JNA для PKCS#11 (Java)

1. Для работы с токенами в Java разработана специальная библиотека-обертка **pkcs11jna**. Эта обертка предоставляет интерфейс над динамическими PKCS#11 библиотеками.
   Исходный код расположен на: https://github.com/AktivCo/pkcs11jna

Работа с библиотекой **pkcs11jna** достаточно проста. С помощью **net.java.dev.jna** нужно загрузить PKCS#11 библиотеку и далее работать с возвращенным объектом по интерфейсу класса RtPkcs11:

RtPkcs11

```
import com.sun.jna.Native;
import ru.rutoken.pkcs11jna.RtPkcs11;

RtPkcs11 rtpkcs11 = Native.load(Native.extractFromResourcePath("rtpkcs11ecp").getAbsolutePath(),
RtPkcs11.class);
```

Класс RtPkcs11 предоставляет интерфейс над PKCS#11 функциям обернутой библиотекой. Для примера покажем, как можно проинициализировать токен:

Пример инициализации токена

```
import com.sun.jna.NativeLong;
import com.sun.jna.Native;
import com.sun.jna.ptr.NativeLongByReference;
import ru.rutoken.pkcs11jna.CK_C_INITIALIZE_ARGS;
import ru.rutoken.pkcs11jna.Pkcs11Constants;
import ru.rutoken.pkcs11jna.RtPkcs11;

public class Init {
    private final static byte[] TOKEN_LABEL = {'M', 'y', ' ', 'R', 'u', 't', 'o', 'k',
            'e', 'n', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
            ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
            ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '};

    public final static CK_C_INITIALIZE_ARGS INITIALIZE_ARGUMENTS =
            new CK_C_INITIALIZE_ARGS(null, null, null, null, new NativeLong(Pkcs11Constants.
CKF_OS_LOCKING_OK), null);


    public static final byte[] DEFAULT_USER_PIN = {'1', '2', '3', '4', '5', '6', '7', '8'};
    public static final byte[] DEFAULT_SO_PIN = {'8', '7', '6', '5', '4', '3', '2', '1'};

    public static void main(String[] args) {
        NativeLong hSession = new NativeLong(Pkcs11Constants.CK_INVALID_HANDLE);
        RtPkcs11 rtpkcs11 = null;


        try {
            System.out.println("Example of token initialization using rtpkcs11ecp via JNA");
            rtpkcs11 = Native.load(Native.extractFromResourcePath("rtpkcs11ecp").getAbsolutePath(),
RtPkcs11.class);

            System.out.println("Library initialization and acquiring of function list");
            NativeLong rv = rtpkcs11.C_Initialize(INITIALIZE_ARGUMENTS);

            if (!Pkcs11Constants.equalsPkcsRV(Pkcs11Constants.CKR_OK, rv)) {
                throw new Exception("C_Initialize failed");
            }

            System.out.println("Acquiring list of slots with connected tokens");

            NativeLongByReference slotsCount = new NativeLongByReference();
```

```
            rv = rtpkcs11.C_GetSlotList(Pkcs11Constants.CK_TRUE, null, slotsCount);
            if (!Pkcs11Constants.equalsPkcsRV(Pkcs11Constants.CKR_OK, rv)) throw new Exception
("C_GetSlotList failed");


            if (slotsCount.getValue().intValue() == 0) {
                throw new Exception("No Rutoken is available!");
            }

            NativeLong[] pSlotList = new NativeLong[slotsCount.getValue().intValue()];
            rv = rtpkcs11.C_GetSlotList(Pkcs11Constants.CK_TRUE, pSlotList, slotsCount);
            if (!Pkcs11Constants.equalsPkcsRV(Pkcs11Constants.CKR_OK, rv)) throw new Exception
("C_GetSlotList failed");

            System.out.println("Token initialization");
            rv = rtpkcs11.C_InitToken(pSlotList[0], DEFAULT_SO_PIN, new NativeLong(DEFAULT_SO_PIN.
length), TOKEN_LABEL);
            if (!Pkcs11Constants.equalsPkcsRV(Pkcs11Constants.CKR_OK, rv)) throw new Exception
("C_GetSlotList failed");

            System.out.println("Opening session");
            NativeLongByReference phSession = new NativeLongByReference();
            rv = rtpkcs11.C_OpenSession(pSlotList[0],
                    new NativeLong(Pkcs11Constants.CKF_SERIAL_SESSION | Pkcs11Constants.CKF_RW_SESSION),
                    null, null, phSession);
            if (!Pkcs11Constants.equalsPkcsRV(Pkcs11Constants.CKR_OK, rv)) throw new Exception
("C_GetSlotList failed");

            hSession = phSession.getValue();

            System.out.println("Logging in as administrator");
            rv = rtpkcs11.C_Login(hSession, new NativeLong(Pkcs11Constants.CKU_SO), DEFAULT_SO_PIN, new
NativeLong(DEFAULT_SO_PIN.length));
            if (!Pkcs11Constants.equalsPkcsRV(Pkcs11Constants.CKR_OK, rv)) throw new Exception
("C_GetSlotList failed");

            System.out.println("User PIN initialization");
            rv = rtpkcs11.C_InitPIN(hSession, DEFAULT_USER_PIN, new NativeLong(DEFAULT_USER_PIN.length));
            if (!Pkcs11Constants.equalsPkcsRV(Pkcs11Constants.CKR_OK, rv)) throw new Exception
("C_GetSlotList failed");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        } finally {
            if (rtpkcs11 != null) {
                System.out.println("Logging out");
                NativeLong rv = rtpkcs11.C_Logout(hSession);

                System.out.println("Closing session");
                rv = rtpkcs11.C_CloseSession(hSession);

                System.out.println("Finalizing PKCS11 library");
                rv = rtpkcs11.C_Finalize(null);

                System.out.println("Test has been completed.");
            }
        }
    }
}
```

Больше примеров по работе с библиотекой **pkcs11jna** можно найти в Рутокен SDK в директории *sdk/java/samples*.