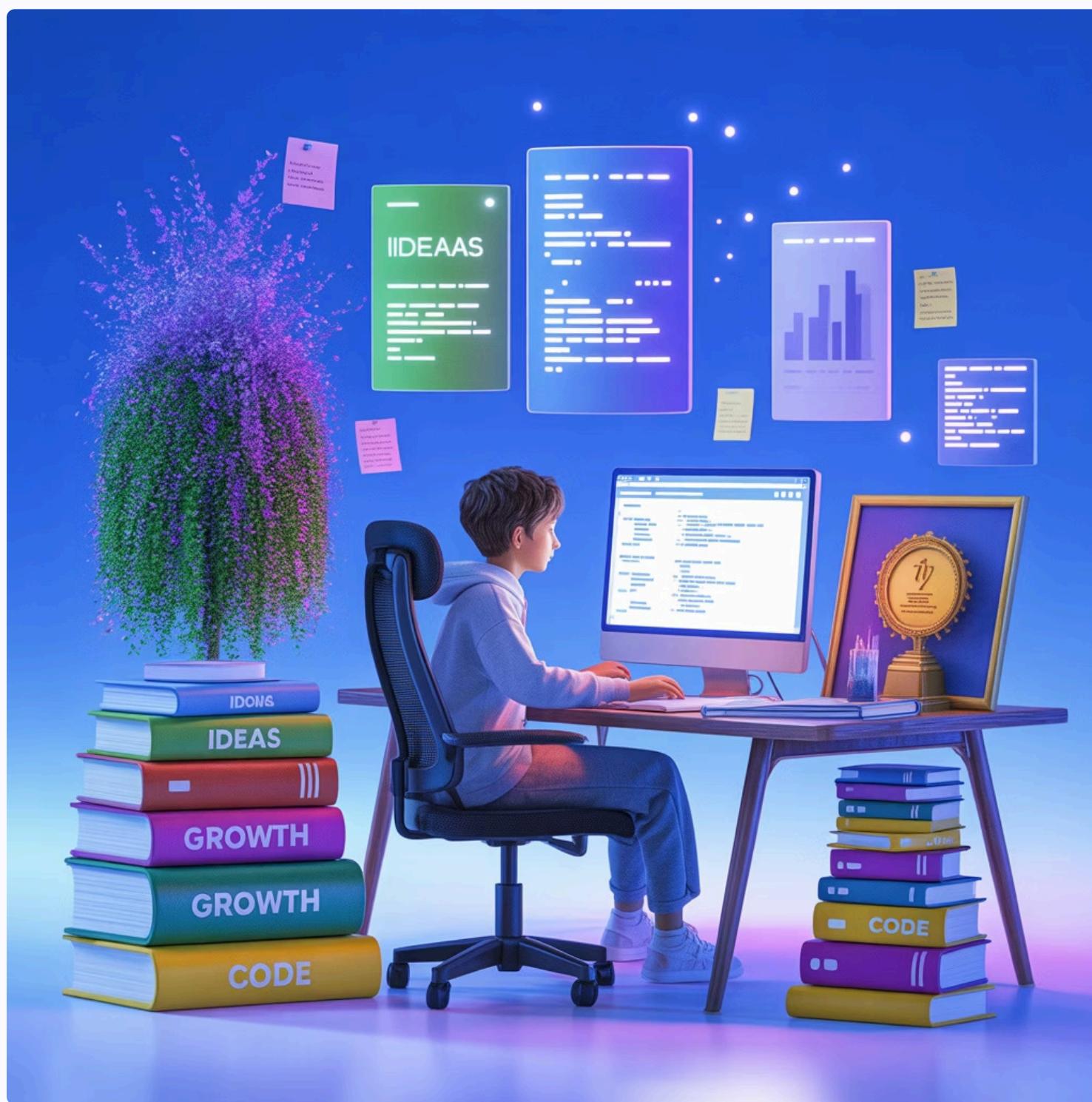
In normal Java, you write `new ClassName()` everywhere in your code. But in Spring, you simply declare a bean or add annotations like `@Component` and Spring gives you the object automatically.



This is very useful for large applications and cloud-based microservices where hundreds of objects need to work together seamlessly. Spring manages all the complexity for you.

Spring's IoC container handles object lifecycles, dependencies, and configuration—freeing you to focus on business logic.

"Every new Spring concept you learn today will save hours in your future projects."





Join DURGASOFT for Expert Spring Training

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.

Contact: 9246212143, 8885252627

Website: www.durgasoftonline.com

Email: durgasoftonlinetraining@gmail.com



DURGASOFT

Page 5

Day-2: What Are Spring Beans and Why Are They Important?

Discover how Spring Beans form the building blocks of enterprise applications.



Spring Beans: Objects Managed by Spring

A Spring Bean is simply a Java object that Spring manages for you. Spring creates it, controls its lifecycle, and injects it wherever needed in your application.



This reduces manual coding and makes your project more modular, testable, and easy to maintain –especially for cloud applications where scalability and reliability matter most.

01

Spring Creates

Automatic object instantiation

02

Spring Manages

Lifecycle control

03

Spring Injects

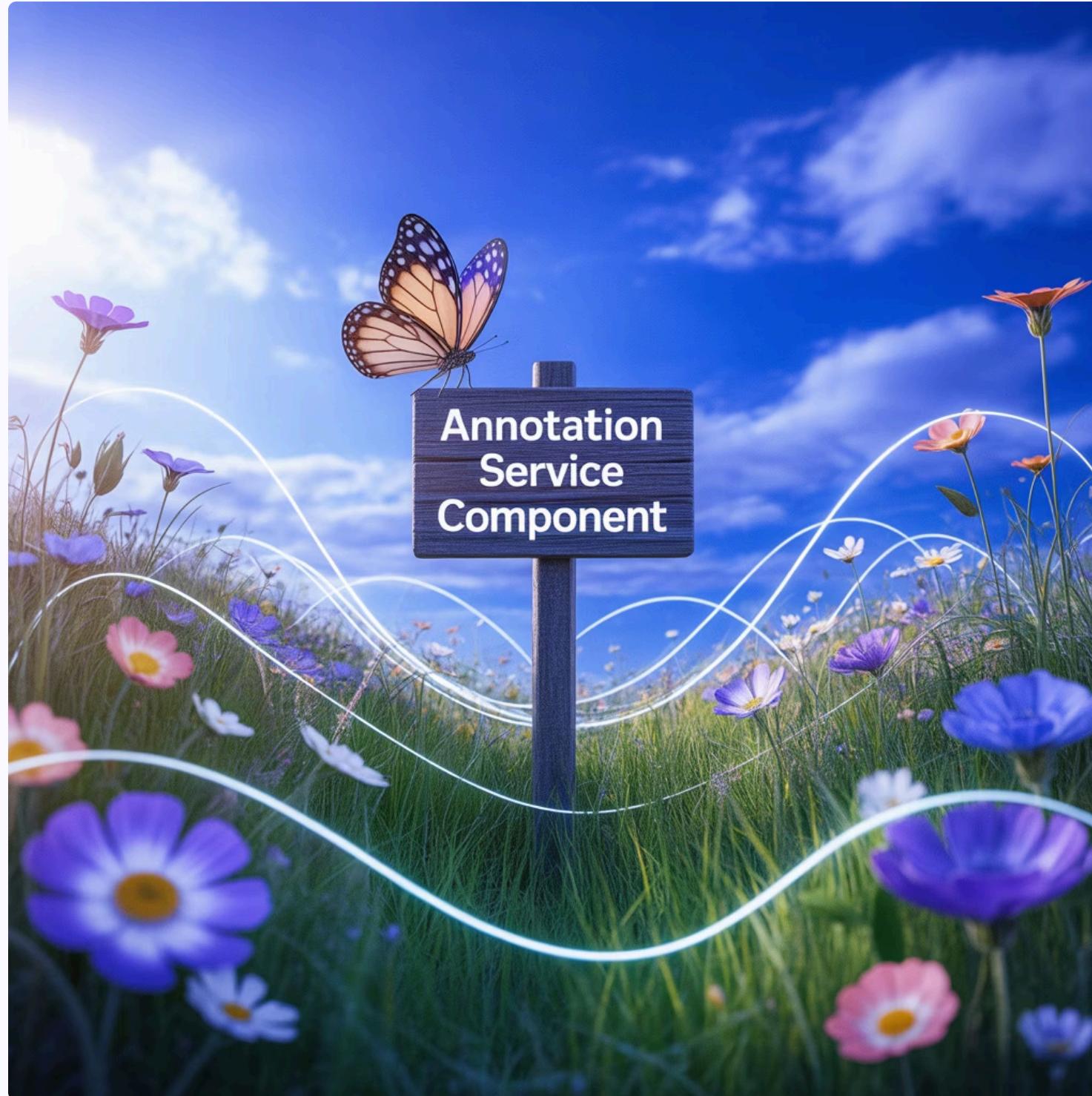
Dependency delivery

Beans in Action: Before and After Spring

Instead of writing manual object creation like this:

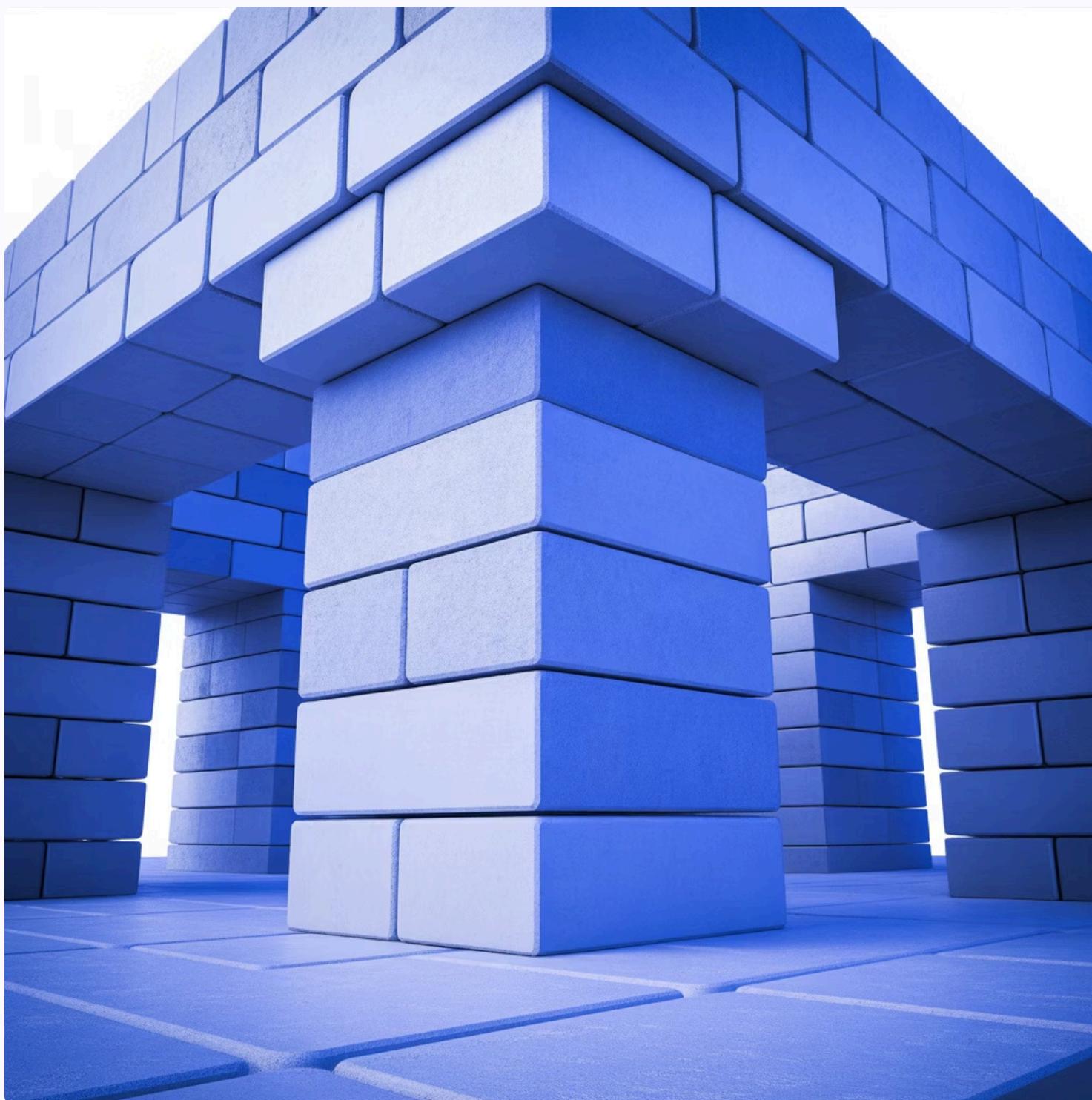
```
UserService service = new UserService();
```

