

Java Classloader

Java Classloader

- A classloader is the component which dynamically loads Java classes in the JVM (Java Virtual Machine)
- A Java class is necessarily loaded by a classloader
- There are at least three classloaders per JVM:
 - Bootstrap classloader: Loads core Java classes
 - Extensions classloader
 - System classloader: Loads the Java classpath

Java Classloader – API (1/2)

- Developers can write their own classloaders, for example to dynamically add classes to a running application
- The Java Classloader API consists in one single class:
 - `java.lang.ClassLoader`
- **Methods to override:**
 - `findClass(String name)`
 - `findResource(String name)`
 - `findResources(String name)`

Java Classloader – API (2/2)

- Simple real-life example:
 - The Hudson Open Source project
 - `hudson.PluginManager.UberClassLoader`

Java Classloader – Delegation

- Each classloader has a parent classloader, except the bootstrap classloader
- A classloader can delegate the search for a class/resource to its parent
- PARENT_FIRST/PARENT_LAST:
 - PARENT_FIRST: By default, a classloader delegates the search to his parent classloader before searching by itself
 - The contrary is called PARENT_LAST

Using dynamically loaded classes

- Dynamically loaded classes can be easily used if they:
 - Implement an interface
 - Extend a well-known class
- In case none is available, the Reflection API can be used
- More generally, the Reflection API is to be used to “examine or modify the runtime behavior of applications running in the Java virtual machine”