

Writing Our Own Classloader



Kevin Jones

@kevinrjones

Overview

- Use the URLClassLoader
 - Understand how this works
- Load from a file based URL
- Load from a network URL
- Write our own classloader
 - Load from a database

Using Interfaces and Implementations

- If using our own classloader
 - Cannot reference implementation in the code
 - This will cause the class to be loaded with the app class loader
- Split code into implementation and interface
 - Interface definitions go on the classpath
 - Implementation is loaded by our classloader

Writing Our Own Classloader

- Will not load all classes
 - Delegate to the 'system' classloader
- If the parent can't load
 - Load the class bytes from the database
 - Use ClassLoader's `defineClass` method to load the class
- Example uses SQL Server

Summary

- URLClassLoader typically used in your application
 - Takes an array of URLs
 - Can be [file:///](#), [http://](#) or some other protocol
- Can write our own classloader
 - Extend `java.lang.ClassLoader`
 - May delegate first