You simply add `@Service` annotation and Spring creates the object automatically. In cloud projects, hundreds of such beans work together cleanly without you worrying about object creation.



Beans eliminate the need for factory patterns and manual dependency wiring in large enterprise systems.

**"Small concepts
like Beans create
big stability in
real-world
projects."**





Master Spring with DURGASOFT

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT
- India's trusted training institute.

Contact: 9246212143, 8885252627

Website: www.durgasoftonline.com

Email: durgasoftonlinetraining@gmail.com



DURGASOFT

Page 10

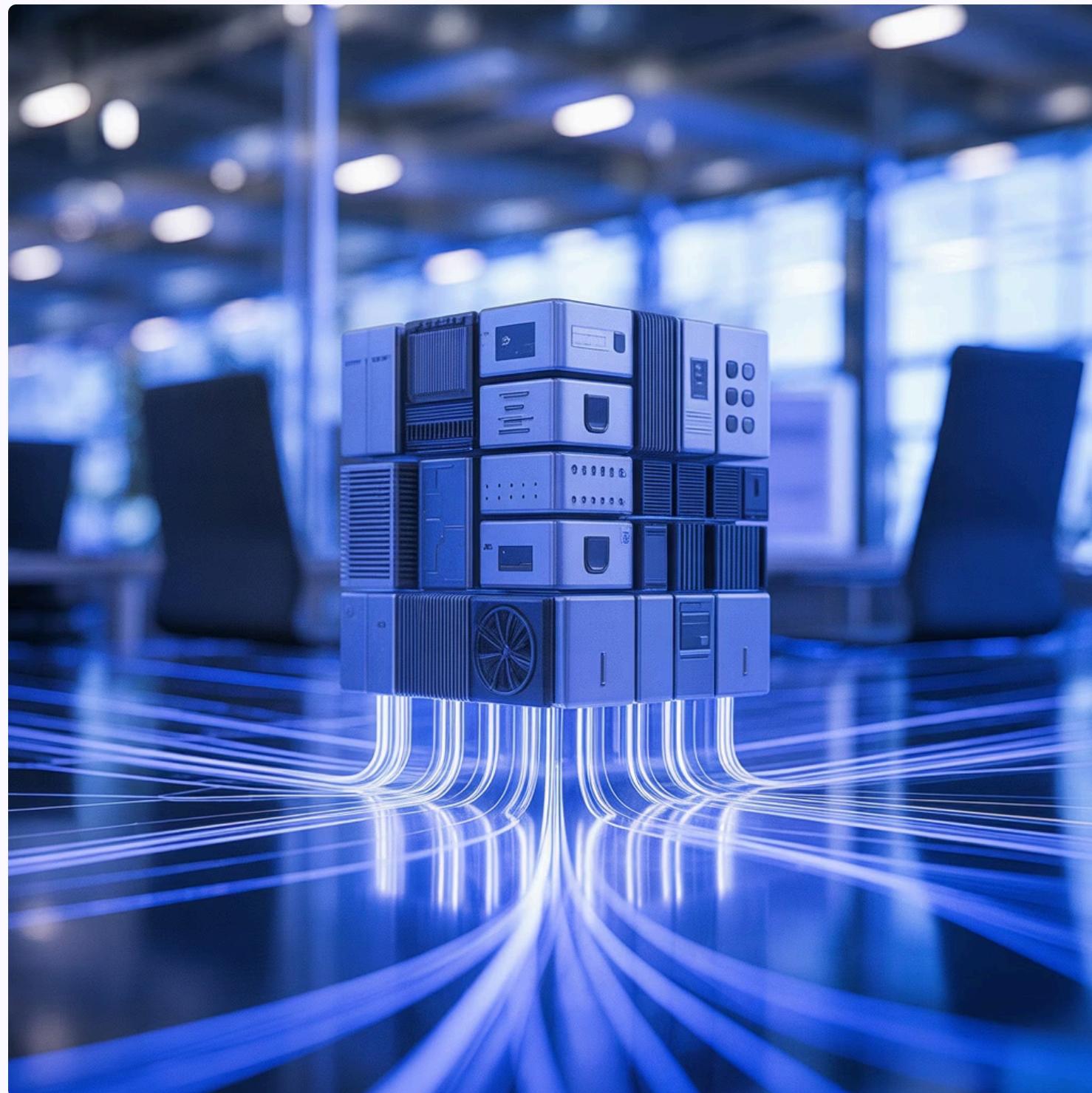
Day-3: What Is Dependency Injection in Spring?

Learn the heart of Spring Framework that makes enterprise development simple and powerful.



Dependency Injection: Spring Gives, You Receive

Dependency Injection (DI) means Spring gives the required objects (dependencies) to your class instead of you creating them manually. This makes your code clean, reduces tight coupling, and improves testing significantly.



DI is the heart of Spring and the reason companies love it for cloud projects. It allows you to change implementations without modifying existing code, making your applications flexible and maintainable.



Tight Coupling

Hard to test and change



Spring DI

Automatic injection



Loose Coupling

Easy to test and modify

DI in Practice: Manual vs Automatic

Without DI (Manual Creation):

```
OrderService service = new OrderService(new PaymentService());
```

With DI (Spring Magic):

```
@Autowired  
private PaymentService paymentService;
```

Just write `@Autowired` and Spring injects `PaymentService` automatically. This helps you build large enterprise systems without messy object creation code scattered everywhere.



In microservices architecture, DI allows hundreds of components to work together seamlessly without manual wiring.

"When you learn
Dependency
Injection, Spring
suddenly feels
simple and
powerful."





Learn Spring DI with Expert Trainers

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT
- India's trusted training institute.

 **Contact:** 9246212143, 8885252627

 **Website:** www.durgasoftonline.com

 **Email:** durgasoftonlinetraining@gmail.com



DURGASOFT

Page 15

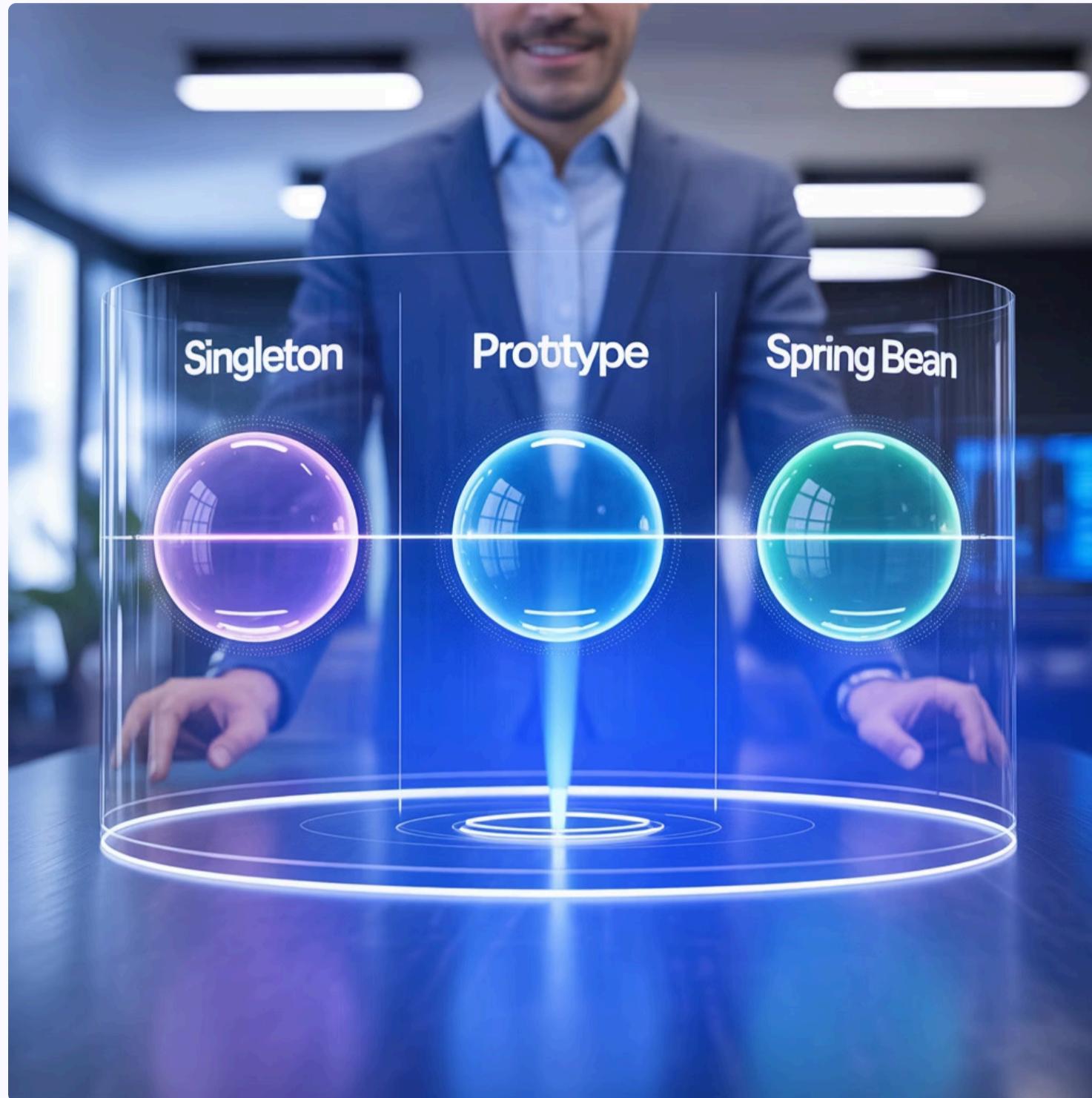


Day-4: What Are Bean Scopes in Spring and Why Do They Matter?

