

SealBlock 钱包 API 文档

更新日期：2018 年 9 月 20 日

SealBlock 硬件热钱包将钱包私钥存储在 SealBlock 硬件中，使用英特尔 SGX 创造完全加密的可信执行环境进行保护，加上安全规则如多签等达到最高安全强度以保护用户数字资产。SealBlock 钱包 API 是一套标准化的 API 接口，可用于调用 SealBlock 硬件热钱包上的功能如创建钱包、转账等。

APP 钱包集成 SealBlock API

需要开发的代码分为两个部分，APP 端和 APP 服务器端。在 APP 端需要生成一个私钥（称为控制私钥），该控制私钥对应一个地址。该控制私钥和地址将用于控制硬件钱包转账。

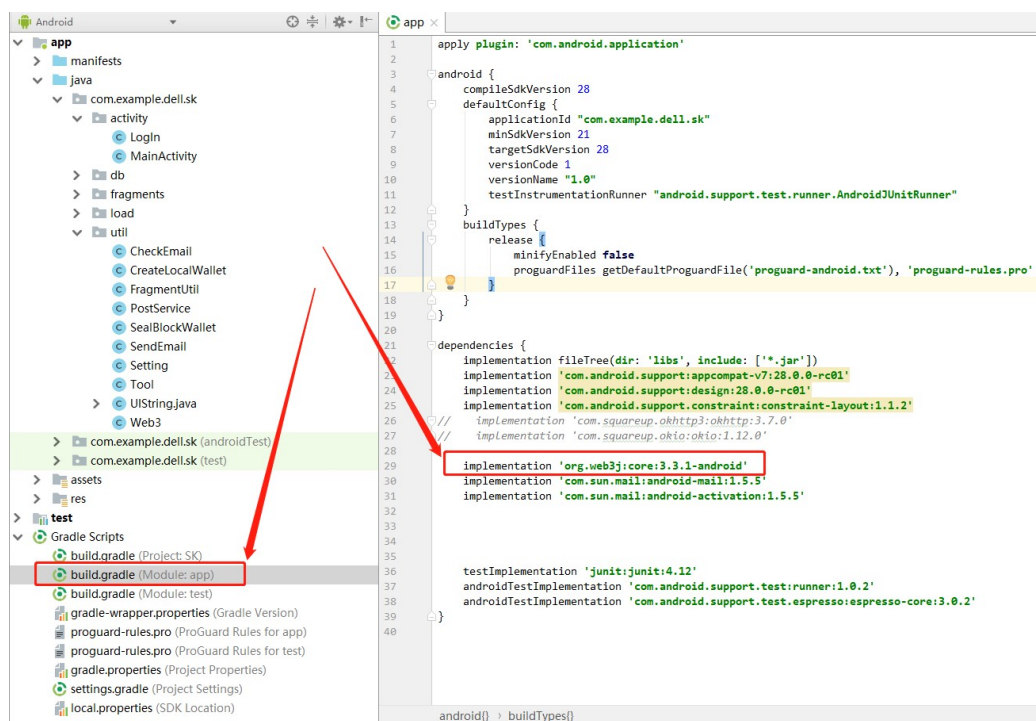
以下实例代码，APP 端为 Java，APP 服务器端通过 zerorpc 与 sealblock 硬件钱包服务器通信。

a. 创建钱包

整体流程：APP 端创建 SealBlockWallet object，以后对钱包的转账通过该 object 进行。服务器端对硬件钱包 APP 发起钱包创建请求，SealBlock 硬件创建钱包并返回钱包地址。

APP (Java):

在 build.gradle 中添加如下依赖库：implementation 'org.web3j:core:3.3.1-android'，见示意图



然后创建 org.sealblock.wallet package，导入 SealBlockWallet.java 文件，该文件定义 SealBlockWallet 类，创建钱包和转账操作必须使用该类，其它操作如查询余额、查询交易状态等 APP 可单独跟服务器通信获取。

调用样例代码如下：

```
import org.sealblock.wallet;

SealBlockWallet wallet = new SealBlockWallet(this); //this is a Context object

wallet.loadWallet("account1");
```

```
String approver_address = wallet.getApproverAddress();

//todo: send approver_address to server, from server call create_wallet and return walletAddr

JSONObject result = wallet.ApproveTransaction(txType,walletAddr,toAddr,amount);

//todo: send result.getString("signed_approval") to server, from server call send_transaction
```

服务器:

创建 rpc 链接

```
var zerorpc = require("zerorpc");
var client = new zerorpc.Client();
var rpcEndPoint = 'tcp://192.168.128.80:4242'; //sealblock_master IP and port
client.on("error", function(error) {
    console.error("RPC client error:", error);
});
```

创建和查询钱包地址

```
var approverAddr = 'c7fb120d27c37b6230893851df77d5edbf24cdea'
var coinType = 'ETH'
client.invoke("create_wallet", approverAddr, coinType, function(error, res, more) {
    if(error) {
        console.error(error);
    } else {
        //res is wallet address
        console.log("create_wallet:", res);
    }
});

var approverAddr = 'c7fb120d27c37b6230893851df77d5edbf24cdea'
client.invoke("list_wallet", approverAddr, function(error, res, more) {
    if(error) {
        console.error(error);
    } else {
        //result is json array which lists all wallets created for the approverAddr
        console.log("list_wallet:", res);
    }
});
```

转账, 查询余额, 查询交易状态

```
var signed_approval = "xxxxxxxxxxxxx"; //obtain signed_approval from APP
client.invoke("send_transaction", "ETH", signed_approval, function(error, res, more) {
    if(error) {
        console.error(error);
    } else {
        //res is transactionId
        console.log("get_transaction:", res);
    }
});

var walletAddr = '4a071eee72bc8664c81b62836932ed0d246da82b';
client.invoke("get_balance", walletAddr, 'ETH', function(error, res, more) {
    if(error) {
        console.error(error);
    } else {
        console.log("get_balance:", res);
    }
});
```

```
var transactionId = '0xf9a7609aa5c45af40ae5d401363c4aa3f27aba75b5babacc6a42a9b4d52d77d0';
client.invoke("get_transaction", transactionId, coinType, function(error, res, more) {
  if(error) {
    console.error(error);
  } else {
    //res is transaction object
    console.log("get_transaction:", res);
  }
});
```