

java.lang.reflect

Interface InvocationHandler

All Known Implementing Classes:

CompositeDataInvocationHandler, EventHandler, MBeanServerInvocationHandler, RemoteObjectInvocationHandler

public interface **InvocationHandler**

InvocationHandler is the interface implemented by the *invocation handler* of a proxy instance.

Each proxy instance has an associated invocation handler. When a method is invoked on a proxy instance, the method invocation is encoded and dispatched to the `invoke` method of its invocation handler.

Since:

1.3

See Also:

Proxy

Method Summary

Methods

Modifier and Type	Method and Description
Object	<code>invoke(Object proxy, Method method, Object[] args)</code> Processes a method invocation on a proxy instance and returns the result.

Method Detail

invoke

```
Object invoke(Object proxy,
              Method method,
              Object[] args)
    throws Throwable
```

Processes a method invocation on a proxy instance and returns the result. This method will be invoked on an invocation handler when a method is invoked on a proxy instance that it is associated with.

Parameters:

`proxy` - the proxy instance that the method was invoked on

`method` - the Method instance corresponding to the interface method invoked on the proxy instance. The declaring class of the Method object will be the interface that the method was declared in, which may be a superinterface of the proxy interface that the proxy class inherits the method through.

`args` - an array of objects containing the values of the arguments passed in the method invocation on the proxy instance, or `null` if interface method takes no arguments. Arguments of primitive types are wrapped in instances of the appropriate primitive wrapper class, such as `java.lang.Integer` or `java.lang.Boolean`.

Returns:

the value to return from the method invocation on the proxy instance. If the declared return type of the interface method is a primitive type, then the value returned by this method must be an instance of the corresponding primitive wrapper class; otherwise, it must be a type assignable to the declared return type. If the value returned by this method is `null` and the interface method's return type is primitive, then a `NullPointerException` will be thrown by the method invocation on the proxy instance. If the value returned by this method is otherwise not compatible with the interface method's declared return type as described above, a `ClassCastException` will be thrown by the method invocation on the proxy instance.

Throws:

`Throwable` - the exception to throw from the method invocation on the proxy instance. The exception's type must be assignable either to any of the exception types declared in the `throws` clause of the interface method or to the unchecked exception types `java.lang.RuntimeException` or `java.lang.Error`. If a checked exception is thrown by this method that is not assignable to any of the exception types declared in the `throws` clause of the interface method, then an `UndeclaredThrowableException` containing the exception that was thrown by this method will be thrown by the method invocation on the proxy instance.

See Also:

`UndeclaredThrowableException`

Submit a bug or feature

For further API reference and developer documentation, see [Java SE Documentation](#). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright © 1993, 2020, Oracle and/or its affiliates. All rights reserved. Use is subject to [license terms](#). Also see the [documentation redistribution policy](#). [Modify Cookie-Einstellungen](#). [Modify Ad Choices](#).