Control how long your beans live and how they behave in your application.

Bean Scopes: Controlling Object Lifespan

Bean Scope tells Spring *how long* a Bean should live. The most common scopes are **singleton** (one object for the entire application) and **prototype** (new object every time you request it).



Understanding scopes helps you manage memory efficiently, optimize performance, and control behaviour in enterprise and cloud applications where resource management is critical.

Singleton Scope

One shared instance for entire application

Prototype Scope

New instance created each time

Request Scope

New instance per HTTP request

Session Scope

New instance per user session

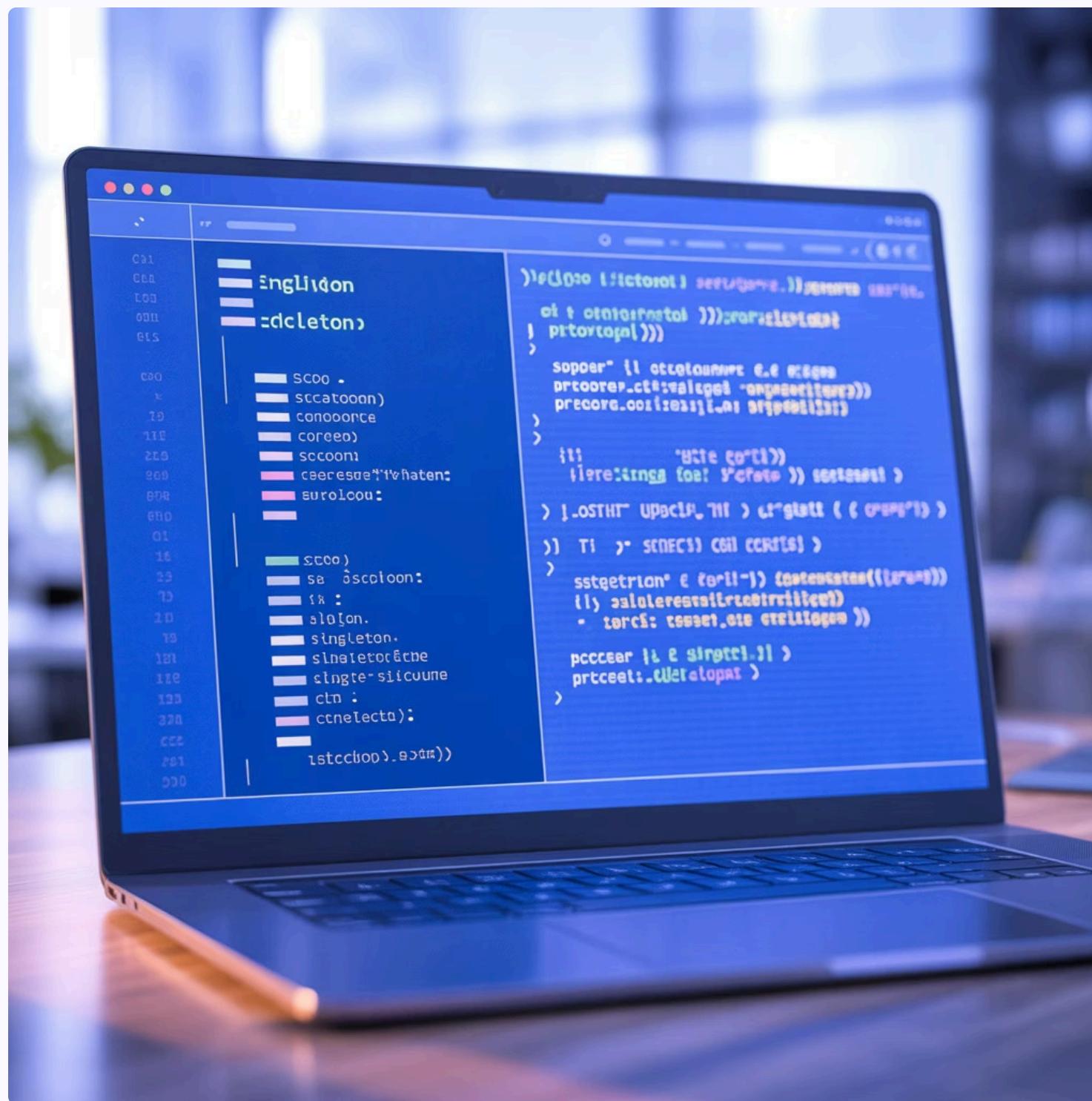
Practical Bean Scope Examples

If you mark a class with `@Scope("singleton")`, Spring creates only **one** object and shares it everywhere in the application. This is memory-efficient and the default behaviour.

```
@Component  
@Scope("singleton")  
public class DatabaseService { }
```

If you use `@Scope("prototype")`, Spring creates **a fresh object** whenever needed. This is useful for stateful operations where each user needs their own instance.

```
@Component  
@Scope("prototype")  
public class ShoppingCart { }
```



**"Knowing how
your objects
behave gives you
the power to
control your entire
application."**





Master Bean Scopes at DURGASOFT

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.

Contact: 9246212143, 8885252627

Website: www.durgasoftonline.com

Email: durgasoftonlinetraining@gmail.com



DURGASOFT

Page 20



Day-5: What Is Autowiring in Spring?

Let Spring automatically connect your objects—no manual wiring needed!

Autowiring: Spring Connects Objects Automatically

Autowiring means Spring automatically finds the correct Bean and injects it into your class. You don't need to write long configuration code to connect objects together.



Just use annotations like `@Autowired` or `@Qualifier`, and Spring does the wiring for you—making development fast, clean, and efficient. This is essential for rapid application development.



Spring Scans

Finds matching beans



Spring Matches

Identifies correct dependency



Spring Injects

Automatically wires objects

Autowiring in One Simple Line

@Autowired

```
private PaymentService paymentService;
```

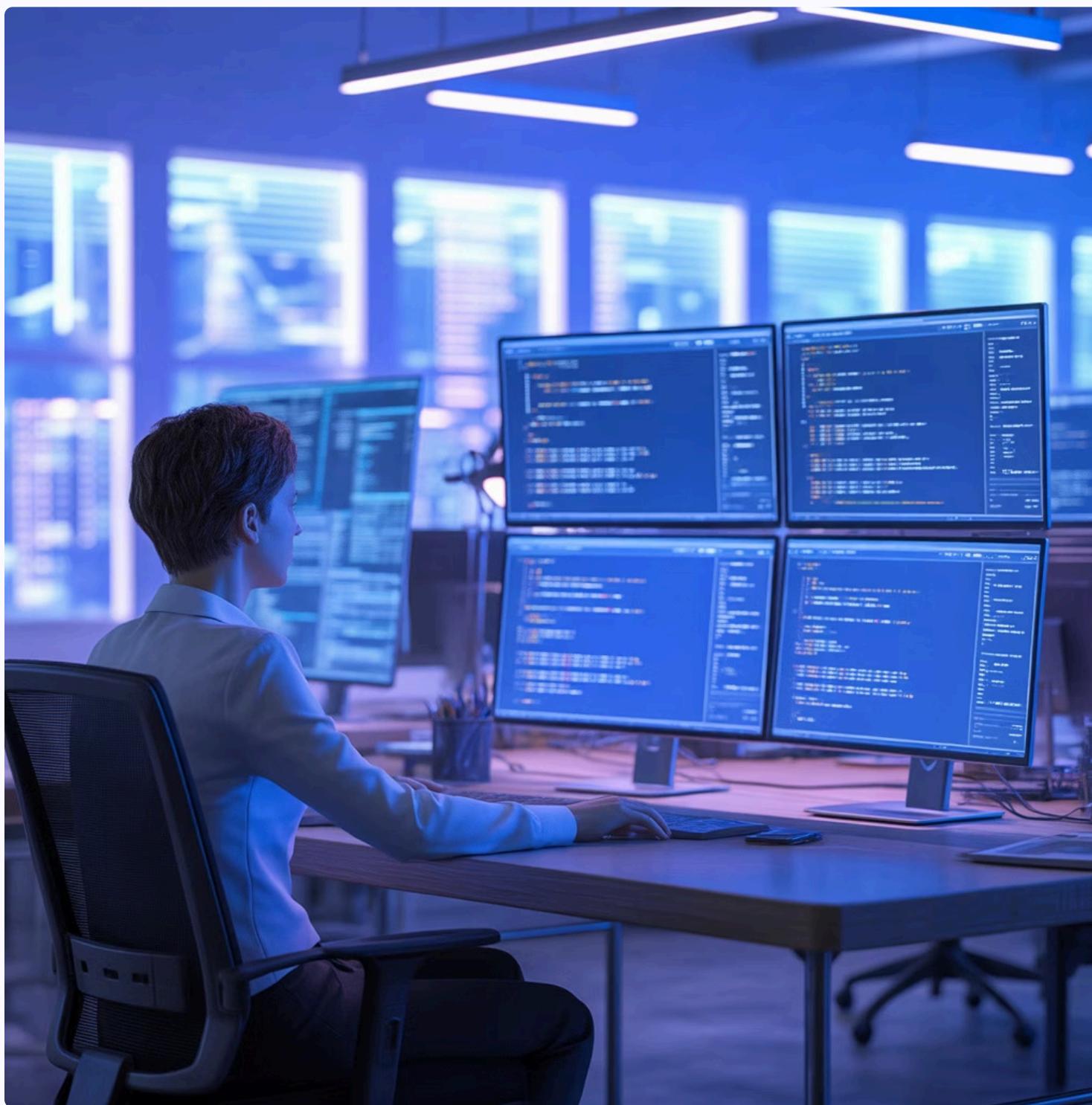
With this single line of code, Spring gives you the ready-made PaymentService object. No manual creation, no factory classes, no complex configuration files needed.



In cloud microservices, autowiring helps connect many components without manual setup. This is how modern applications handle hundreds of interconnected services efficiently.

Autowiring reduces configuration time by up to 70% compared to manual XML-based wiring in traditional applications.

**"When Spring
wires your objects,
you can focus on
writing powerful
business logic."**





Accelerate Your Learning with DURGASOFT

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.

