MoaWallet Core API for JavaScript 개발자 가이드

(JavaScript 기반 개발을 위한 시뮬레이터 버전)



|  |  |
| --- | --- |
| 개발부서 | 융합보안연구소 |
| 개발자 | 강현석 |
| 버전(종결 유무) | 0.0.3 |
| 소유자 | ㈜모아플래닛 |

**문서 수정 히스토리**

|  |  |  |
| --- | --- | --- |
| **버전** | **수정내역** | **날짜(수정자)** |
| 0.0.1 | Moa Wallet Core API 개발자 가이드 문서 최초 작성  (시뮬레이터 코드) | 2019.02.13  (이주화) |
| 0.0.2 | Moa Wallet Core API 개발자 가이드 업데이트 | 2019.05.06 (강현석) |
| 0.0.3 | 복원형 지갑 생성 및 복원 로직 업데이트 | 2019.07.01 (강현석) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**1. 개발 환경 및 적용 대상**

1.1 개발 도구

- 개발 에디터 도구 : Visual Studio Code / version : 1.33.1

- 참조 암호 모듈명 및 버전 :**moaCryptoModule-v1.0.x.js**

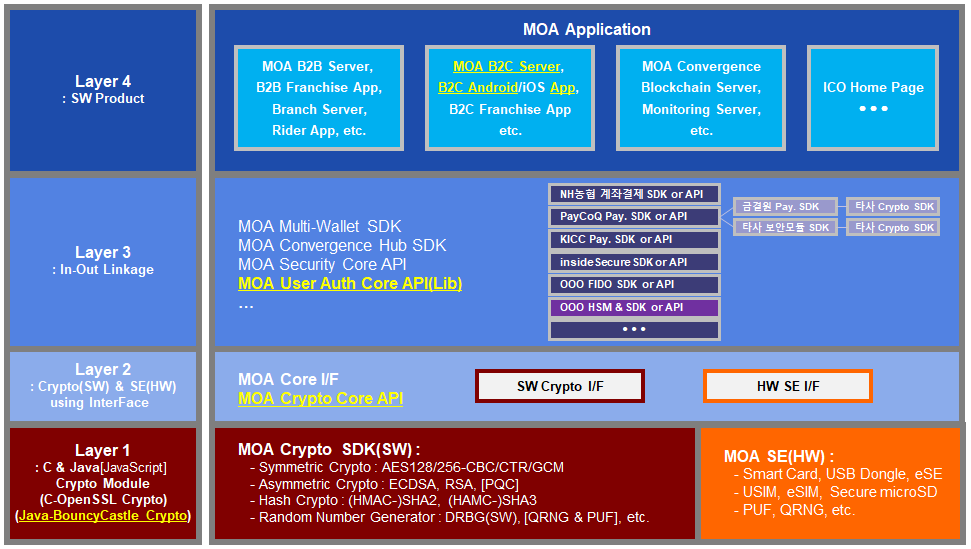
- 참조 전자서명 모듈명 및 버전 :**moaECDSACoreAPI-v1.0.x.js**

