

# ApplicationContextAware, Ивенты.

Чтобы получить доступ к контексту(например для публикации ивента) класс должен поддерживать интерфейс ApplicationContextAware

```
public class Manager implements ApplicationContextAware {  
  
    private ApplicationContext applicationContext;  
  
    @Override  
    public void setApplicationContext(ApplicationContext applicationContext) throws  
BeansException {  
        this.applicationContext = applicationContext;  
    }  
  
    public ApplicationContext getApplicationContext() {  
        return applicationContext;  
    }  
}
```

- Такой класс сможет публиковать ивенты через контекст через метод `publishEvent()`, в который нужно передать параметр - ивент

```
manager.getApplicationContext().publishEvent();
```

- Для получения ивентов класс должен реализовать интерфейс `ApplicationListener<>`

```
public class CustomEventListener implements ApplicationListener<CustomEvent> {
```

- В методе `onApplicationEvent()` прописывается необходимая логика.

```
@Override  
public void onApplicationEvent(CustomEvent customEvent) {  
    System.out.println("Custom event received. Кастомное событие принято");  
}
```

В следующих примерах подробнее разобрано и показано, как это применяется.