

Class Loaders in Java

-By SivaReddy

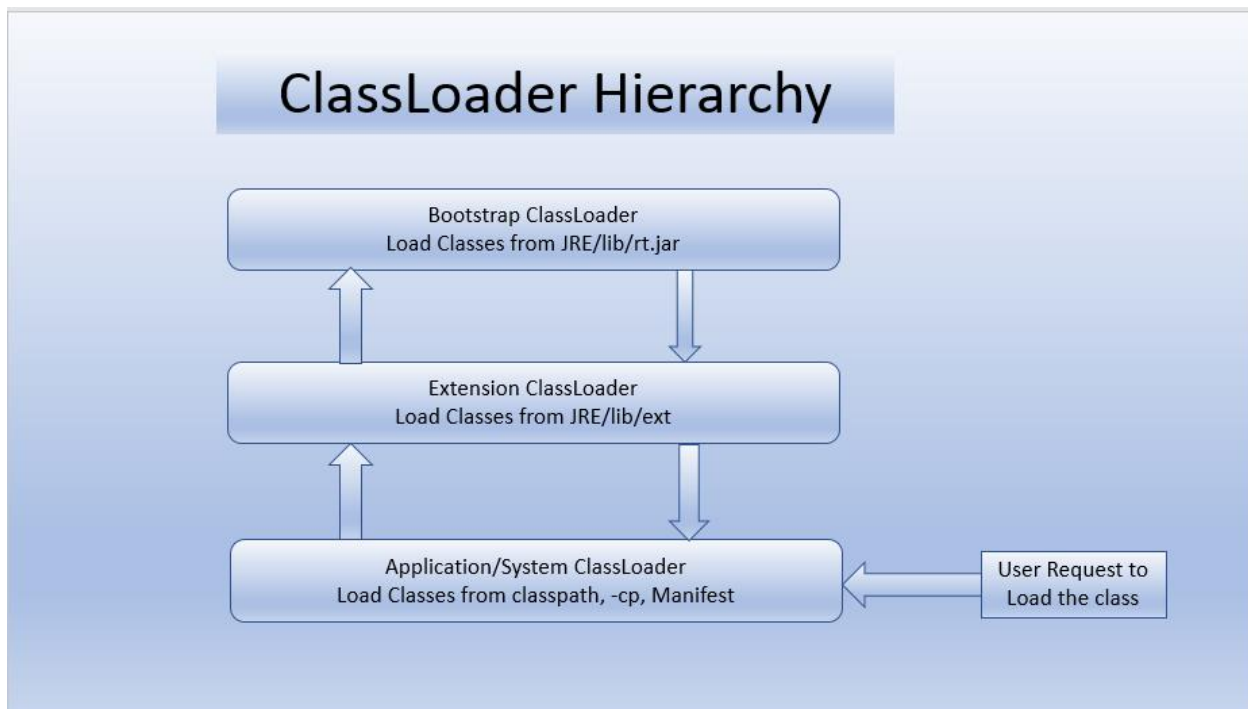
What is Class Loader?

A class loader is an object that is responsible for loading classes. The class `ClassLoader` is an abstract class. Given the binary name of a class, a class loader should attempt to locate or generate data that constitutes a definition for the class. A typical strategy is to transform the name into a file name and then read a "class file" of that name from a file system.

What are the different Class Loaders available?

- Bootstrap Class Loader
- Extension Class Loader
- Application/System Class Loader

Explain how Class Loaders works in Java?



Technical Part:

- Works based on Delegation Model
- Uniqueness
- Visibility

```
protected synchronized Class<?>loadClass
    (String name, boolean resolve)
    throws ClassNotFoundException{
```

```

// First check if the class is already loaded
Class c = findLoadedClass(name);
if (c == null) {
    try {
        if (parent != null) {
            c = parent.loadClass(name, false);
        } else {
            c = findBootstrapClass0(name);
        }
    } catch (ClassNotFoundException e) {
        // If still not found, then invoke
        // findClass to find the class.
        c = findClass(name);
    }
}
if (resolve) {
    resolveClass(c);
}
return c;
}

```

Why we need custom class loaders and How to create Custom Class Loaders?

Custom Class Loader can be created by SubClassing the **ClassLoader** Class

```

public class MyClassLoader extends ClassLoader{

    public MyClassLoader() {
        super(MyClassLoader.class.getClassLoader());
    }

}

```

How to Load Class Dynamically?

- Using `Class.forName(<Fully qualified name of the class>)`
- `ClassLoader classloader = ClassLoader.getSystemClassLoader();`
`classloader.loadClass(<Fully Qualified Class Name>)`

What is the Class Loading precedence in Tomcat?

1. Bootstrap classes of your JVM(Runtime classes provided by the Java Virtual Machine, plus any classes from JAR files present in the System Extensions directory)
2. System class loader classes (`$CATALINA_HOME/bin/bootstrap.jar`, `$CATALINA_BASE/bin/tomcat-juli.jar`, `$CATALINA_HOME/bin/commons-daemon.jar`)
3. `/WEB-INF/classes` of the webapp
4. `/WEB-INF/lib/*.jar` of the webapp
5. Common class loader classes (`$CATALINA_BASE/lib`, `$CATALINA_HOME/lib`)