- 개발 OS : Windows 10 / Chrome Browser

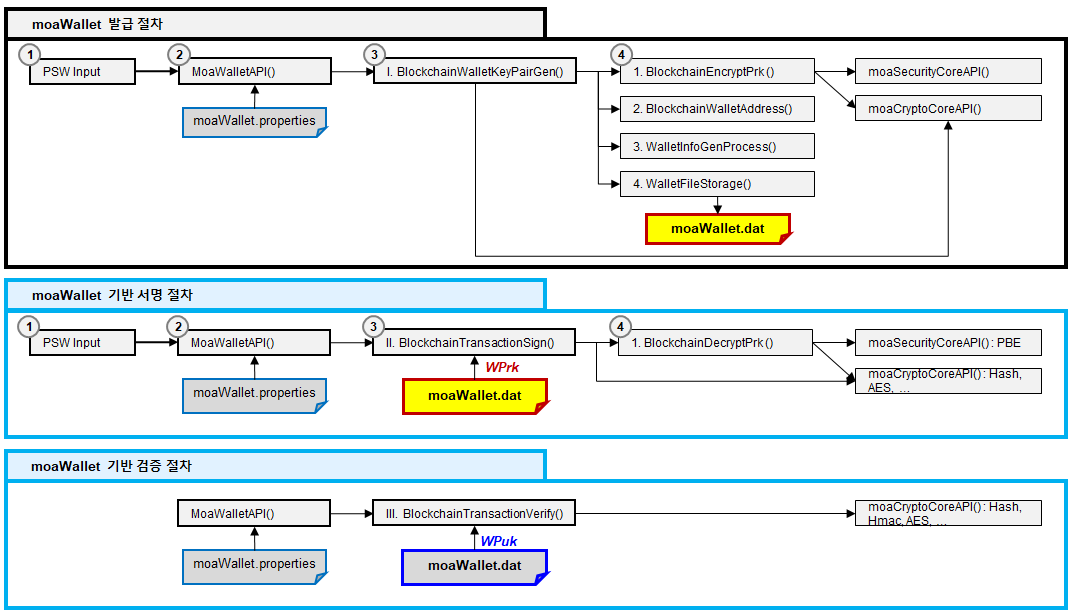
1.2 모아 월릿 모듈 핵심 API : 메소드 설명은 소스코드 참고

- moaBCWalletLib-v1.0.x.js : MoaWallet 키 발급, Wallet Addr 생성, 거래 데이터에 대한 서명 및 검증 등

1.3 모아 개발 스택 계층 구조(Layer3)



1.4 모아 간편 사용자 인증모듈 API 호출 구조



1.5 모듈 구성 및 샘플코드 간략 설명서

|  |  |  |
| --- | --- | --- |
| **Manager & Franchise 인증 핵심 API 및 샘플 코드** | | |
| **Web Brower**  **(Client)** | 🗁**moaCryptoJS** | 자바스크립트 암호모듈 디렉토리 |
| 🗏**moaCryptoModule-v1.0.0.min.js** | 자바스크립트 암호모듈 |
| 🗁**moaSecurityJS** | 인증모듈 디렉토리 |
| 🗏**moaBCWalletLib-v1.0.1.min.js** | 자바스크립트 공통 핵심 Wallet 모듈 |
| 🗏NonRestoreWalletTest.html | 웹브라우저 비복원형 지갑 샘플코드 |
| 🗏RestoreWalletTest.html | 웹브라우저 복원형 지갑 샘플코드 |
| 🗏NonAndRestoreWalletTest.html | 웹브라우저 비복원형 및 복원형 지갑 샘플코드 |

**2. Moa Wallet API 활용 가이드**

**2.1 moaBCWalletLib API 설명**

|  |  |
| --- | --- |
| **파일명** | **설명** |
| moaBCWalletLib | createNonRestoreWallet : password |
| * 기능   + 비복원형 지갑 데이터 생성 * Parameter   + String password     - 개인키 암호화 및 MAC 데이터 생성 시 필요한 패스워드 정보 |
| generateRestoreDataFormat: password |
| * 기능   + 복원형 지갑 데이터 생성을 위한 메시지 생성 * Parameter   + String password     - 개인키 암호화 시 필요한 패스워드 정보 * Return (String)   + 복원형 지갑 데이터 생성을 위한 메시지   + HMAC[E(Puk)] % E(Prk) $ E(Puk) $ Salt |
| createRestoreWallet: (password, data) |
| * 기능   + 복원형 지갑 데이터 생성 * Parameter   + String password     - 개인키 암호화 및 MAC 데이터 생성 시 필요한 패스워드 정보   + Object data     - 복원형 지갑을 생성하기 위한 메시지 정보     - Prk $ Puk $ Salt |
| getPublicKey |
| * 기능   + 공개키 조회 * Return (String)   + 공개키 정보 |
| getWalletAddr |
| * 기능   + 지갑 주소 조회 * Return (String)   + 지갑 주소 정보 |
| moaBCWalletLib | generateTransactionSign: (transaction, password) |
| * 기능   + 트렌젝션 서명 값 생성 및 조회 * Parameter   + String transaction     - 서명 값을 생성할 트렌젝션 정보   + String password     - 서명 생성 시 사용할 암호화된 개인키를 복호화 하기 위한  패스워드 정보 * Return (String)   + 생성된 서명 값 |
| verifyTransactionSign: (transaction, sign, puk) |
| * 기능   + 트렌젝션 서명 값 검증 결과 조회 * Parameter   + String transaction     - 서명 시 사용된 트렌젝션 정보   + String sign     - 트렌젝션 서명 값   + String puk     - 서명을 검증하기 위해 사용될 공개키 * Return(boolean)   + 서명 검증 여부 |
| setType: type |
| * 기능   + 비복원형 또는 복원형 타입 설정   + 비복원형 지갑과 복원형 지갑 모두 소유한 경우에만 해당 함수 사용 * Parameter   + String type     - 비복원형 또는 복원형 타입 설정 값       * TYPE\_RESTORE (복원형)       * TYPE\_NORMAL (비복원형) |

