

JNI using JNR

Create a C Library

testLib.h

```
#ifndef TESTLIB
#define TESTLIB
    int foo(int a);
#endif
```

testLib.c

```
#include "testLib.h"
int foo(int a) {
    return a + 1;
}
```

```
gcc -c testLib.c -o testLib.o
```

Make it a dynamic lib

```
gcc -dynamiclib -undefined suppress -flat_namespace testLib.o -o libtestLib.dylib
```

Call our Library from Java

```
package com.test.jnr;

import jnr.ffi.LibraryLoader;

public class LibS {

    public static interface LibC {
        int foo(int i);
    }

    public static void main(String[] args) {
        LibC libc = LibraryLoader.create(LibC.class)
            .search("/Users/john.mehan/projects/bledata/cloud/services/cloud-service/src/main/java/com/irdeto/keystone/jni/testLib")
            .load("testLib");

        System.out.println("100 +1 = " + libc.foo(100));
    }
}
```

Call some standard LibC libraries

```
package com.test.jnr;

import jnr.ffi.LibraryLoader;

public class LibS {

    public static interface LibC {
        int puts(String s);
    }

    public static void main(String[] args) {
        LibC libc = LibraryLoader.create(LibC.class).load("C");
        libc.puts("Hello, World");
    }
}
```