JAVA.UTIL.SERVICELOADER.RELOAD METHOD

Description

The **java.util.ServiceLoader.reload** method clears this loader's provider cache so that all providers will be reloaded.

After invoking this method, subsequent invocations of the iterator method will lazily look up and instantiate providers from scratch, just as is done by a newly-created loader.

This method is intended for use in situations in which new providers can be installed into a running Java virtual machine.

Declaration

Following is the declaration for **java.util.ServiceLoader.reload** method

```
public void reload()
```

Parameters

- **NA**

Return Value

This method does not return a value

Exception

- **NA**

Example

In order the service to be registered, we need a META-INF/service folder in our classpath. In this particular folder, we need a text file with the name of the interface we implementing containing a single line listing the concrete class name of the implementation. In our case the name of the file is **com.tutorialspoint.ServiceProvider** and contains this line:

```
com.tutorialspoint.ServiceImplementation
```

Our service code is the following:

```
package com.tutorialspoint;

public class ServiceImplementation extends ServiceProvider {

   public String getMessage() {
   return "Hello World";
   }
}
```

The following code loads the service that is registered and uses it to get the message from the service:

```
package com.tutorialspoint;

import java.util.Iterator;
import java.util.ServiceLoader;

public abstract class ServiceProvider {
```

```
    public static ServiceProvider getDefault() {

    // load our plugin
    ServiceLoader<ServiceProvider> serviceLoader =
    ServiceLoader.load(ServiceProvider.class);

    // reload the service
    System.out.println("Reloading...");
    serviceLoader.reload();
    System.out.println("Reloaded.");

    // checking if load was successful
    for (ServiceProvider provider : serviceLoader) {
    return provider;
    }
    throw new Error("Something is wrong with registering the addon");
    }

    public abstract String getMessage();

    public static void main(String[] ignored) {
    // create a new provider and call getMessage()
    ServiceProvider provider = ServiceProvider.getDefault();
    System.out.println(provider.getMessage());

    }
}
```

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

```
Reloading...
Reloaded.
Hello World
```