|  |  |
| --- | --- |
| moaBCWalletLib | convertRestoreDataFormat: msg |
| * 기능   + 서버에서 전달받은 Restore Data를 지갑 복원 시  필요한 메시지에 맞게 변환 * Parameter   + String msg     - 복원형 메시지 (지갑 복원 가능 여부 $ 복원형 메시지) * Return(String) * 전자지갑 복원이 가능한 메시지 [E(Prk) $ E(Puk) $ Salt] |
| getHmacPsw: psw |
| * 기능   + 전자지갑 비밀번호를 hmac으로 변환 * Parameter   + String psw     - 전자지갑 패스워드 원문 * Return(String)   + Hmac 처리 된 패스워드 |
| getEncryptedHmacPsw: (id, psw, dateOfBirth) |
| * 기능   + 생년월일을 기반으로 암호화된 hmac 데이터 생성 * Parameter   + String id     - 전자지갑을 보유한 아이디   + String psw     - 전자지갑 패스워드 원문   + String dateOfBirth     - 생년월일 * Return(String)   + 암호화된 Hmac 데이터 |

|  |  |
| --- | --- |
| moaBCWalletLib | verifyPsw: (psw, msg) |
| * 기능   + 전자지갑 복원 시 사용한 패스워드 검증 * Parameter   + String psw     - 전자지갑 패스워드 원문   + String msg     - 서버에서 전달받은 지갑 복원 관련 메시지  [HMAC[E(Puk)] % E(Prk) $ E(Puk) $ Salt] * Return(boolean)   + 패스워드 검증 여부 |

**2.2 Wallet 키쌍 & 주소 생성, 거래에 대한 서명 및 검증 테스트 설명서**

|  |  |  |
| --- | --- | --- |
| **구분** | **설명** | **비고** |
| 용도 | 사용자로부터 PSW를 입력 받아 moaWallet 용의 키 쌍(WPrk, WPuk)과 WPuk 기준으로 Wallet 주소를 생성한 후 WPrk를 2중 암호화하여 Local Storage에 저장하는 샘플 | 비복원 |
| 테스트  파일명 | NonRestoreWalletTest.html |  |
| 사용법/  절차 | <script type="text/javascript" src="./moaSecurityJS/moaBCWalletLib-v1.0.0.min.js"></script>  <script>  // 1.1 사용자가 패스워드 입력  let psw = "zxcv%$#@!123456";  // 1.2 패스워드 기반 비복원형 지갑 생성  MoaWallet.createNonRestoreWallet(psw);  // 1.3 지갑 주소 조회  const walletAddr = MoaWallet.getWalletAddr();  document.write("Wallet Addr [HexString] : " + walletAddr + "<br>");  psw = ""; // 패스워드 초기화  document.write("Deleted Password : " + psw + "<br>");  // 2.1 개인키를 복호화하기 위한 패스워드 입력  let pswForSign = "zxcv%$#@!123456";  // 2.2 트렌젝션 서명 생성  const transaction = "It is 10000$ payment Alice to Bob";  const sign = MoaWallet.generateTransactionSign(transaction, pswForSign);  document.write("Transaction Sign [HexString : " + sign.length + " ] = " + sign.toString() + "<br>");  pswForSign = ""; // 패스워드 초기화  document.write("Deleted Password : " + pswForSign + "<br>");  // 2.3 공개키 조회  const puk = MoaWallet.getPublicKey();  // 2.4 서명된 트렌젝션 검증  const isVerify = MoaWallet.verifyTransactionSign(transaction, sign, puk);  document.write("Transaction Verify Success or Fail : " + isVerify); |  |

