

# 调用合约API

## 源码：

```
export async function execute(params:Execute):Promise<any>{
  try{
    // @ts-ignore
    let executeRes = await window.wallet.features['standard:execute'].execute(params);
    console.log(executeRes);
    return JSON.parse(executeRes);
  }catch(e){}
}
```

## 参数：

```
export interface Execute{
  programID: string;           // leo程序名称
  functionName: string;       // 需要调用的方法名称
  inputs: string;              // 方法输入
  fee: number;                 // 调用合约消耗fee费用
}
```

## 示例

```
const param = {
  programID: "powervoting.leo",
  functionName: "vote",
  inputs: "1u64 1field 2field
{owner:aleo1l44dfmwcu7j2e26yhr1hrxla4lsrmr0rxymxfzj6h8m2mnegyqs8x0end,gates:0u64,power:1u64}",
  fee: 1000,
}

execute(param).then(res => {
  console.log(res)
}).catch(err => {
  console.log(err)
})
```

解释：调用powervoting.leo合约的vote方法，传入“1u64 1field 2field

{owner:aleo1l44dfmwcu7j2e26yhr1hrxla4lsrmr0rxymxfzj6h8m2mnegyqs8x0end,gates:0u64,power:1u64}”参数，fee为1000

# 签名API

## 源码

```
export async function sign(params: Sign): Promise<any> {
  try {
    // @ts-ignore
    let signResult = await window.wallet.features['standard:sign'].sign(params);
    console.log(signResult);
    return JSON.parse(signResult);
  } catch (e) {
    return "";
  }
}
```

## 参数

```
export interface Sign{
  message:string;      // 签名消息
}
```

# 转账API

## 源码

```
export async function transfer(params: Transfer): Promise<any> {
  try {
    // @ts-ignore
    let transferRes = await window.wallet.features['standard:transfer'].transfer(params);
    return transferRes;
  } catch (e) {
    return "";
  }
};
```

## 参数

```
export interface Transfer {
  to: string;      // 转账目标账户
  amount: string;  // 转账金额
  record: string;  // record
}
```

## 解密record API

### 源码

```
export async function decryptRecord(record: string): Promise<DecryptRecordData[]> {
  try {
    let records = [] as string[]
    records.unshift(record)
    // @ts-ignore
    let recordData = await window.wallet.features['standard:decrypt'].decrypt(records)
    return recordData.result
  } catch (e) {
    return [] as DecryptRecordData[]
  }
}
```

### 参数

record: 加密的record

## 连接钱包API

### 源码

```
export async function connectWalletPlugin(): Promise<ConnectData> {
  // @ts-ignore
  let account = await window.wallet.features['standard:connect'].connect()
  return account
}
```

## 判断钱包是否连接API

### 源码

```
export async function walletConnected() {
  // @ts-ignore
  let connected = window.wallet.connected
  return connected
}
```

# 获取钱包账户API

## 源码

```
export async function walletAccount(): Promise<AccountData> {  
  // @ts-ignore  
  let accounts = window.wallet.accounts  
  return accounts  
}
```