Spring aop aspectJ pointcut expression examples

Last Updated: December 26, 2020 Spring AOP AOP Annotations

In this tutorial, I am listing down some examples which will help you to write **pointcut expressions** to match any kind of method joint points into your spring application. For complete AspectJ pointcut language, please refer to the AspectJ programming guide available on AspectJ's web site.

1. How to match method signature patterns

The most typical pointcut expressions are used to match a number of methods by their signatures.

1.1. Match all methods within a class in another package

For example, the following pointcut expression matches all of the methods declared in the `EmployeeManager` interface. The preceding wildcard matches methods with any modifier (public, protected, and private) and any return type. The two dots in the argument list match any number of arguments.

```
execution(* com.howtodoinjava.EmployeeManager.*(..))
```

1.2. Match all methods within a class within same package

You can omit the package name if the target class or interface is located in the same package as this aspect.

```
execution(* EmployeeManager.*(..))
execution(public * EmployeeManager.*(..))
```

1.4. Match all public methods in EmployeeManager with return type EmployeeDTO

Use public keyword and return type in start.

```
execution(public EmployeeDTO EmployeeManager.*(..))
```

1.5. Match all public methods in EmployeeManager with return type EmployeeDTO and first parameter as EmployeeDTO

Use public keyword and return type in start. Also, specify your first parameter as well. Rest parameters can be matched through two dots.

```
execution(public EmployeeDTO EmployeeManager.*(EmployeeDTO, ..))
```

1.6. Match all public methods in EmployeeManager with return type EmployeeDTO and definite parameters

Use public keyword and return type in start. Also, specify all parameter types as well.

```
execution(public EmployeeDTO EmployeeManager.*(EmployeeDTO, Integer))
```

2. How to match class type signature patterns

When applied to Spring AOP, the scope of these pointcuts will be narrowed to matching all method executions within the certain types only.

It's much like previous example.

```
within(com.howtodoinjava.*)
```

2.2. Match all methods defined in classes inside package com.howtodoinjava and classes inside all sub-packages as well

For including, sub-packages use two dots.

```
within(com.howtodoinjava..*)
```

2.3. Match all methods with a class in another package

Much like previous example using execution keyword.

```
within(com.howtodoinjava.EmployeeManagerImpl)
```

2.4. Match all methods with a class in same package

In case of same package, drop package name.

```
within(EmployeeManagerImpl)
```

2.5. Match all methods within all all implementing classes of EmployeeManager interface

Use + (plus) sign to match all implementations of an interface.

```
within(EmployeeManagerImpl+)
```

3. How to match class name patterns

You can match all beans as well having a common naming pattern e.g.

It's quite easy one. Use an * to match anything preceding in bean name and then matching word.

```
bean(*Manager)
```

4. How to combine pointcut expressions

In AspectJ, pointcut expressions can be combined with the operators `&& (and)`, `|| (or)`, and `! (not)`. e.g.

4.1. Match all methods with names ending with Manager and DAO

Use `'||'` sign to combine both expressions.

```
bean(*Manager) || bean(*DAO)
```

I hope that above information will help you when you face any difficulty in determining the correct pointcut expression in your application.

Happy Learning !!