<http://www.cnblogs.com/yudy/archive/2012/03/22/2411175.html>

# [Spring Aop配置时的切入点表达式](http://www.cnblogs.com/yudy/archive/2012/03/22/2411175.html)

Some examples of common pointcut expressions are given below.

* the execution of any public method:
* execution(public \* \*(..))
* the execution of any method with a name beginning with "set":
* execution(\* set\*(..))
* the execution of any method defined by the AccountService interface:
* execution(\* com.xyz.service.AccountService.\*(..))
* the execution of any method defined in the service package:
* execution(\* com.xyz.service.\*.\*(..))
* the execution of any method defined in the service package or a sub-package:
* execution(\* com.xyz.service..\*.\*(..))
* any join point (method execution only in Spring AOP) within the service package:
* within(com.xyz.service.\*)
* any join point (method execution only in Spring AOP) within the service package or a sub-package:

within(com.xyz.service..\*)

* any join point (method execution only in Spring AOP) where the proxy implements the AccountService interface:

this(com.xyz.service.AccountService)

*'this' is more commonly used in a binding form :- see the following section on advice for how to make the proxy object available in the advice body.*

* any join point (method execution only in Spring AOP) where the target object implements the AccountService interface:

target(com.xyz.service.AccountService)

*'target' is more commonly used in a binding form :- see the following section on advice for how to make the target object available in the advice body.*

* any join point (method execution only in Spring AOP) which takes a single parameter, and where the argument passed at runtime is Serializable:

args(java.io.Serializable)

*'args' is more commonly used in a binding form :- see the following section on advice for how to make the method arguments available in the advice body.*

Note that the pointcut given in this example is different to execution(\* \*(java.io.Serializable)): the args version matches if the argument passed at runtime is Serializable, the execution version matches if the method signature declares a single parameter of type Serializable.

* any join point (method execution only in Spring AOP) where the target object has an @Transactional annotation:

@target(org.springframework.transaction.annotation.Transactional)

*'@target' can also be used in a binding form :- see the following section on advice for how to make the annotation object available in the advice body.*

* any join point (method execution only in Spring AOP) where the declared type of the target object has an @Transactional annotation:

@within(org.springframework.transaction.annotation.Transactional)

*'@within' can also be used in a binding form :- see the following section on advice for how to make the annotation object available in the advice body.*

* any join point (method execution only in Spring AOP) where the executing method has an @Transactional annotation:

@annotation(org.springframework.transaction.annotation.Transactional)

*'@annotation' can also be used in a binding form :- see the following section on advice for how to make the annotation object available in the advice body.*

* any join point (method execution only in Spring AOP) which takes a single parameter, and where the runtime type of the argument passed has the @Classified annotation:

@args(com.xyz.security.Classified)

*'@args' can also be used in a binding form :- see the following section on advice for how to make the annotation object(s) available in the advice body.*

* any join point (method execution only in Spring AOP) on a Spring bean named 'tradeService':

bean(tradeService)

* any join point (method execution only in Spring AOP) on Spring beans having names that match the wildcard expression '\*Service':

bean(\*Service)

对应的中文：

任意公共方法的执行：   
execution(public \* \*(..))   
任何一个以“set”开始的方法的执行：   
execution(\* set\*(..))   
AccountService 接口的任意方法的执行：   
execution(\* com.xyz.service.AccountService.\*(..))   
定义在service包里的任意方法的执行：   
execution(\* com.xyz.service.\*.\*(..))   
定义在service包或者子包里的任意方法的执行：   
execution(\* com.xyz.service..\*.\*(..))   
在service包里的任意连接点（在Spring AOP中只是方法执行） ：   
within(com.xyz.service.\*)   
在service包或者子包里的任意连接点（在Spring AOP中只是方法执行） ：   
within(com.xyz.service..\*)   
实现了 AccountService 接口的代理对象的任意连接点（在Spring AOP中只是方法执行） ：   
this(com.xyz.service.AccountService)   
'this'在binding form中用的更多：- 请常见以下讨论通知的章节中关于如何使得代理对象可以在通知体内访问到的部分。   
实现了 AccountService 接口的目标对象的任意连接点（在Spring AOP中只是方法执行） ：   
target(com.xyz.service.AccountService)   
'target'在binding form中用的更多：- 请常见以下讨论通知的章节中关于如何使得目标对象可以在通知体内访问到的部分。   
任何一个只接受一个参数，且在运行时传入的参数实现了 Serializable 接口的连接点 （在Spring AOP中只是方法执行）   
args(java.io.Serializable)   
'args'在binding form中用的更多：- 请常见以下讨论通知的章节中关于如何使得方法参数可以在通知体内访问到的部分。 请注意在例子中给出的切入点不同于 execution(\* \*(java.io.Serializable))： args只有在动态运行时候传入参数是可序列化的（Serializable）才匹配，而execution 在传入参数的签名声明的类型实现了 Serializable 接口时候匹配。   
有一个 @Transactional 注解的目标对象中的任意连接点（在Spring AOP中只是方法执行）   
@target(org.springframework.transaction.annotation.Transactional)   
'@target' 也可以在binding form中使用：请常见以下讨论通知的章节中关于如何使得annotation对象可以在通知体内访问到的部分。   
任何一个目标对象声明的类型有一个 @Transactional 注解的连接点（在Spring AOP中只是方法执行）   
@within(org.springframework.transaction.annotation.Transactional)   
'@within'也可以在binding form中使用：- 请常见以下讨论通知的章节中关于如何使得annotation对象可以在通知体内访问到的部分。   
任何一个执行的方法有一个 @Transactional annotation的连接点（在Spring AOP中只是方法执行）   
@annotation(org.springframework.transaction.annotation.Transactional)   
'@annotation' 也可以在binding form中使用：- 请常见以下讨论通知的章节中关于如何使得annotation对象可以在通知体内访问到的部分。   
任何一个接受一个参数，并且传入的参数在运行时的类型实现了 @Classified annotation的连接点（在Spring AOP中只是方法执行）   
@args(com.xyz.security.Classified)