

Table of Contents

JVx's reflective class allows you to create class instances or call methods via reflection but without directly using `java.lang.reflect`.

How does it work?

We have a short example for you. Our class:

MyClient.java

```
package com.sibvisions.app;

public class MyClient
{
    public Result doUpload(String pKey, String pValue)
    {
        ...
    }

    public Result doDownload(String pPath)
    {
        ...
    }

    public Result doDownload(File pFile)
    {
        ...
    }
}
```

A simple method call:

```
Object oClient = Reflective.construct("com.sibvisions.app.MyClient");
Reflective.call(oClient, "doUpload", "Key", "Value");
```

Now we call `doDownload`:

```
Object oClient = Reflective.construct("com.sibvisions.app.MyClient");

//no problem
Reflective.call(oClient, "doDownload", "file.txt");

//without Parameter, maybe we call the wrong method
Reflective.call(oClient, "doDownload", new
Reflective.Parameter(String.class, null));
```

As you can see, we use the class `Reflective.Parameter`. It is useful if you have methods with the same name and same number of parameters but with different parameter types. Without this class, it is not

guaranteed that the desired method is called.

From:

<http://doc.sibvisions.com/> - **Documentation**

Permanent link:

<http://doc.sibvisions.com/jvx/common/util/reflective>



Last update: **2020/07/28 09:52**