JAVA.UTIL.SERVICELOADER.LOADINSTALLED METHOD

Description

The **java.util.ServiceLoader.loadInstalled***Class* < *S* > *service* method creates a new service loader for the given service type, using the extension class loader. This method is intended for use when only installed providers are desired. The resulting service will only find and load providers that have been installed into the current Java virtual machine; providers on the application's class path will be ignored.

Declaration

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

```
public static <S> ServiceLoader<S> loadInstalled(Class<S> service)
```

Parameters

- **service** -- The interface or abstract class representing the service

Return Value

This method returns a new service loader

Exception

- **NA**

Example

This example requires the service to be installed in our Java Virtual Machine

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.ServiceLoader;

public abstract class ServiceProvider {

   public static ServiceProvider getDefault() {

   // load our plugin provided it is installed in our Java Virtual Machine
   ServiceLoader<ServiceProvider> serviceLoader =
   ServiceLoader.loadInstalled(ServiceProvider.class);

   //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:

```
Hello World
```