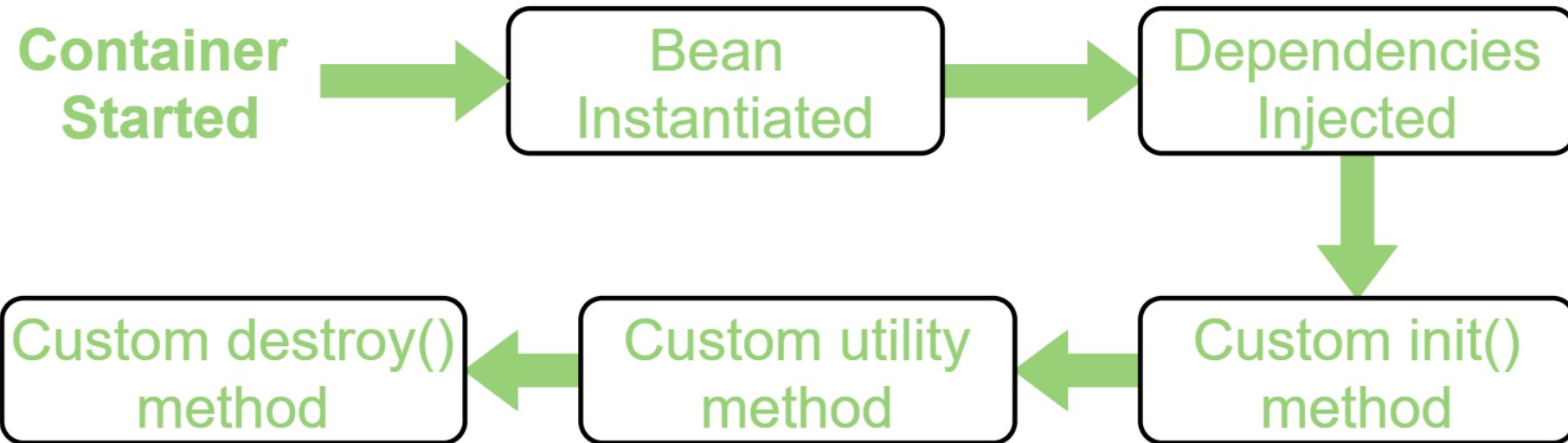




Bean Life Cycle

Vòng đời của Bean




<https://www.geeksforgeeks.org/bean-life-cycle-in-java-spring/>



@PostConstruct

@PostConstruct được đặt trước một phương thức trong một bean để chỉ định rằng phương thức đó sẽ được gọi tự động sau khi bean được khởi tạo và trước khi nó được sử dụng. Phương thức được chú thích bằng @PostConstruct phải không có tham số và không trả về giá trị.



```
import javax.annotation.PostConstruct;
import org.springframework.stereotype.Component;

@Component
public class MyBean {


    @PostConstruct
    public void init() {
        // Các công việc khởi tạo sau khi bean được tạo
    }

    // Các phương thức và logic khác của bean
}
```



@PreDestroy

Được sử dụng để xác định phương thức sẽ được thực thi trước khi bean bị hủy bỏ. Annotation này thường được sử dụng để thực hiện các tác vụ dọn dẹp, giải phóng tài nguyên hoặc lưu trữ dữ liệu trước khi bean bị hủy. Tương tự như @PostConstruct, một bean có thể có nhiều phương thức được chú thích bằng @PreDestroy, và các phương thức này sẽ được gọi theo thứ tự định nghĩa.




```
import javax.annotation.PostConstruct;
import javax.annotation.PreDestroy;
import org.springframework.stereotype.Component;

@Component
public class MyBean {

    @PostConstruct
    public void init() {
        // Thực hiện các tác vụ khởi tạo
    }

    @PreDestroy
    public void cleanup() {
        // Thực hiện các tác vụ dọn dẹp hoặc giải phóng tài nguyên
    }

    // Các phương thức và thuộc tính khác của bean
}
```



```
@Component
```

```
public class MyAppInitializer {
```

```
    @PostConstruct
```

```
    public void init() {
```

```
        // Thực hiện các tác vụ cấu hình khi ứng dụng khởi động
```

```
        System.out.println("Initializing application...");
```

```
        // Các tác vụ khác, ví dụ: kết nối cơ sở dữ liệu, tải cấu hình, ...
```

```
    }
```

```
    @PreDestroy
```

```
    public void cleanup() {
```

```
        // Thực hiện các tác vụ giải phóng tài nguyên khi ứng dụng dừng
```

```
        System.out.println("Cleaning up application...");
```

```
        // Các tác vụ khác, ví dụ: đóng kết nối cơ sở dữ liệu, lưu trạng thái
```

```
    }
```

```
    // Các phương thức và thuộc tính khác của bean
```

```
}
```



```
@PostConstruct
```

```
public void init() {
```

```
    // Thực hiện các tác vụ khởi tạo sau khi bean được tạo
```

```
    System.out.println("Initializing player: " + name);
```

```
}
```

```
@PreDestroy
```

```
public void cleanup() {
```

```
    // Thực hiện các tác vụ giải phóng tài nguyên trước khi bean bị hủy
```

```
    System.out.println("Cleaning up player: " + name);
```

```
    savePlayerState(); // Lưu trạng thái người chơi trước khi bị hủy
```

```
}
```




Lưu ý

Khi sử dụng “Prototype” => Spring không gọi phương thức Destroy

Khi sử dụng “Prototype” => Không cần khai báo @Lazy



Thực hành

Xây dựng project dưới dạng singleton
Viết các phương thức sử dụng:
`@PostConstruct` và `@PreDestroy`