|  |  |  |
| --- | --- | --- |
| 테스트  결과 | Wallet Addr [HexString] : f88642bfe142d82ca3a24715aeb274ecaa03e0e9  Deleted Password :  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~[moaWallet KeyPair Load, Transaction Sign & Verify]~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  Transaction Sign [HexString : 142 ] = 3045022100af5dc5bcb079c96144caa9b216afe1285ef836cfc5aa23768413a41cf0f65ca302207d51a842914c313bd99d726d935278157414e703f0d3640343402dfceccddbf9  Deleted Password :  Transaction Verify Success or Fail : true |  |

|  |  |  |
| --- | --- | --- |
| **구분** | **설명** | **비고** |
| 용도 | 사용자로부터 PSW를 입력 받아 메시지(Prk, Puk, Salt 조합)를 생성하고, 생성된 메시지로 moaWallet 용의 키 쌍(WPrk, WPuk)과 WPuk 기준으로 Wallet 주소를 생성한 후 WPrk를 2중 암호화하여 Local Storage에 저장하는 샘플 | 복원 |
| 테스트  파일명 | RestoreWalletTest.html |  |
| 사용법/  절차 | <script type="text/javascript" src="./moaSecurityJS/moaBCWalletLib-v1.0.0.min.js"></script>  <script>  // 1.1 사용자가 패스워드 입력  let psw = MoaWallet.getHmacPsw("zxcv%$#@!123456");  // 1.2 패스워드 기반 복원형 지갑 메시지 생성  const restoreData = MoaWallet.generateRestoreDataFormat(psw);  const splitMsg = restoreData.split('%')  document.write("Data for AuthServer DB Storage : " + splitMsg[1] + "<br>");  // 1.2 생성된 메시지를 기반으로 복원형 지갑 생성  MoaWallet.createRestoreWallet(psw, splitMsg[1]);  // 1.3 지갑 주소 조회  const walletAddr = MoaWallet.getWalletAddr();  document.write("Wallet Addr [HexString] : " + walletAddr + "<br>");  psw = ""; // 패스워드 초기화  document.write("Deleted Password : " + psw + "<br>");  // 2.1 개인키를 복호화하기 위한 패스워드 입력  let pswForSign = MoaWallet.getHmacPsw("zxcv%$#@!123456");  // 2.2 트렌젝션 서명 생성  const transaction = "It is 10000$ payment Alice to Bob";  const sign = MoaWallet.generateTransactionSign(transaction, pswForSign);  document.write("Transaction Sign [HexString : " + sign.length + " ] = " + sign.toString() + "<br>");  pswForSign = ""; // 패스워드 초기화  document.write("Deleted Password : " + pswForSign + "<br>");  // 2.3 공개키 조회  const puk = MoaWallet.getPublicKey();  // 2.4 서명된 트렌젝션 검증  const isVerify = MoaWallet.verifyTransactionSign(transaction, sign, puk);  document.write("Transaction Verify Success or Fail : " + isVerify); |  |

|  |  |  |
| --- | --- | --- |
| 테스트  결과 | Data for AuthServer DB Storage : SsSZITC4btInQq1CO1YMly87swa58SMy13XlUtSUGiw=$BsmTxTHlFQkDweWXjRkaJP3bO4n679WsOpa82wW6HXgvvFUHsI+mT20L9bL7yIjG7O957TD9XI5YKhI2jExbiTU=$tL7zw8Ug84FLKwRMqqjxy0/FwI6+G7cxK5lZcKILlyU=  Wallet Addr [HexString] : f88642bfe142d82ca3a24715aeb274ecaa03e0e9  Deleted Password :  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~[moaWallet KeyPair Load, Transaction Sign & Verify]~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  Transaction Sign [HexString : 142 ] = 304502201cc5d92d6ffcb4f54e593760583659ac314322ebc7120db78bfe818c00563233022100a30cf62edd37becffdaaae915e9409a1301da7a3837f15e9a0974209a4bcb86e  Deleted Password :  Transaction Verify Success or Fail : true |  |