Contact: 9246212143, 8885252627

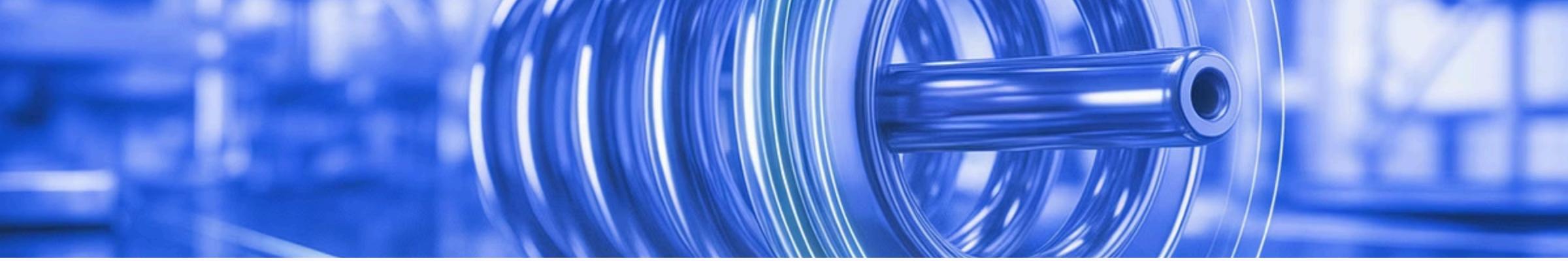
Website: www.durgasoftonline.com

Email: durgasoftonlinetraining@gmail.com



DURGASOFT

Page 25

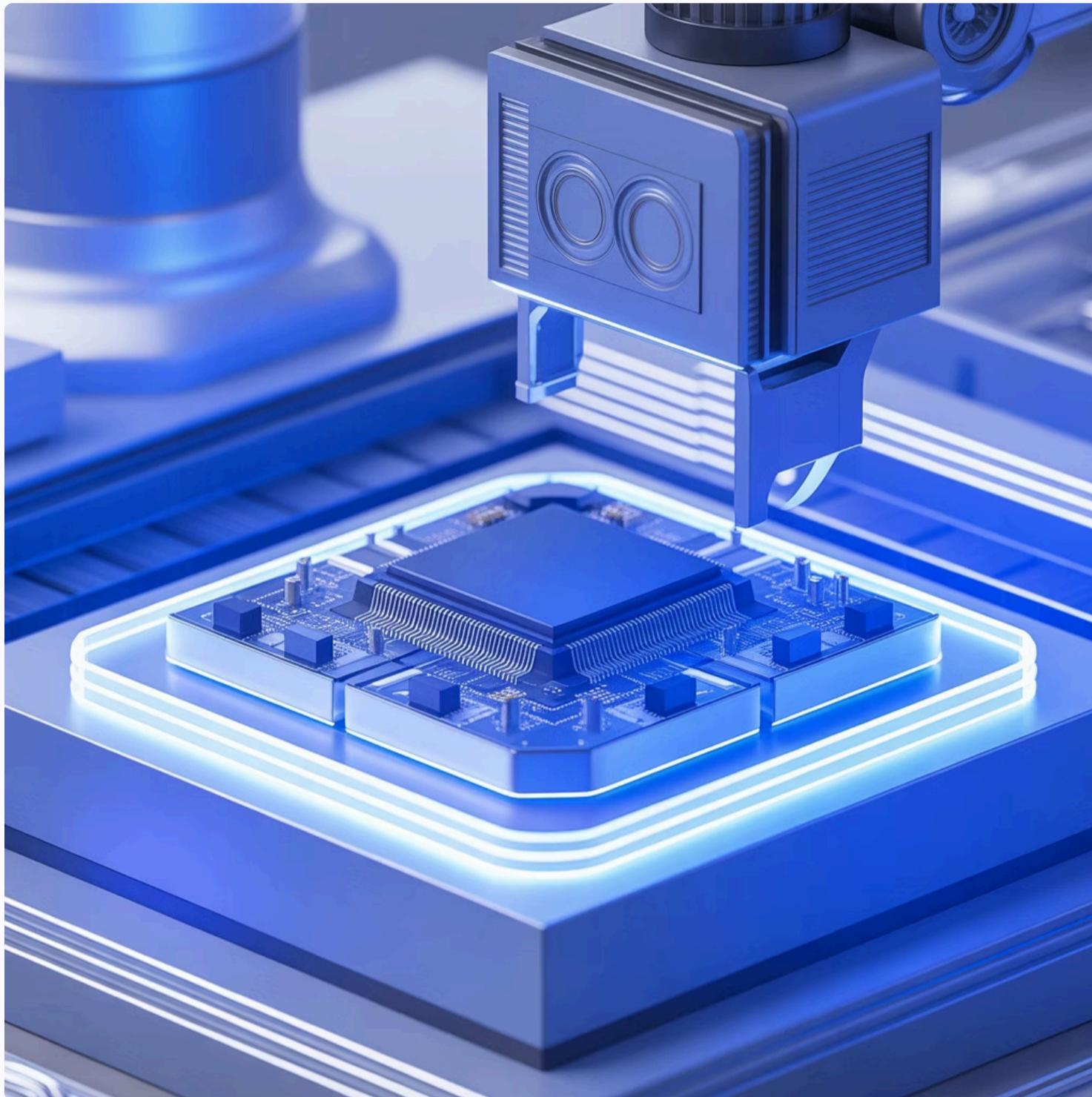


Day-6: What Is @Component in Spring?

The foundation of annotation-based Spring development starts here.

@Component: Automatic Bean Detection

@Component tells Spring to automatically detect the class and create a Bean for it during component scanning. Instead of writing XML configuration or manual Java config, you just mark the class with this annotation.



This is the foundation for annotation-based Spring development and is widely used in modern cloud-ready applications. It simplifies configuration and makes your code more readable and maintainable.



@Component Makes Bean Creation Effortless

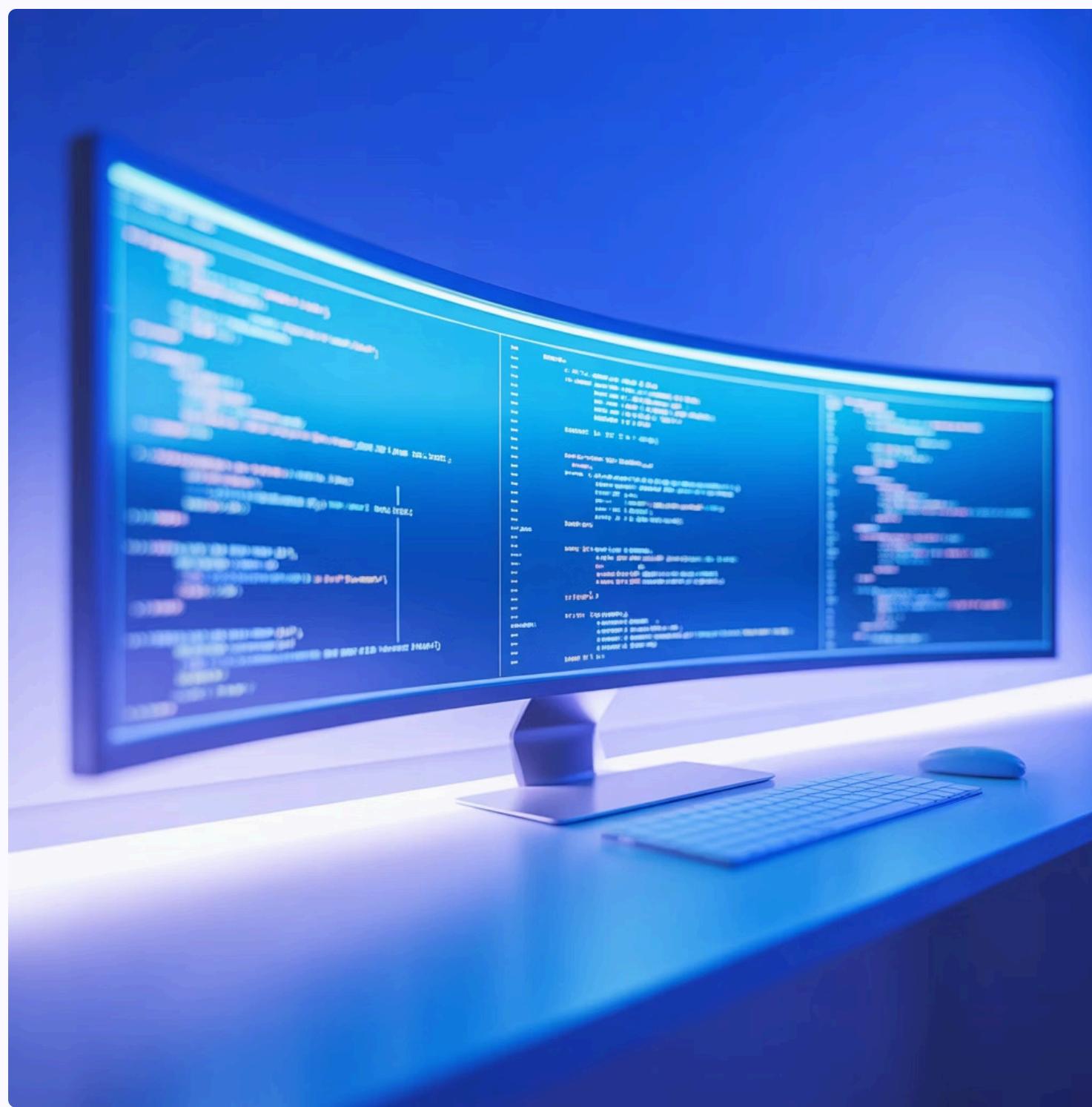
```
@Component  
public class EmailService {  
    public void sendEmail(String to, String message) {  
        // Email sending logic  
    }  
}
```

Spring will scan the package, find this class automatically, and create one EmailService Bean for you. No XML files, no complex configuration classes needed.



This reduces configuration time dramatically and keeps the project clean and easy to expand. As your application grows, adding new components becomes as simple as adding an annotation.

**"One small
annotation can
remove dozens of
lines of
configuration.
That's the beauty
of Spring."**





Become a Spring Expert with DURGASOFT

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT
- India's trusted training institute.

