

```

package com.sxt.reflection;

import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.lang.reflect.Method;

/**
 * 利用反射的API，获取类的信息（类的名字，属性，方法，构造器等）
 * @author 江
 *
 */

public class Demo02 {
    public static void main(String[] args) {
        String path="com.sxt.bean.User";

        try {
            Class clazz=Class.forName(path);

            //获取类的名字
            System.err.println(clazz.getName()); //获得包名+类名：
com.sxt.bean.User
            System.err.println(clazz.getSimpleName()); //获得类名： User

            //获取属性信息
            Field[] fields01=clazz.getFields(); //只能获得public的field
            Field[] fields02=clazz.getDeclaredFields(); //可以获得非public
的field

            System.err.println(fields01.length); //0
            System.err.println(fields02.length); //3

            Field field01=clazz.getField("id"); //获得指定的field
            Field field02=clazz.getDeclaredField("id");
            System.err.println(field01); //错误
            System.err.println(field02);

```

```

//获取方法信息
Method[] methods01=clazz.getMethods();
Method[] methods02=clazz.getDeclaredMethods();

Method method01=clazz.getMethod("getUname", null);
Method method03=clazz.getDeclaredMethod("getUname", null);
//如果方法有参，则必须传递参数类型对应的class对象
Method method02=clazz.getMethod("setUname", String.class);
Method
method04=clazz.getDeclaredMethod("SetUname", String.class);

//获得构造方法的信息
Constructor[] constructors=clazz.getConstructors();
Constructor constructor01=clazz.getConstructor(null);
Constructor constructor02=clazz.getDeclaredConstructor(null);
Constructor
constructor03=clazz.getConstructor(int.class, int.class, String.class);
Constructor
constructor04=clazz.getDeclaredConstructor(int.class, int.class, String.class);

} catch (Exception e) {
    System.err.println("反射错误");
}

}

}

```