

Forum:[Hack Forums](#)

Thread: [\[Reflection\] Get field in a field](#)

Josh? - September 3, 2014, 5:57 am

Right, in the code I have, Class A. Class A has Field B. Field B is an instance of Class C. Class C has Field D.

I want to access A.B.D via reflection. How do I do this?

GammaG - September 3, 2014, 6:16 am

I find this a nice example on how to use it.

It takes any object as a parameter and uses the Java reflection API print out every field name and value.

```
Code import java.lang.reflect.Array;
import java.lang.reflect.Field;
```

```
public static String dump(Object o, int callCount) {
    callCount++;
    StringBuffer tabs = new StringBuffer();
    for (int k = 0; k < callCount; k++) {
        tabs.append("\t");
    }
    StringBuffer buffer = new StringBuffer();
    Class oClass = o.getClass();
    if (oClass.isArray()) {
        buffer.append("\n");
        buffer.append(tabs.toString());
        buffer.append("[");
        for (int i = 0; i < Array.getLength(o); i++) {
            if (i < 0)
                buffer.append(",");
            Object value = Array.get(o, i);
            if (value.getClass().isPrimitive() ||
                value.getClass() == java.lang.Long.class ||
                value.getClass() == java.lang.String.class ||
                value.getClass() == java.lang.Integer.class ||
                value.getClass() == java.lang.Boolean.class
            ) {
                buffer.append(value);
            } else {
                buffer.append(dump(value, callCount));
            }
        }
        buffer.append(tabs.toString());
        buffer.append("]\n");
    } else {
        buffer.append("\n");
        buffer.append(tabs.toString());
    }
}
```

```

buffer.append("{\n");
while (oClass != null) {
Field[] fields = oClass.getDeclaredFields();
for (int i = 0; i < fields.length; i++) {
    buffer.append(tabs.toString());
    fields[i].setAccessible(true);
    buffer.append(fields[i].getName());
    buffer.append("=");
    try {
        Object value = fields[i].get(o);
        if (value != null) {
            if (value.getClass().isPrimitive() ||
                value.getClass() == java.lang.Long.class ||
                value.getClass() == java.lang.String.class ||
                value.getClass() == java.lang.Integer.class ||
                value.getClass() == java.lang.Boolean.class
            ) {
                buffer.append(value);
            } else {
                buffer.append(dump(value, callCount));
            }
        }
    } catch (IllegalAccessException e) {
        buffer.append(e.getMessage());
    }
    buffer.append("\n");
}
oClass = oClass.getSuperclass();
}
buffer.append(tabs.toString());
buffer.append("}\n");
}
return buffer.toString();
}

```

found here:

<https://stackoverflow.com/questions/37628/what-is-reflection-and-why-is-it-useful>

MrOrang - September 3, 2014, 9:33 am

Code obj = a object

```
Class tehClazz = obj.getClass();
```

```
Field tehFirstField tehClazz.getDeclaredField("ur args to ur b class field thing");
```

```
tehFirstField.setAccessible(true);
```

```
Class urBClazz = tehFirstField.getValue(obj).getClass();
```

```
Field tehFieldUWant = urBClazz.getDeclaredField("teh args");
```

```
tehFieldUWant.setAccessible(true);
```

//The value you want

Object value = tehFieldUWant.getValue(tehFirstField.getValue(obj)); Out of my head, you might want to fix some stuff though.