Contact: 9246212143, 8885252627

Website: www.durgasoftonline.com

Email: durgasoftonlinetraining@gmail.com

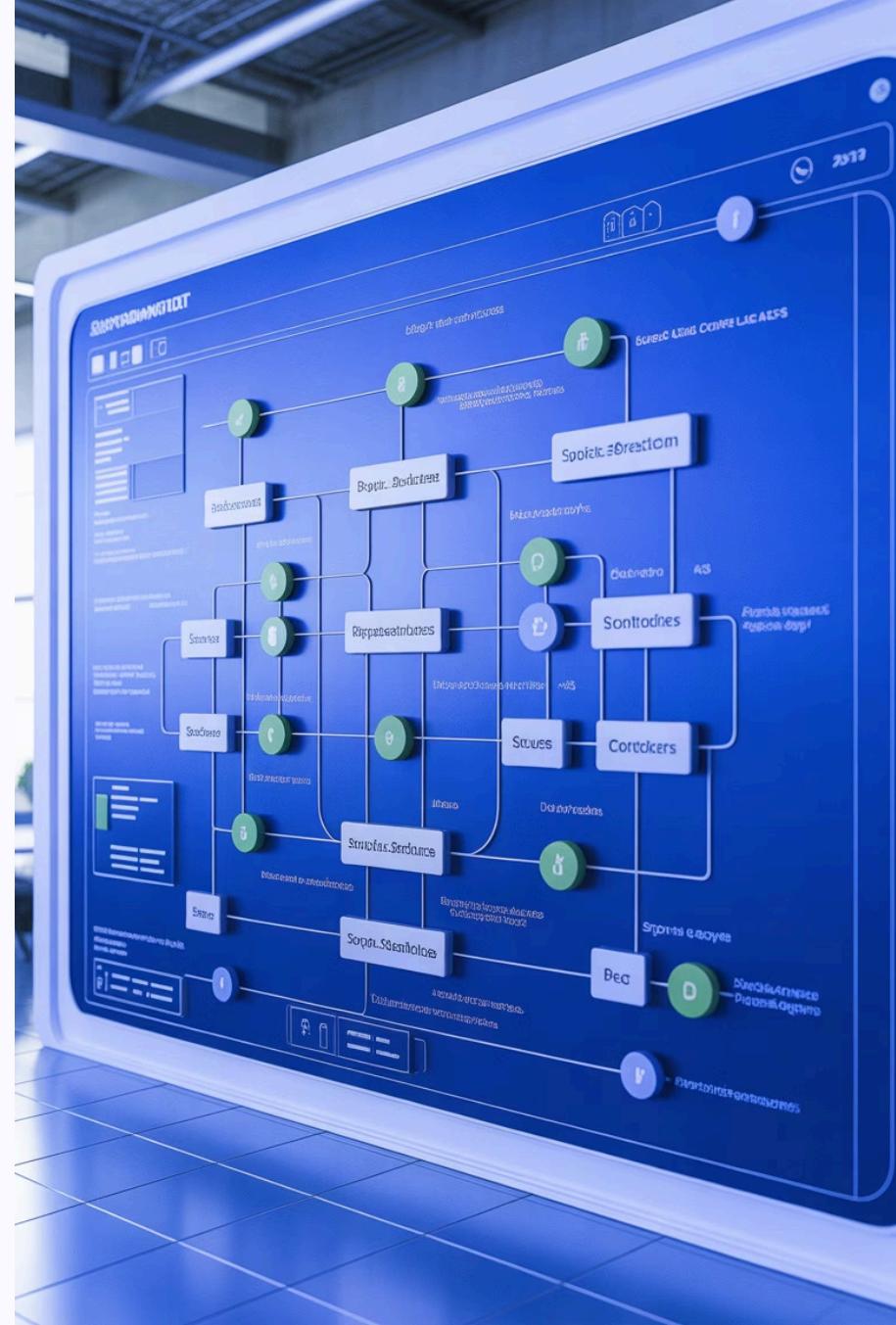


DURGASOFT

Page 30

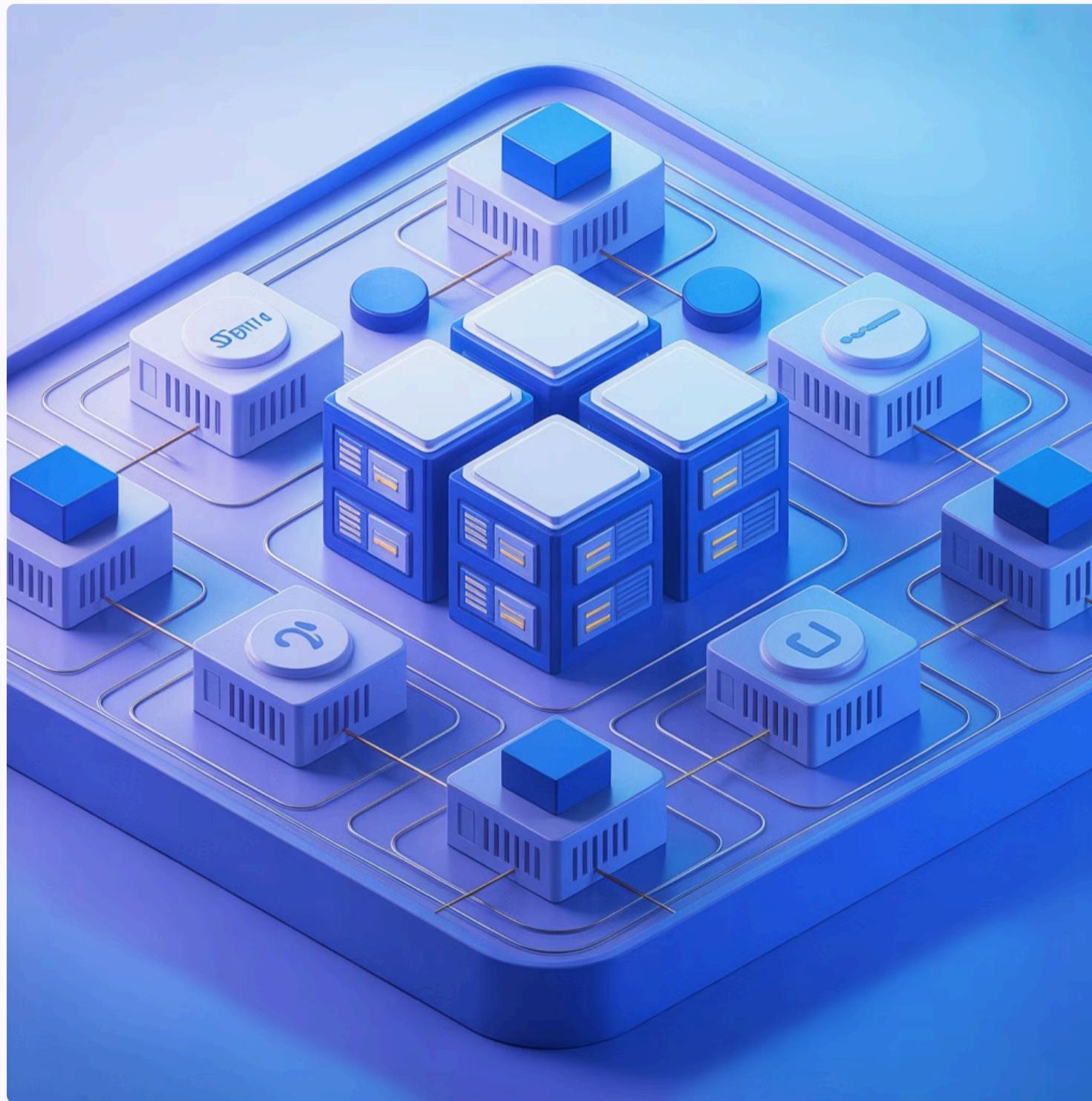
Day-7: What Is @Service in Spring and When Should You Use It?

Learn how to organize your business logic professionally using Spring stereotypes.

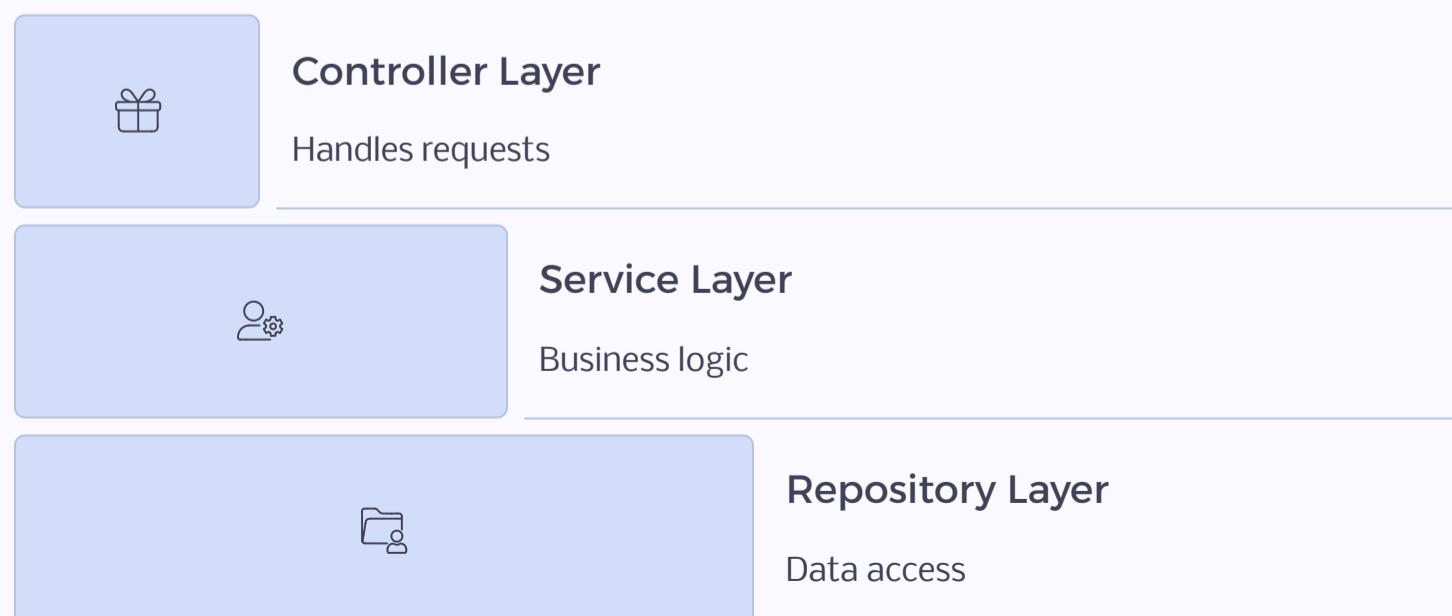


@Service: Specialized Bean for Business Logic

@Service is a specialization of @Component used specifically for classes that hold business logic. It helps developers clearly separate responsibilities in an application following proper layered architecture.



Spring also provides additional features like transaction management when using @Service, making it ideal for backend services and cloud applications where data consistency is critical.



@Service in Real Applications

```
@Service
public class PaymentService {

    @Autowired
    private PaymentRepository paymentRepository;

    public void processPayment(Order order) {
        // Business logic for payment processing
        // Validation, calculation, transaction handling
    }
}
```

Spring treats PaymentService as a service-layer Bean, making your project well-structured and easy to maintain. This separation becomes crucial when multiple teams work on the same cloud application.



The @Service annotation also enables Spring's transaction management, ensuring data consistency across database operations.

**"Clean structure
creates clean
growth – both for
applications and
your career."**





Build Professional Skills at DURGASOFT

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.

