# स्प्रिंग के एप्लिकेशन कॉन्टेक्स्ट तक पहुंच

□□□□□□□ के ApplicationContextAware □□□□□□□□□□ का उपयोग करके, एक □□□□□□ □□□□□ □□□□□□□, जैसे कि एक □□□□□□ □□□□□□□□□, में ApplicationContext को □□□□□□□ करने के □□□□□, □□□□□□ □□ □□□□□□ □□ □□□□□□□-□□□□□□□□ □□□□□□ □□ □□□□□□ □□□□□□ □□□□□□□ □□ □□□□□□ □□□□□□□ □□□□□□ □□□□□□ □□□□. □□□□ □□□□□□ □□□□□□ □□□□□□ □□□□:

□□□□□□□□ □□□□□□□□□

ApplicationContextAware □□□□□□□□□□ □□ □□□□□□□-□□□□□□□□ □□□□□ □□ □□□□□□□ □□ ApplicationContext □□ □□□□□□□□□-□□□□□□ □□□□□□□□ □□□□□□ □□ □□□□□□□□ □□□□□ □□□□. □□□□ □□□□□ □□□□ □□□□ □□□□□□ □□□□□ □□ □□□□□□ □□□□□ □□□□□□□ □□ □□□□□ □□ □□□□□□□ □□ □□□□□□ □□□□□□□ □□□□□ □□□□□ □□□□□□ □□□□ (□.□., □□□□□□□ □□□□□□□□□□ □□□□□□ □□□□□□□ □□□□□ □□□□□ □□□□□ □□□□□).

□□□□□□□□□□□□□□□□□□□□ □□ □□□□□□□ □□□□□□ □□ □□□□□□

1. □□□□□□□-□□□□□□□□ □□□□□□□ □□□□□□ □□□□□□□ □□□□□□□: □□ □□□□□□ □□□□□□□ □□□□□□□ □□ ApplicationContextAware □□-□□□□□□□□ □□□□□□ □□□□ □□□□ □□□□□□□ □□□□□□ □□□□□□□ □□□□□ □□□□□□ □□□□ (□.□., @Component □□ □□□□□□□□□ □□□□□ □□□□□). □□□□ □□□□□□ □□□□□□□ □□□□□□□□□□□□□ □□□□□□ □□□□□ □□□□ ApplicationContext □□□□□□□□ □□□□□□□.

   ```
   import org.springframework.context.ApplicationContext;
   import org.springframework.context.ApplicationContextAware;
   import org.springframework.stereotype.Component;

   @Component
   public class ApplicationContextProvider implements ApplicationContextAware {
       private static ApplicationContext context;

       @Override
       public void setApplicationContext(ApplicationContext applicationContext) {
           context = applicationContext;
       }

       public static ApplicationContext getApplicationContext() {
           return context;
       }
   }
   ```

   - @Component □□□□□□□ □□ □□□□ □□□□□ □□□□□□□ □□□□□□ □□ □□□□□ □□□□□□□ □□□□□□ □□□□.
   - setApplicationContext □□□□□□□ □□□□□□ ApplicationContext □□□□□□□ □□□□□□ □□ □□□□□ □□□□□ □□□□□ □□□□□□ □□□□.
   - □□ □□□□□□□ context □□□□□□□□□ □□□□ □□□□□□□ □□□□□ □□□□ □□□□□□ □□ □□□□□□□ □□□□□□ □□ □□□□□□□□ □□□□□ □□□□.

2. □□□□□□ □□□□□ □□□□□□□ □□□□□ □□□□□□□□ □□ □□□□□□□ □□□□□□: □□□□□□ □□□□□□ □□□□□ □□□□□□□ (□.□., □□□□□□□□□ □□□□□□□ □□□□□ □□□□□ □□□□□□ □□□□□□□□□) □□□□□, □□□□□□□ □□□□□□ □□ □□□□ □□□□□□ ApplicationContext □□ □□□□□□□□□ □□□□□□□ □□□ □□□□□ □□□□□□□-□□□□□□□□ □□□□□□ □□ □□□□□□□ □□□□□□ □□ □□□□□ □□□□ □□□□□□□.

```java
public class MyKafkaConsumer {
    public void processMessage() {
        ApplicationContext context = ApplicationContextProvider.getApplicationContext();
        SomeService service = context.getBean(SomeService.class);
        // Service ya dusre beans ka use karein jahan zaroori hai
    }
}
```

- □□□□ □□□□□ □□□□□□ □□□□ □□□□□□□ ApplicationContextProvider □□□□□□□ □□□□□□ □□□□□□□□ □□□□ □□□□□□□□□□□□ □□□□□ □□□□□□ □□□□, □□□□□□□□ □□ □□□□□□□□□□□ □□□□□□□□□ □□□□□□.

