java.lang.reflect Package Classes

java.lang.reflect - Home

java.lang.reflect - AccessibleObject

java.lang.reflect - Array

java.lang.reflect - Constructor<T>

java.lang.reflect - Field

java.lang.reflect - Method

java.lang.reflect - Modifier

java.lang.reflect - Proxy

java.lang.reflect Package Extras

java.lang.reflect - Interfaces

java.lang.reflect - Exceptions

java.lang.reflect - Error

java.lang.reflect Useful Resources

java.lang.reflect - Quick Guide

java.lang.reflect - Useful Resources

java.lang.reflect - Discussion

java.lang.reflect.Proxy.newProxyInstance() Method Example

Previous Page   Next Page

Description

The **java.lang.reflect.Proxy.newProxyInstance(ClassLoader loader, Class<?>[] interfaces, InvocationHandler h)** method returns an instance of a proxy class for the specified interfaces that dispatches method invocations to the specified invocation handler.

Declaration

Following is the declaration for **java.lang.reflect.Proxy.newProxyInstance(ClassLoader loader, Class<?>[] interfaces, InvocationHandler h)** method.

```
public static Object newProxyInstance(ClassLoader loader, Class<?>[] interfaces,
   InvocationHandler h)
      throws IllegalArgumentException
```

Parameters

loader - the class loader to define the proxy class.

interfaces - the list of interfaces for the proxy class to implement.

h - the invocation handler to dispatch method invocations to.

Returns

A proxy instance with the specified invocation handler of a proxy class that is defined by the specified class loader and that implements the specified interfaces.

Exceptions

IllegalArgumentException - if any of the restrictions on the parameters that may be passed to getProxyClass are violated.

NullPointerException - if the interfaces array argument or any of its elements are null, or if the invocation handler, h, is null.

Example

The following example shows the usage of java.lang.reflect.Proxy.newProxyInstance(ClassLoader loader, Class<?>[] interfaces, InvocationHandler h) method.

```java
package com.tutorialspoint;

import java.lang.reflect.InvocationHandler;
import java.lang.reflect.Method;
import java.lang.reflect.Proxy;

public class ProxyDemo {
   public static void main(String[] args) throws IllegalArgumentException {
      InvocationHandler handler = new SampleInvocationHandler() ;
      SampleInterface proxy = (SampleInterface) Proxy.newProxyInstance(
         SampleInterface.class.getClassLoader(),
         new Class[] { SampleInterface.class },
         handler);
      Class invocationHandler = Proxy.getInvocationHandler(proxy).getClass();

      System.out.println(invocationHandler.getName());
   }
}

class SampleInvocationHandler implements InvocationHandler {

   @Override
   public Object invoke(Object proxy, Method method, Object[] args)
      throws Throwable {
      System.out.println("Welcome to TutorialsPoint");
      return null;
   }
}

interface SampleInterface {
   void showMessage();
}

class SampleClass implements SampleInterface {
   public void showMessage(){
      System.out.println("Hello World");
   }
}
```

Let us compile and run the above program, this will produce the following result −

```
com.tutorialspoint.SampleInvocationHandler
```

Previous Page   Next Page

Advertisements

Tutorials Point (India) Pvt. Ltd.

YouTube 66K

Write for us   FAQ's   Helping   Contact

Enter email for newslett go