Contact: 9246212143, 8885252627

Website: www.durgasoftonline.com

Email: durgasoftonlinetraining@gmail.com

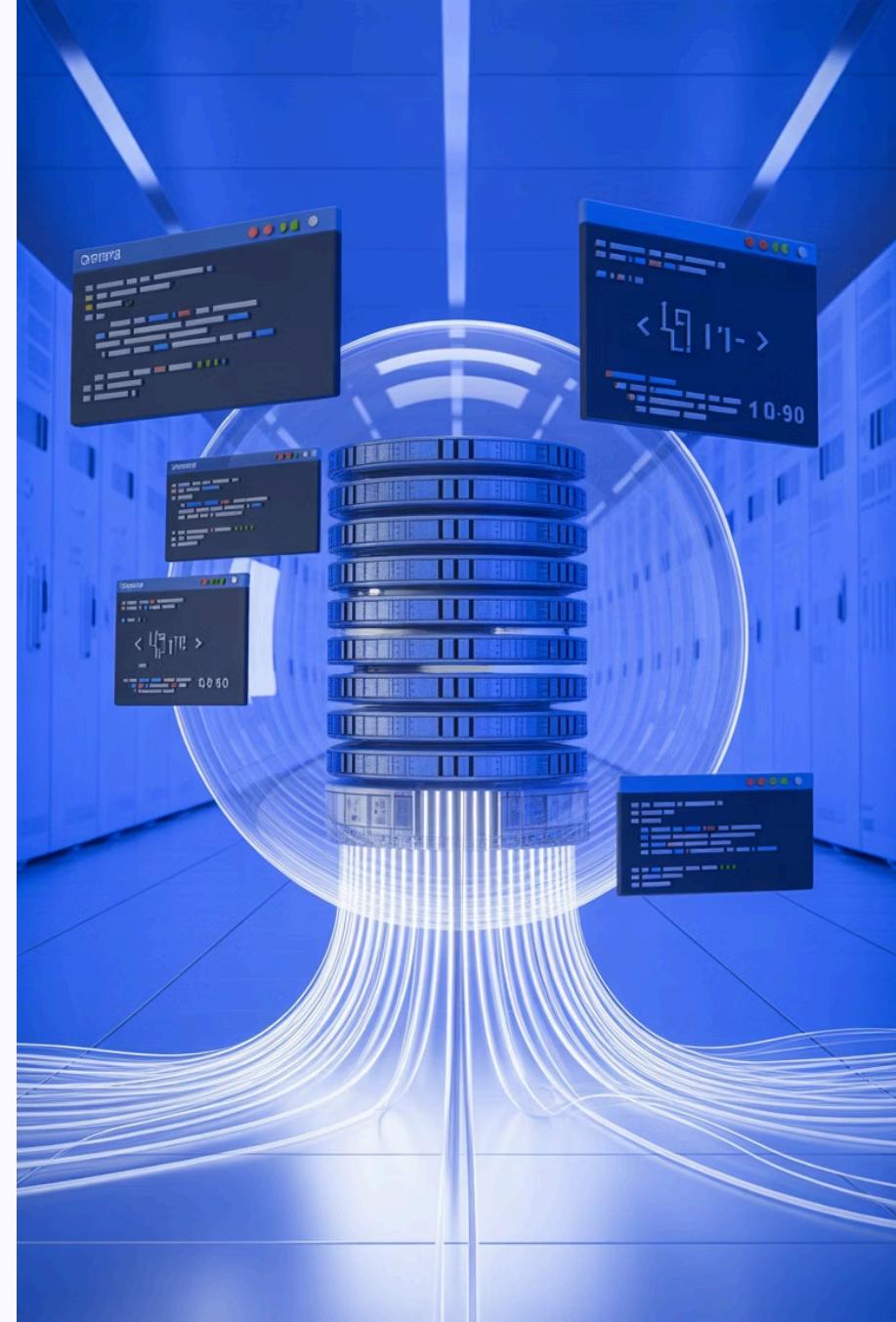


DURGASOFT

Page 35

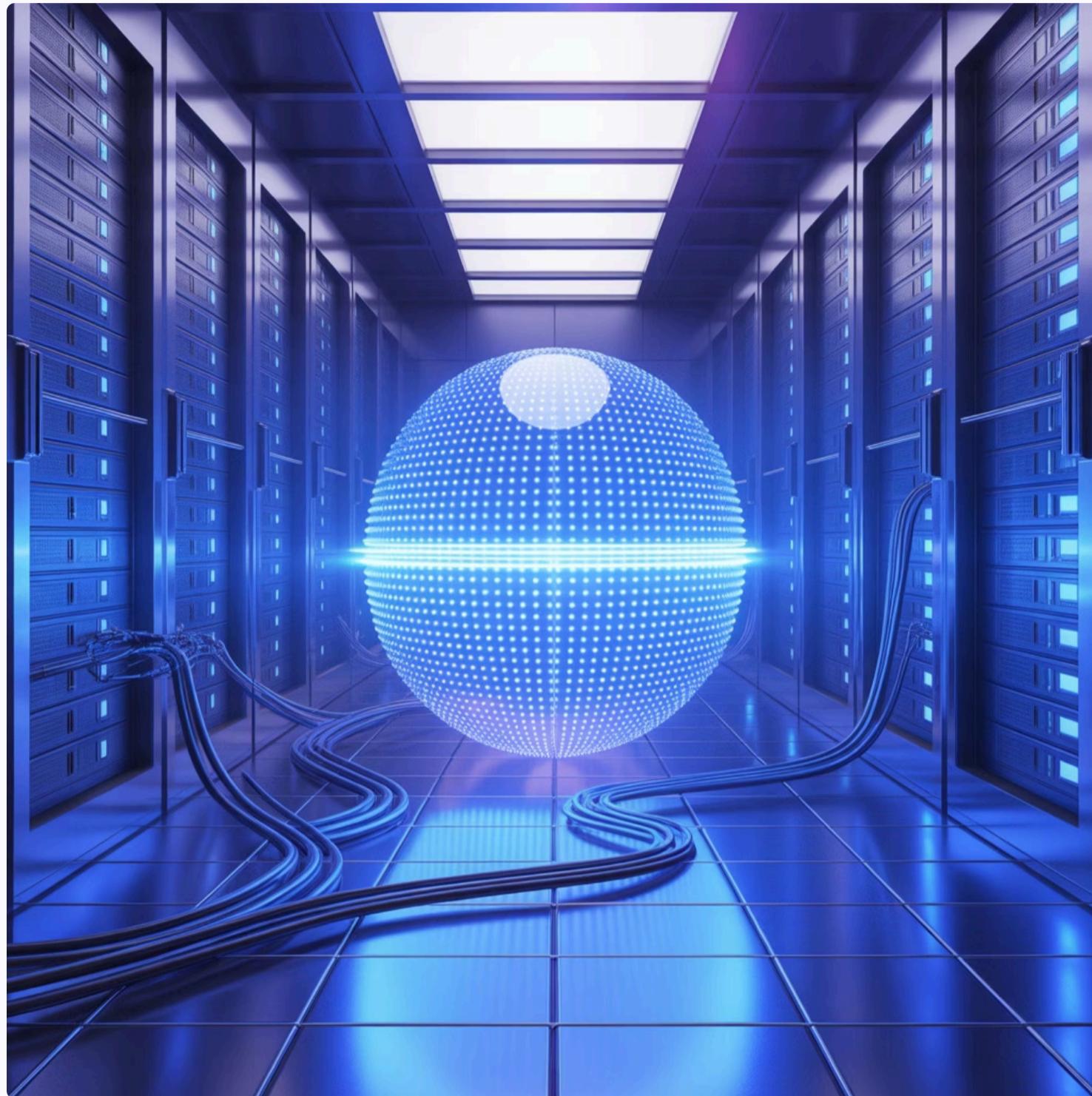
Day-8: What Is @Repository in Spring and Why Do We Use It?

Master the data access layer with Spring's specialized repository annotation.



@Repository: For Database Operations

@Repository is used for classes that talk directly to the database. It's a specialized component that marks the Data Access Object (DAO) layer of your application.



Spring automatically translates database exceptions into Spring-friendly exceptions, making error handling simple and consistent. This helps keep all database operations clean, separate, and easy to maintain in large cloud-ready applications.

Exception Translation

Converts database-specific exceptions to Spring's `DataAccessException`

Clear Separation

Isolates data access logic from business logic

Easy Testing

Mock repositories for unit testing without database

@Repository in Database Applications

```
@Repository
public class UserDAO {

    @Autowired
    private JdbcTemplate jdbcTemplate;

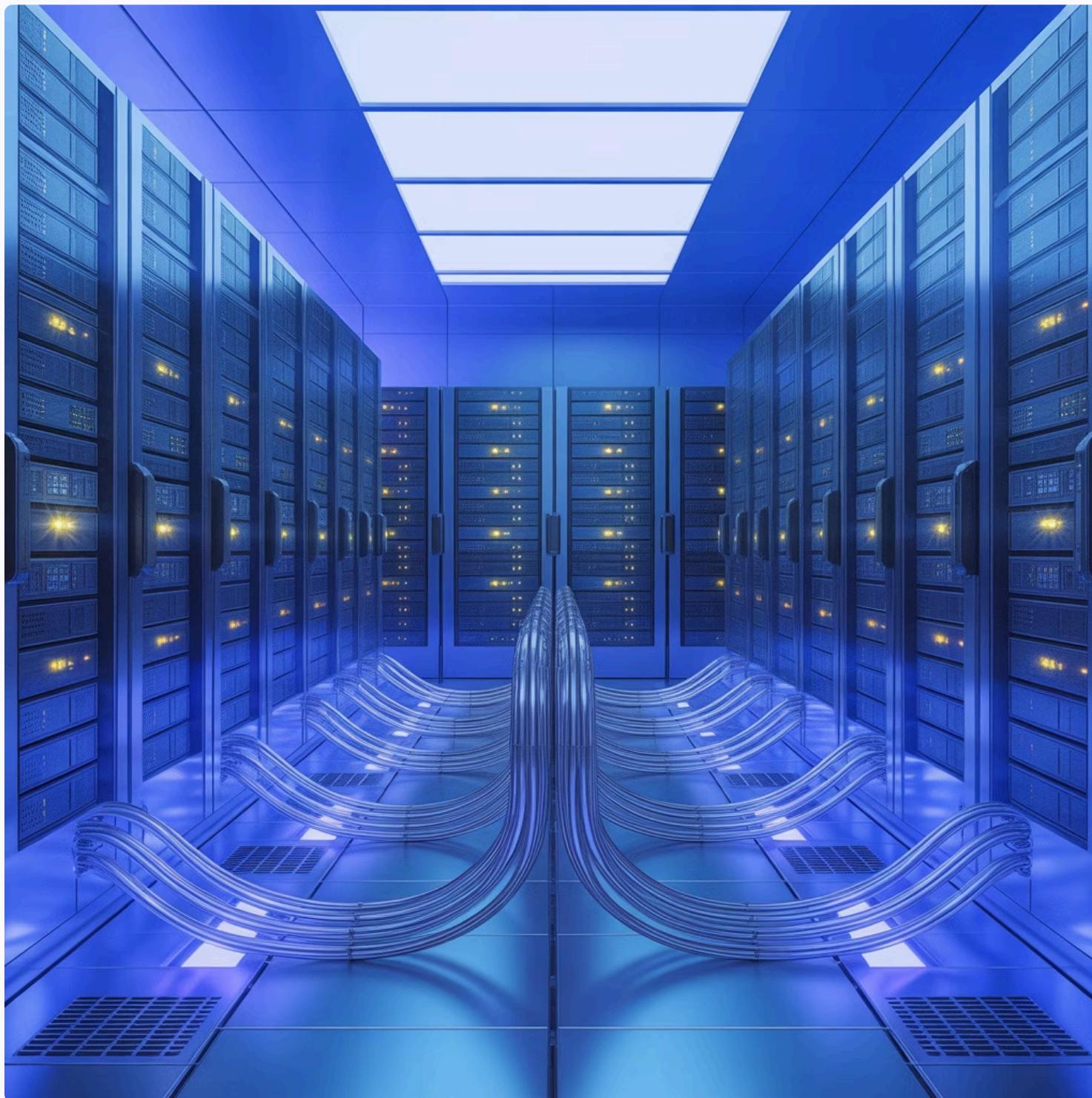
    public void saveUser(User user) {
        String sql = "INSERT INTO users VALUES (?, ?)";
        jdbcTemplate.update(sql, user.getId(), user.getName());
    }

    public User findUserById(Long id) {
        String sql = "SELECT * FROM users WHERE id = ?";
        return jdbcTemplate.queryForObject(sql, new UserMapper(), id);
    }
}
```

Spring treats UserDAO as a Data Access Object. This makes the data layer organized and provides better exception management—essential for real-world enterprise and cloud systems.