3. □□□□□□□□□□□□□: □□□□□□□□ □□ □□□□□□□□□□□□ □□□□□ □□□□□□□: □□□□□ □□□□□□ □□□□□□ □□□□□ □□□□□□□ □□ □□□□□□□-□□□□□□□□ □□□□□ □□□□□□ □□□□□□□ □□□□□ □□□□□□ □□□□, □□ □□□□ ApplicationContext □□ □□ □□□□□ □□□□□ □□□□□□□□□□ □□□□ □□□□□□ □□□□□ □□□ □□□□□ □□□□□□ □□□□□□□ □□ □□ □□□□□□□□□□□□ □□ □□□□□□□ □□ □□□□□□ □□□□□ □□□□ □□□□□□ □□□□□.

```java
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
import org.springframework.stereotype.Component;

@Component
public class KafkaConsumerCreator {
    @Autowired
    private ApplicationContext context;

    public MyKafkaConsumer createConsumer() {
        return new MyKafkaConsumer(context);
    }
}

public class MyKafkaConsumer {
    private final ApplicationContext context;

    public MyKafkaConsumer(ApplicationContext context) {
        this.context = context;
    }

    public void processMessage() {
        SomeService service = context.getBean(SomeService.class);
```

```
            // Service ka use karein
        }
    }
```

- [illegible]

[illegible]-[illegible]

[illegible] **@KafkaListener** [illegible] `ApplicationContext` [illegible] **@KafkaListener**. [illegible]

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
import org.springframework.kafka.annotation.KafkaListener;
import org.springframework.stereotype.Component;

@Component
public class MyKafkaConsumer {
    @Autowired
    private ApplicationContext context;

    @Autowired
    private SomeService someService;

    @KafkaListener(topics = "my-topic")
    public void consume(String message) {
        // Context ya someService ko directly use karein
        SomeService service = context.getBean(SomeService.class); // Agar zaroori hai
        someService.process(message);
    }
}
```

- [illegible]
- [illegible]

□□□□□□□ □□□□□□ □□□□□□□□□  □□□□□ □□□□□□ KafkaConsumer □□ □□□□□□□□□ □□□□□□□ □□□□□□ □□□□□□ □□□□ (□.□., □□□□□□ □□□□□□□ □□□□□□□□ □□ □□□□□□□□□□ □□□□ □□□□□□), □□ □□□□□ □□ □□□□□□□-□□□□□□□□ □□□□□ □□ □□□□□□ □□□□□□□ □□□□□□□ □□□□ □□□□□ ApplicationContext □□ □□□□□ □□□□□□□, □□□□□□ □□□□□□□□□ □□□□□□□□ □□□□□□□□□ □□□□□ □□□□□□□□ □□□□ □□□.

□□□□□□□□□□□□□□□□□□

- □□□□□□ □□□□□□□□□□□ □□ □□□□□: □□□□□□ □□□□ □□□□□□□□□□ □□□□, □□□□□□□ □□□□□□ □□ @KafkaListener □□ □□□□ □□□□□□□ □□ □□□□□□□-□□□□□□□□ □□□□□ □□□□□. □□□□ □□□□□□□ □□ □□□□□□ □□ □□□□□□ □□□□□□□□□□□□□ □□ □□□□□□□□□ □□□□□□ □□□□ □□□□ ApplicationContext □□□□ □□□□□□ □□□□□□ □□ □□□□□□□ □□□□□□ □□ □□□□□□□□□ □□□□□□ □□□□.
- □□□□□□□□ □□□□□□ □□□□□ □□□□□□□□ □□ □□□□□: □□ □□ □□□□□□□ ApplicationContextProvider □□□□□□□□□ □□ □□□□ □□□□□□□ □□ ApplicationContext □□ □□ □□□□□□□-□□□□□□□□ □□□□□ □□ □□□□□□□□□□□□ □□□□□ □□□□□□□. □□□□□□□□□ □□□□□□□□ □□□□□□□ □□□□□□□□□□□ □□□□□ □□□□ □□□□ □□□□□□□□□ □□□□, □□□□□□□ □□□□ □□□□□□□ □□□□□□ □□ □□□□□□ □□□□□□ □□□□ □□□□ □□□□□□□□□□□□□ □□ □□□□□□□□□ □□□□□□ □□□□.

□□□□□ □□□□□□□□ □□ □□□□□□□ □□□□□□, □□□□ ApplicationContextAware □□ □□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□ □□□□□□□ □□□□□, □□□□□□ □□ □□□□□□ □□□□□□□□□ □□□□ □□□□□□□ □□ □□□□□□ □□□□, □□□□□□□ □□□□□□□□ □□ □□□□□□□□□□□□□ □□□□□□□□□□ □□□□ □□□□□□ □□□□□, □□□□□□ □□□□□□□□□ □□□□□□ □□ □□□□□□□ □□.