

स्प्रिंग के एप्लिकेशन कॉन्टेक्स्ट तक पहुंच

स्प्रिंग के ApplicationContextAware अवधारणाएँ का उपयोग करके, एक विविध तरीकों से, जैसे कि एक बीमारी विवरण, में ApplicationContext को प्रवेश करने के लिए, विविध विवरणों को संबोधित-संबोधित करने के लिए विवरणों को विविध विवरणों में विवरण करने के लिए। यह विवरण विवरण है:

विवरण विवरण

ApplicationContextAware विवरणों को संबोधित-संबोधित करने के लिए विवरण का ApplicationContext विवरण-विवरण विवरणों को विवरण करने के लिए। यह विवरण विवरण विवरण को विवरण विवरण को विवरण विवरण को विवरण करने के लिए (इ.ए., विवरण विवरण विवरण को विवरण विवरण को विवरण करने के लिए)।

विवरण विवरण विवरण को विवरण विवरण करने के लिए

- विवरण-विवरण विवरण विवरण को विवरण करने के लिए: यह विवरण विवरण विवरण को ApplicationContextAware विवरण-विवरण विवरण करने के लिए विवरण विवरण करने के लिए विवरण विवरण करने के लिए (इ.ए., @Component विवरण विवरण विवरण करने के लिए). यह विवरण विवरण विवरण को विवरण विवरण करने के लिए ApplicationContext विवरण-विवरण विवरण करने के लिए.

```
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 विवरण को विवरण विवरण विवरण को विवरण विवरण करने के लिए विवरण करने के लिए.
- setApplicationContext विवरण को विवरण ApplicationContext विवरण विवरण को विवरण विवरण करने के लिए.
- यह विवरण context विवरण विवरण को विवरण विवरण करने के लिए विवरण करने के लिए विवरण करने के लिए.

2. ہمیں کوئی ہدایت نہیں دیتے کہ کہاں سے چکر کروں: ہمیں ہدایت دیتے ہوئے (جسے، ہمیں کوئی ہدایت نہیں دیتے) ہے، ہمیں کوئی ہدایت نہیں دیتے ApplicationContext کو ہمیں کوئی ہدایت نہیں دیتے۔ ہمیں کوئی ہدایت نہیں دیتے۔

```
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
    }
}
```

□ ہم ہدایت دیتے ہوئے ApplicationContextProvider کو ہدایت دیتے ہوئے ہدایت دیتے ہوئے ہدایت دیتے، ہمیں کوئی ہدایت دیتے ہوئے ہدایت دیتے۔

3. ہدایت دیتے: ہدایت دیتے کو ہدایت دیتے کو ہدایت دیتے: ہمیں ہدایت دیتے ہوئے ہدایت دیتے ہوئے ہدایت دیتے ہوئے ہدایت دیتے ہوئے ہدایت دیتے، ہمیں کوئی ApplicationContext کو ہدایت دیتے ہوئے ہدایت دیتے ہوئے ہدایت دیتے ہوئے ہدایت دیتے ہوئے ہدایت دیتے۔

```
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
}
}
```

□ यहाँ एक सेवा का उपयोग करने का बाबा है, जिससे आप अपनी सेवा को लोड कर सकते हैं।

मुख्य सेवा-संबद्धी क्या हैं?

मुख्य सेवा का मुख्य उपयोग यह है कि आप अपनी सेवा को लोड करने के लिए, एक सेवा-संबद्धी का उपयोग करना। इस सेवा-संबद्धी का उपयोग आपको सेवा को लोड करने के लिए एक सरल तरीका प्रदान करता है, जिससे आप अपनी सेवा को लोड कर सकते हैं।

@KafkaListener भी एक मुख्य सेवा का उपयोग करने का एक अन्य रूप है। यह सेवा को लोड करने के लिए, एक **@KafkaListener** अवधारणा का उपयोग करता है। इस सेवा का ApplicationContext का उपयोग करने के लिए सेवा को लोड करने के लिए।

```

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);
    }
}
```

□ यहाँ एक सेवा का उपयोग करने का बाबा है, जिससे आप अपनी सेवा को लोड कर सकते हैं।

□ एक सेवा का उपयोग करने का बाबा है, जिससे आप अपनी सेवा को लोड कर सकते हैं।

Java 中的 KafkaConsumer 可以通过 KafkaConsumer 构造方法注入 ApplicationContext，从而实现对 Kafka 的自动配置。ApplicationContext 提供了对 Kafka 配置的自动注入功能，使得消费者可以方便地从外部环境读取配置参数。

A horizontal row of fifteen empty square boxes, intended for children to write their names in, likely as part of a classroom activity.