Connecting Java File to JavaScript File

To access primitive and reference Java types from JavaScript, call the Java.type() function, which returns a type object that corresponds to the full name of the class passed in as a string. The following example shows you how to get various type objects:

var ArrayList = Java.type("java.util.ArrayList");

var intType = Java.type("int");

var StringArrayType = Java.type("java.lang.String[]");

var int2DArrayType = Java.type("int[][]");

The type object returned by the Java.type() function can be used in JavaScript code similar to how a class name is used in Java. For example, you can can use it to instantiate new objects as follows:

var anArrayList = new Java.type("java.util.ArrayList");

Java type objects can be used to instantiate new Java objects. The following example shows you how to instantiate new objects using the default constructor and by passing arguments to another constructor:

var ArrayList = Java.type("java.util.ArrayList");

var defaultSizeArrayList = new ArrayList;

var customSizeArrayList = new ArrayList(16);

You can use the type object returned by the Java.type() function to access static fields and methods as follows:

var File = Java.type("java.io.File");

File.createTempFile("nashorn", ".tmp");

To access a static inner class, use the dollar sign ($) in the argument passed to the Java.type() method. The following example shows how to return the type object of the Float inner class in java.awt.geom.Arc2D:

var Float = Java.type("java.awt.geom.Arc2D$Float");

If you already have the outer class type object, then you can access the inner class as a property of the outer class as follows:

var Arc2D = Java.type("java.awt.geom.Arc2D")

var Float = Arc2D.Float

Instructions and code above sourced from <https://docs.oracle.com/javase/8/docs/technotes/guides/scripting/prog_guide/javascript.html>