

Spring Boot Dependency Injection - Summary Notes

Exercise 1: Basic Constructor Injection

Goal: Inject one class into another using constructor.

Learnings:

- Use `@Component` to mark beans.
- Constructor injection improves testability and immutability.

Q&A:

- Yes, constructor can have access modifiers.
- 'this' refers to current instance.

Exercise 2: Interface Injection

Goal: Inject interface with multiple implementations.

Learnings:

- Interface allows flexibility.
- Spring injects correct implementation (manually or via `@Qualifier`).

Q&A:

- Can't instantiate interfaces directly.
- Use concrete classes for injection.

Exercise 3: Multiple Interface Injection (CakeBaker)

Goal: Inject multiple interfaces.

Learnings:

- Use constructor injection for each.
- All injected types must be Spring-managed beans.

Exercise 4: Setter Injection

Goal: Inject via setter methods.

Learnings:

- `@Autowired` must be on setter.
- Spring calls it automatically if needed.

Q&A:

- No manual setter call needed.

Exercise 5: Manual Bean Configuration with `@Bean`

Goal: Use `@Configuration` and `@Bean` to wire manually.

Learnings:

- Offers manual control.
- Can avoid `@Component` in classes.

Exercise 6: Strategy Injection via Config Class

Goal: Choose implementation externally.

Learnings:

- Keeps logic clean.
- Implementation choice stays in config.

Spring Boot Dependency Injection - Summary Notes

Q&A:

- Similar to Ex 5 but focus is on strategy switch.

Exercise 7: Nested Dependency Injection

Goal: Demonstrate dependency chains (FuelPump -> Engine -> Car).

Learnings:

- Spring resolves all levels recursively.

Q&A:

- Constructor order doesn't matter.
- Access modifiers allowed.

Exercise 8: Optional Dependency Injection

Goal: Inject dependency only if it exists.

Learnings:

- Use `@Autowired(required=false)` or `Optional<T>`.

Q&A:

- May work without `@Autowired` due to reflection, but not guaranteed.

Exercise 9: Singleton Scope - Shared Bean

Goal: Inject same bean into multiple classes.

Learnings:

- Singleton scope means shared instance.

Q&A:

- Changed to prototype shows different instances.

Exercise 10: Interface Strategy via Config

Goal: Inject selected implementation without qualifiers.

Learnings:

- Swap strategy in config file.

Q&A:

- Order of `@Bean` methods does not matter.