What is class loaders?

* It responsible to loading Java classes into JVM during runtime.
* All the Java classes are not loaded all at once

There are 3 types of class loaders

1. Bootstrap class loader
2. Extension class loader
3. Application class loader

Bootstrap class loader:

* Java Classes are loaded by instance “java.lang.ClassLoader”.
* Responsible for loading rt.jar and other core libraries located in the $JAVA\_HOME/jre/lib/
* It is parent of all other Class Loader instances.
* Bootstrap class loader is part of core JVM and is written in native code .

Extension Class Loader :

* It is child of Bootstrap Loader and takes care of loading the extensions of standard core Java classes to make avaialbe to all the application running on the platform.
* It loads the JDK extension directory - $JAVA\_HOME/lib/ext/directory.java.ext.dirs- system property.

System class Loader:

* It takes care of all the application level classes into JVM. It loads the classpath environment variable , -classpath or -cp command line option .
* It is also child of extension class loader

How it works ?

* Classloaders are part of JRE.
* When JVM request classes the class loader tries to locate the class and load the class definition in to runtime using fully qualified class name .
* Java.lang.ClassLoader.loadClass() method is responsible method for loading the class definition in to runtime .
* If class is not already loaded it delegates the request to paresnt class loader and happens recursively.
* If the parent class loader doesn’t find the class then the chils class will call the java.net.URLClassLoader.findClass() method to look for class in the filesystem itself.
* If the child class loader is notable to find the class , then it throws java.lang.NoClassDefFoundError or Java.lang.ClassNotFoundException.

Features of Class Loader

* Delegation Model
  + ClassLoader instance will delegate the search of class or resource to the paresnt class loader .
  + If the parent class not able to load the class then system class loader tries to load the class itself.
* Visibility
  + Children class loaders are visisble to classes loaded by their parent classloders.

Example : Class A loaded by application calss loader

Class B is loaded by Extension Class loader

A and B are visible for other classes loaded by application loader .

Class B is only vivible for other classes loaded by extension class loader .

Custom Class Laoder :

1. Helping to modify existing bytecode- weaving agents
2. Creating classes dynamically 🡪 JDBC class loading
3. Implementing class versioning mechanisam while loading different bytecodes for classes with the same name and packages.

Implementation of method searches for classes in following oreder

. Invokes findLoadedClass(String) metnhod to see the class is already loaded

Invokes laodClass(String) method on the parent class loader

Invoke findClass9String) method to find the alss .

DefineClass() method

findClass() Method

getParent()Method

getResource Method

Context Class Loaders