"Once your data layer is clean, the whole application becomes rock solid."





Master Data Access with DURGASOFT

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.

Contact: 9246212143, 8885252627

Website: www.durgasoftonline.com

Email: durgasoftonlinetraining@gmail.com

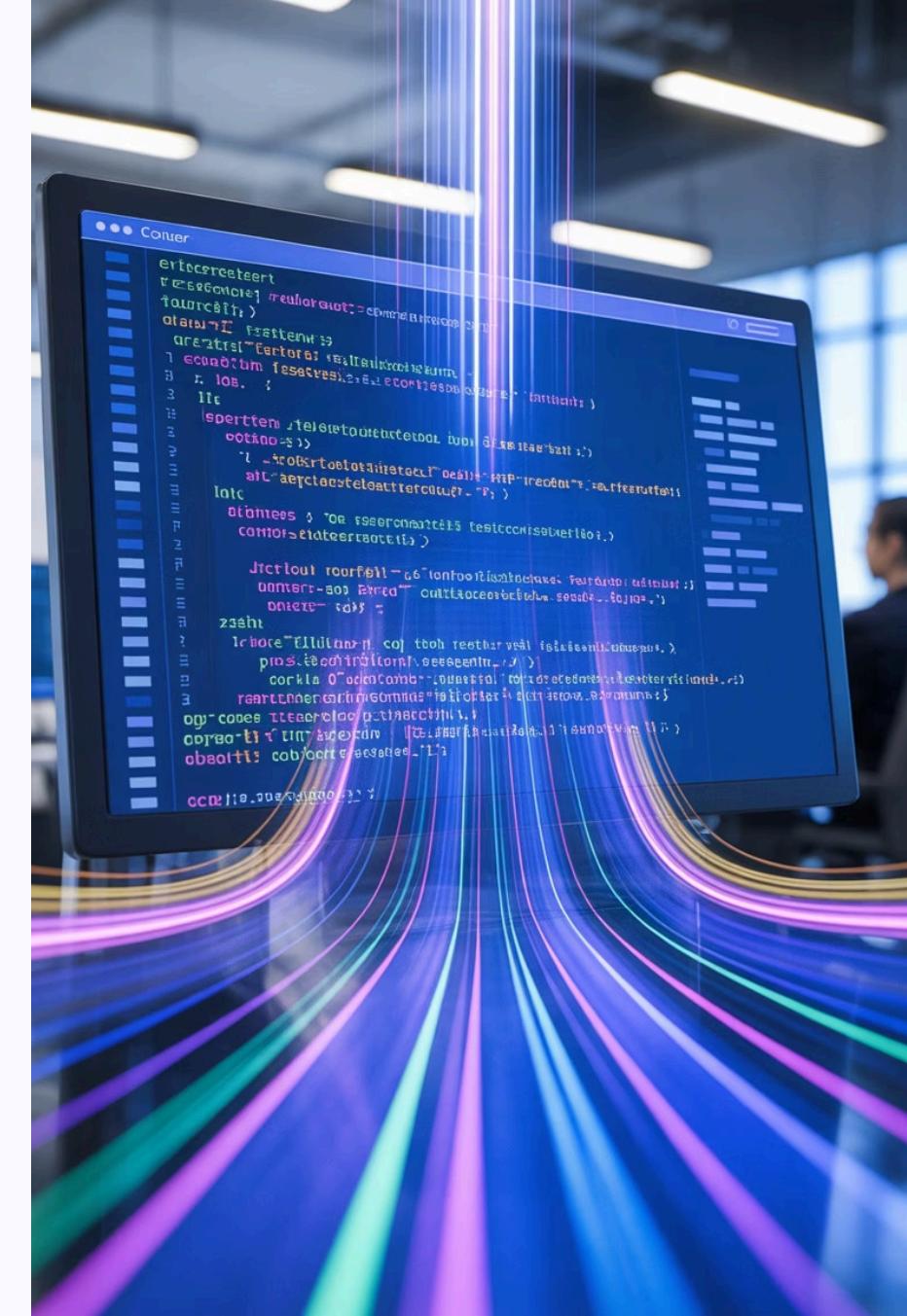


DURGASOFT

Page 40

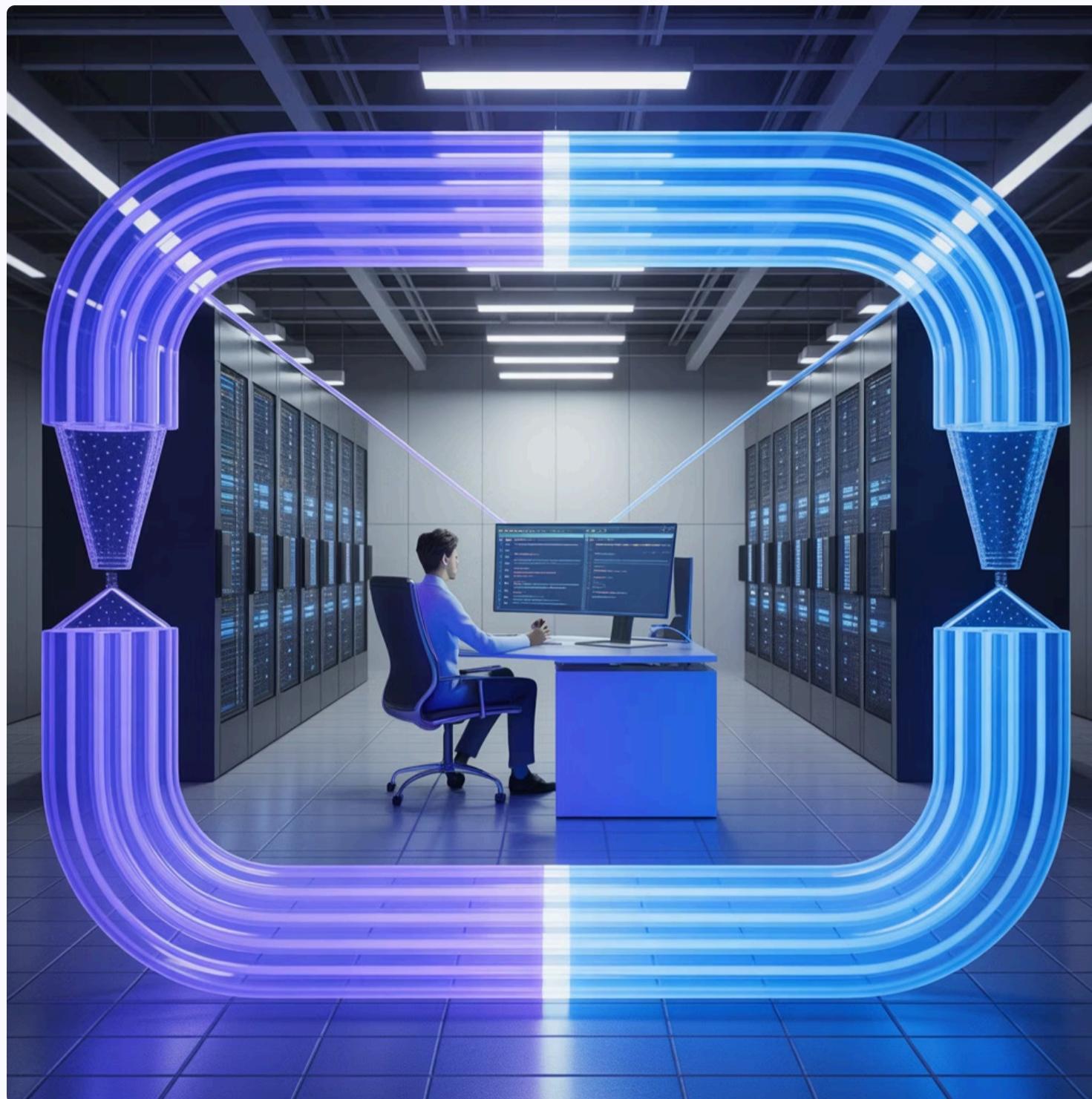
Day-9: What Is @Controller in Spring MVC?

Start building web applications with Spring MVC's powerful controller layer.

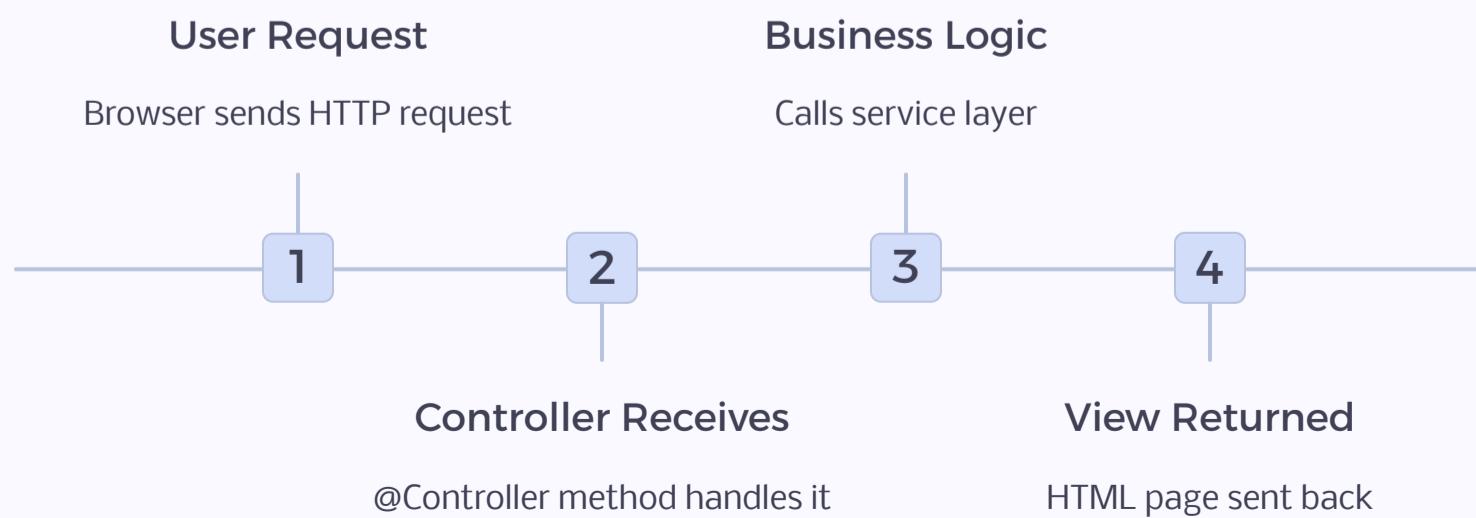


@Controller: Handling Web Requests

@Controller is used to handle web requests in Spring MVC applications. When a user opens a URL in their browser, the controller method receives the request, processes it, and returns the appropriate response or view.



