

合约部署

像上一次热身报告时将写好的合约部署在自己的私链上

```
> miner.start();admin.sleepBlocks(1);miner.stop();
null
> Contract mined! Address: 0x7990f026041b01dbc8c402312ddf96a65ac3a9e3
[object Object]
> ci.address
"0x7990f026041b01dbc8c402312ddf96a65ac3a9e3"
```

合约数据结构

```
struct info {
    string nickname;
    string key;
    bool ableToGetInfo;
    string email;
    string phone;
}

mapping (address => bool) private exist;
mapping (address => info) private infoMap;
mapping (address => string) private certificationMap;
```

exist 记录账户是否使用该合约

infoMap 记录账户存储的信息

certificationMap 记录账户的密文内容

接口解释

```
function setExist(bool _b) public {
    exist[msg.sender] = _b;
}
```

设置是否使用合约功能, true 为使用, false 为不使用

```
function userExisted(address user) public view returns (bool) {
    return exist[user];
}
```

获取一个账户是否使用该合约

```
> ci.setExist.sendTransaction(true, {from:ac1})
"0x2c8ca16bcfdc00574ef7b071a9d7331589b1e9bcbf8efb7a8858a4e22733030f"
> miner.start();admin.sleepBlocks(1);miner.stop();
null
> ci.userExisted(ac1)
true
```

```
function setInfo(string _nickname, string _key) public {
    if (exist[msg.sender] == false) {
        return;
    }
    infoMap[msg.sender].nickname = _nickname;
    infoMap[msg.sender].key = _key;
}
```

设置账户的昵称与公钥

```
function getInfo(address user) public view returns (string, string) {
    if (exist[user] == false) {
        return ("", "");
    }
    return (infoMap[user].nickname, infoMap[user].key);
}
```

获取一个账户的昵称与公钥信息

```
> ci.setInfo.sendTransaction("mike", "key",{from:ac1})
"0xc167c7f2a096cd56c76d8dc50379b20d19485dadb28e06d3ed8243cbdedad521"
> miner.start();admin.sleepBlocks(1);miner.stop();
null
> ci.getInfo(ac1)
["mike", "key"]
```

```
function setExtraInfo(string _email, string _phone) public returns (bool) {
    if (exist[msg.sender] == false) {
        return false;
    }
    infoMap[msg.sender].email = _email;
    infoMap[msg.sender].phone = _phone;
    return true;
}
```

设置账户额外信息

```
function getExtraInfo(address user) public view returns (string, string) {
    if (infoMap[user].ableToGetInfo == false) {
        return ("", "");
    }
    return (infoMap[user].email, infoMap[user].phone);
}
```

获取一个账户的额外信息

```
> ci.setExtraInfo.sendTransaction("test@163.com", "10010",{from:ac1})
"0x03f82c3219e90cb0bfff01003c313b07e0b22dbd8551604f246487d12537e77a9"
> miner.start();admin.sleepBlocks(1);miner.stop();
null
> ci.getExtraInfo(ac1)
["test@163.com", "10010"]
```

```
function setGettable(bool _able) public returns (bool) {
    if (exist[msg.sender] == false) {
        return false;
    }
    infoMap[msg.sender].ableToGetInfo = _able;
    return true;
}
```

设置是否运行获取账户信息

```
function setCertification(string text) public returns (bool) {
    if (exist[msg.sender] == false) {
        return false;
    }
    certificationMap[msg.sender] = text;
    return true;
}
```

设置账户密文

```
function getCertificate(address user) public view returns (string) {
    if (exist[user] == false) {
        return ("");
    }
    return (certificationMap[user]);
}
```

获取一个账户的密文, 可用公钥信息来解密密文

```
> ci.setCertification.sendTransaction("text",{from:ac1})
"0x8e94499269ac85e059d24db3ebbd84fbd9faea253686eea20fdf84161c74cefb"
> miner.start();admin.sleepBlocks(1);miner.stop();
null
> ci.getCertificate(ac1)
"text"
```