|  |  |  |
| --- | --- | --- |
| **구분** | **설명** | **비고** |
| 용도 | 비복원형 지갑과 복원형 지갑 두 가지를 전부 사용하는 경우, 지갑의 타입을 설정하여 해당 타입에 대한 지갑 데이터를 얻어오는 샘플 | 복원 |
| 테스트  파일명 | NonAndRestoreWalletTest.html |  |
| 사용법/  절차 | <script type="text/javascript" src="./moaSecurityJS/moaBCWalletLib-v1.0.0.min.js"></script>  <script>  // 1.1 지갑 타입 설정 (복원형)  MoaWallet.setType(MoaWallet.TYPE\_RESTORE);  document.write("Set Mode : " + MoaWallet.TYPE\_RESTORE + "<br>");  document.write("TEST [Wallet PUK : " + MoaWallet.getPublicKey() + "]<br>");  // 2.1 개인키를 복호화하기 위한 패스워드 입력  let pswForSign = MoaWallet.getHmacPsw("zxcv%$#@!123456");  // 2.2 복원형 트렌젝션 서명 생성  let transaction = "It is 10000$ payment Alice to Bob";  let sign = MoaWallet.generateTransactionSign(transaction, pswForSign);  document.write("Transaction Sign [HexString : " + sign.length + " ] = " + sign.toString() + "<br>");  pswForSign = ""; // 패스워드 초기화  document.write("Deleted Password : " + pswForSign + "<br>");  // 2.3 복원형 공개키 조회  let puk = MoaWallet.getPublicKey();  // 2.4 복원형 트렌젝션 서명 검증  let isVerify = MoaWallet.verifyTransactionSign(transaction, sign, puk);  document.write("Transaction Verify Success or Fail : " + isVerify + "<br>");  // 3.1 지갑 타입 설정 (비복원형)  MoaWallet.setType(MoaWallet.TYPE\_NORMAL);  document.write("Set Mode : " + MoaWallet.TYPE\_NORMAL + "<br>");  document.write("TEST [Wallet PUK : " + MoaWallet.getPublicKey() + "]<br>");  // 4.1 개인키를 복호화하기 위한 패스워드 입력  pswForSign = "zxcv%$#@!123456";  // 4.2 비복원형 트렌젝션 서명 생성  transaction = "It is 10000$ payment Alice to Bob";  sign = MoaWallet.generateTransactionSign(transaction, pswForSign);  document.write("Transaction Sign [HexString : " + sign.length + " ] = " + sign.toString() + "<br>");  pswForSign = ""; // 패스워드 초기화  document.write("Deleted Password : " + pswForSign + "<br>");  // 4.3 비복원형 공개키 조회  puk = MoaWallet.getPublicKey();  // 4.4 비복원형 트렌젝션 서명 검증  isVerify = MoaWallet.verifyTransactionSign(transaction, sign, puk);  document.write("Transaction Verify Success or Fail : " + isVerify); |  |

|  |  |  |
| --- | --- | --- |
| 테스트  결과 | Set Mode : R  TEST [Wallet PUK : 04eb9d0f73fc5594b2d965b592767fcea848da990ebb1116d67815e455dc090e4998fc50786ce7550853d5777a715ba91064c5eb212f73ae969a1190b6716f004c]  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~[moaRestoreWallet KeyPair Load, Transaction Sign & Verify]~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  Transaction Sign [HexString : 140 ] = 30440220067a468ff8a31cb10bf672b9ab371da406372024a7be1b3588edd9f2bcec5ce602205efdceabbc53ede2bbc10d356908b3d59eb86295cd7ea9c3ff18ff9c5249c9d3  Deleted Password :  Transaction Verify Success or Fail : true  Set Mode :  TEST [Wallet PUK : 0463cd33ec1743789958ff8438879420dce54a17d4ef60b9e26f80a63557752d50a0b1cfaa3ea1f5bc8cdf596ecc3acea7e44986d9c931bd21cd593d443255126d]  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~[moaNonRestoreWallet KeyPair Load, Transaction Sign & Verify]~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  Transaction Sign [HexString : 140 ] = 3044022023c3772478849fcf605ee6781dbfca636ee4a28f2ea2b3fa1354f08a0f0666f202200d7667749b689b6791bd4e744a1fa41a45830ea51f4f80f54e22fd45e59e05d6  Deleted Password :  Transaction Verify Success or Fail : true |  |