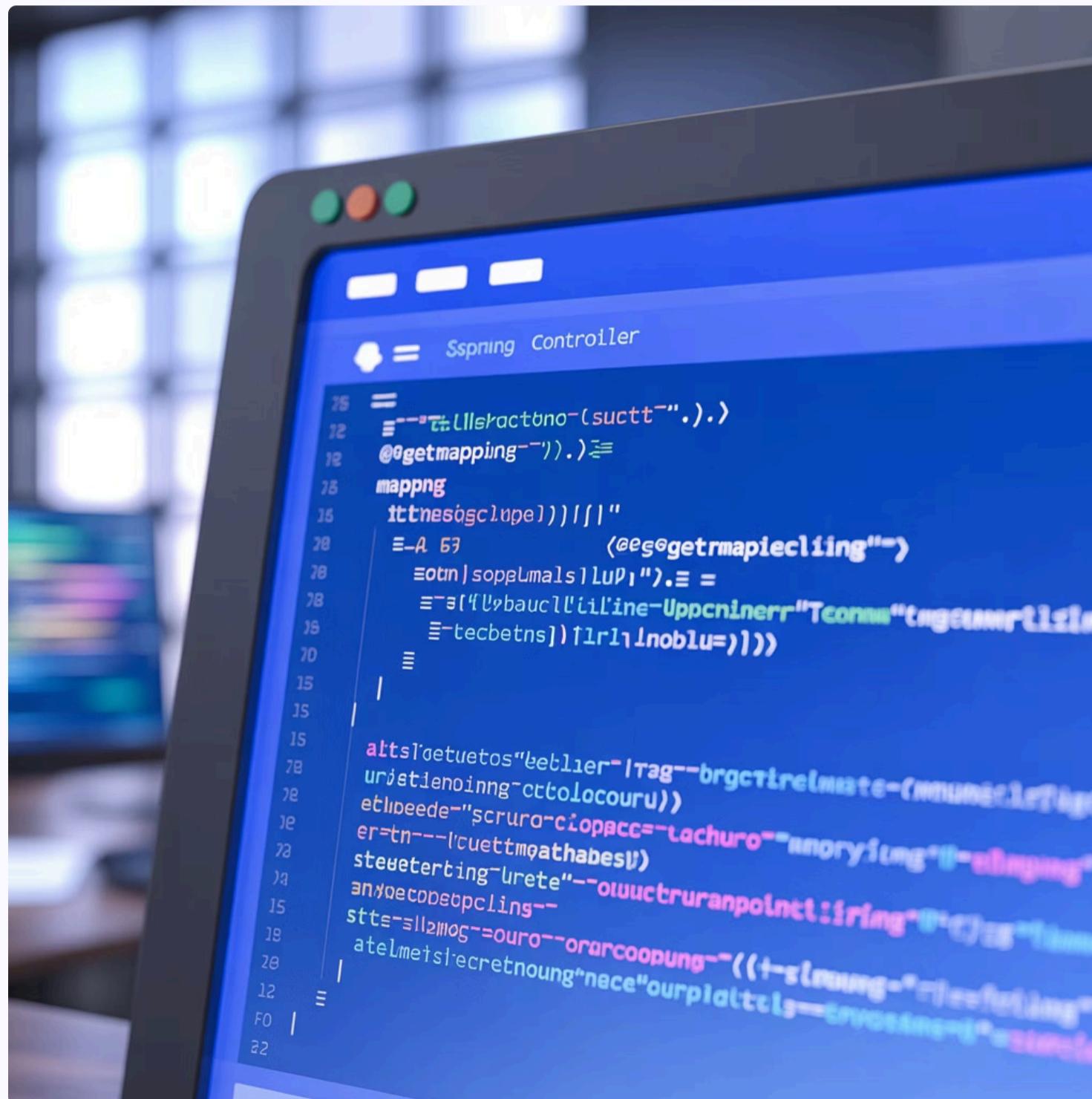
This is the starting point for building web applications using Spring Web MVC and Spring Boot. Controllers act as the bridge between the user interface and your business logic layer.



@Controller Maps URLs to Java Methods

```
@Controller  
public class HomeController {  
  
    @GetMapping("/home")  
    public String showHome(Model model) {  
        model.addAttribute("message", "Welcome to DURGASOFT!");  
        return "home";  
    }  
  
    @GetMapping("/about")  
    public String showAbout() {  
        return "about";  
    }  
}
```

Whenever someone accesses `/home`, the `showHome()` method gets executed. This is how Spring maps URLs to your Java methods—clean, simple, and powerful.



The Model object allows you to pass data from the controller to the view, creating dynamic web pages easily.

"Every great web application begins with one clean Controller."





Build Web Apps with DURGASOFT Training

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.

Contact: 9246212143, 8885252627

Website: www.durgasoftonline.com

Email: durgasoftonlinetraining@gmail.com



DURGASOFT

Page 45



Day-10: What Is @RestController in Spring Boot?

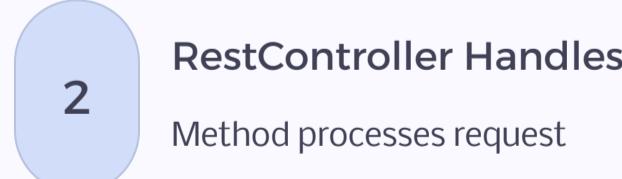
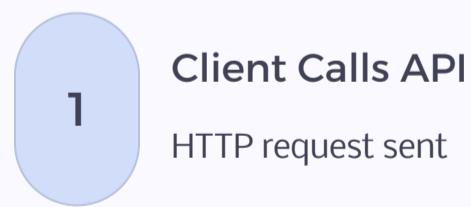
Master REST API development—the backbone of modern microservices architecture.

@RestController: Building REST APIs Made Easy

@RestController is used to build REST APIs in Spring Boot. It automatically converts Java objects into JSON responses using the built-in @ResponseBody functionality.



This is the most common annotation in modern cloud-based applications, microservices, and mobile backend systems. It eliminates the need for manual JSON conversion, making API development incredibly fast.



@RestController Returns JSON Automatically

```
@RestController
public class UserController {

    @GetMapping("/user")
    public User getUser() {
        return new User("Durga", "Hyderabad");
    }

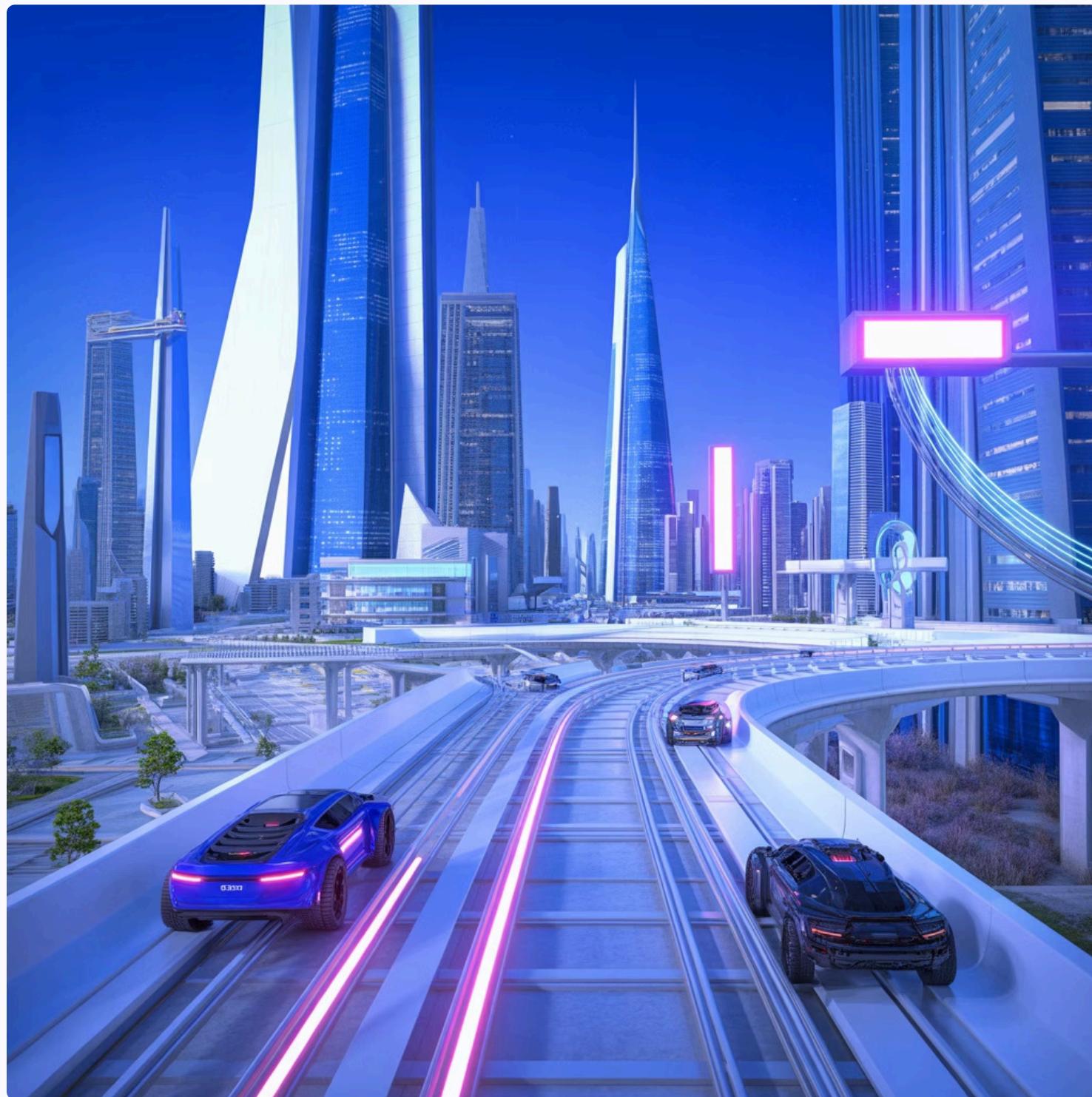
    @PostMapping("/user")
    public User createUser(@RequestBody User user) {
        // Save user to database
        return user;
    }
}
```

When you call `/user`, you get JSON output directly—no extra coding required. Spring Boot automatically serializes the User object to JSON format.



This is why Spring Boot is the top choice for REST API development in modern cloud and microservices architectures.

**"Learning REST
makes you ready
for today's
microservices
world."**





Complete Your Spring Journey with DURGASOFT

If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT

- India's trusted training institute.

Contact: 9246212143, 8885252627

Website: www.durgasoftonline.com

Email: durgasoftonlinetraining@gmail.com

Congratulations on completing this 10-day journey through Spring Framework fundamentals!

Keep practicing these concepts and building real projects. Your Spring expertise starts today!



DURGASOFT

Page 50