

# DAPP智能合约模式系统开发（搭建）定制源码

产品名称	DAPP智能合约模式系统开发（搭建）定制源码
公司名称	深圳漫云网络科技有限公司
价格	.00/件
规格参数	
公司地址	深圳市南山区粤海街道麻岭社区科研路9号比克科技大厦1701D
联系电话	18638161680 18638161680

## 产品详情

DAPP是在底层\*\*\*平台衍生的各种分布式应用，\*\*-181系259I统-3365是\*\*\*世界中的服务提供形式。它被认为是开启\*\*\*3.0时代的标志。

DAPP之于\*\*\*，就像APP之于Android和iOS，都是应用程序和底层环境之间的关系。目前主流的DAPP\*\*\*有三家，分别是：ETH(以太坊)、EOS(柚子)、TRON(波场)。这些就跟Android系统分不同的ROM类似，比如华为的EMUI、小米的MIUI、OPPO的ColorOS等等。

总之，DAPP就是\*\*\*系统上的应用程序，我们使用起来和普通APP并不会有太大的差别。

### 1.1.2例子

Steemit是一个内容社交DAPP，作为基于Steem公链\*\*的DAPP，在业内拥有极高的人气。用户可以在上面分享，文章，视频，图片，绘画作品，摄影作品等等。上传完作品之后，通过用户投票，决定上传质量的高低，上传者从而获得相应的代币。

跟Instagram、Twitter其他内容平台相比，Steemit上的用户能通过自己的贡献而获得相应的回报，能够有效地激励用户参与内容建设。

DAPP是什么意思？总的来说，DAPP就是去中心化应用，DAPP在继承传统APP的优势之后，再结合\*\*\*的特点而形成的，DAPP比较像是众创模式、共享模式、去中心化模式。

编写智能合约：

```
pragmasolidity^0.5.0;
```

```
import"truffle/Assert.sol";//引入的断言
```

```
import"truffle/DeployedAddresses.sol";//用来获取被测试合约的地址
```

```

import "../contracts/Adoption.sol"; // 被用来测试的合约

contract TestAdoption {

    Adoption adoption = Adoption(DeployedAddresses.Adoption());

    // 领养测试用例

    function testUserCanAdoptPet() public {

        uint returnedId = adoption.adopt(8);

        uint expected = 8;

        Assert.equal(returnedId, expected, "Adoption of pet ID 8 should be recorded.");

    }

    // 宠物所有者测试用例

    function testGetAdopterAddressByPetId() public {

        // 期望领养者的地址就是本合约地址，因为交易是由测试合约发起交易，

        address expected = address(this);

        address adopter = adoption.adopters(8);

        Assert.equal(adopter, expected, "Owner of pet ID 8 should be recorded.");

    }

    // 测试所有领养者

    function testGetAdopterAddressByPetIdInArray() public {

        // 领养者的地址就是本合约地址

        // address expected = this; 因为编译器版本是 0.5.0，代码之前编写是按照 0.4.17 的规范来写的

        address expected = address(this);

        address[16] memory adopters = adoption.getAdopters();

        Assert.equal(adopters[8], expected, "Owner of pet ID 8 should be recorded.");

    }

}

```

DAPP的基本架构

主要分为三个部分：

\*\*\*以及本身提供的能力

三方钱包等提供的能力

我们要写的DAPP部分----分为三个部分：

和EVM交互的合约层--solidity编写；

和合约层、钱包交互的交互层-web3js，其和Ajax类似，可以用来读取以太坊\*\*\*，主要使用的JSONRPC和\*\*\*进行通信，对\*\*\*网络进行数据的读写操作；

前端代码业务层-vue、react等编写；