

1. 编写智能合约

```
// 初始化合约地址
constructor() public {
    Owner = msg.sender;
}

// 区块链存储的文件数量
uint public storagesNumber;

// 文件哈希对应的文件存储信息
mapping(bytes => storages) storages;

// 录入文件信息
function setStorages( bytes memory fileHash ) public {
    // 存储文件信息
    storages[ fileHash ] = storages( storagesNumber,fileHash,msg.sender,block.timestamp);
    // 存储数量加一
    storagesNumber++;
}

// 查询文件信息
function queryStorages( bytes memory fileHash ) public view returns(uint,bytes memory,address,uint) {
    storages storage s = storages[ fileHash ];
    return (s.fileID, s.fileHash, s.fileOwner, s.createTime);
}
```

2. 部署智能合约

验证智能合约

初始化参数

```
1 {
2     " Owner " : " msg.sender "
3 }
```

3. 调用智能合约

