```
/*
 * What is annotation? 1.5
 * -----------------
 * AKA meta data ..
 *
 *       annotation
 *          |
 *     ----------------------
 *     |                     |
 *     predefine             Custom annotation
 *        |
 *   -----------
 *   |           |
 *   basics    meta for other
 *               annotations
 *
 * @Override
@SuppressWarning
@SuppressWarnings({"rawtypes", "unchecked"})
@Deprecated
@Target
@Retention
@Documented
@Inherited
@FunctionalInterface (Java 8)
 */
```

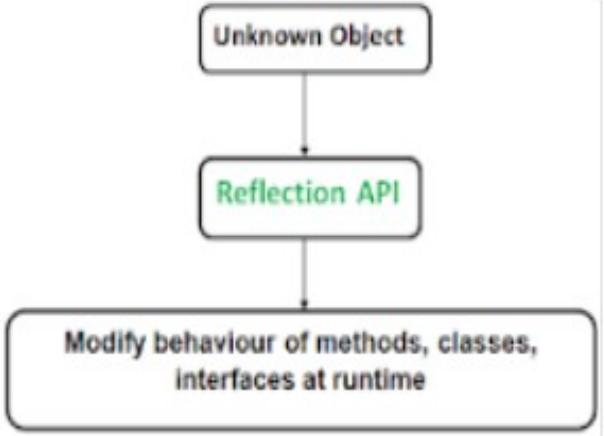
what is java reflection

All Videos News Images Shopping More Settings Tools

About 5,26,00,000 results (0.55 seconds)

Reflection is an API which is used to examine or modify the behavior of methods, classes, interfaces at runtime. The required classes for **reflection** are provided under **java**. lang. reflect package. ... Through **reflection** we can invoke methods at runtime irrespective of the access specifier used with them.

Mar 22, 2017



www.geeksforgeeks.org › reflection-in-java

Reflection in Java - GeeksforGeeks

```java
    @Foo(val="my new value", isValid=true)
    public  void fooClient(){
        System.out.println("foo client method");
    }
}

public class DemoAnn {

    public static void main(String[] args) {
        //I wann to process that annotation whenever my fooClient method runs...

        //How to use java ref! next session i will discuss in details!

        // 3 ways to get Class object in java? "what the hell this Class class"
        Class c=new ClientFoo().getClass();

        Method[] methods=c.getDeclaredMethods();

        for(Method m:methods){
            if(m.isAnnotationPresent(Foo.class)){
                Foo foo=m.getAnnotation(Foo.class);
                System.out.println(foo.val());
                System.out.println(foo.isValid());

            }
        }


    }
```

```
    // Dog
    Dog d = new Dog();

    Class dClass = d.getClass();

    Method[] methods = dClass.getMethods();// detMethod vs
                                           // dClass.getDeclaredMethods();

    for (Method m : methods) {
        System.out.println(m.getName());
        System.out.println(m.getModifiers());
        System.out.println(Modifier.toString(m.getModifiers()));
        System.out.println(m.getParameterCount());
        m.getParameters();
    }

    Constructor [] constructors=dClass.getDeclaredConstructors();


    for(Constructor c: constructors){
        System.out.println(c.getName());
        System.out.println(c.getModifiers());

    }
}
```