

汽车租赁系统

如何运行

1. 在本地启动 ganache 应用并创建测试链。
2. 在 `./contracts` 中安装需要的依赖，运行如下的命令：

```
npm install
```

3. 在 `./contracts` 中编译合约，运行如下的命令：

```
npx hardhat compile
```

4. 在 `./contracts` 中部署合约到 ganache 测试链上，在 `./contracts/hardhat.config.ts` 文件中输入测试链上的账户私钥以导入用户，然后运行如下的命令：

```
npx hardhat run ./scripts/deploy.ts --network ganache
```

将输出的合约部署地址填写到 `./frontend/src/utils/contract-addresses.json` 中

5. 复制 `./contracts/artifacts/contracts/BorrowYourCar.sol/BorrowYourCar.json` 和 `./contracts/artifacts/contracts/QiushiToken.sol/QiushiToken.json` 到 `./frontend/src/utils/abis` 中
6. 在 `./frontend` 中安装需要的依赖，运行如下的命令：

```
npm install
```

7. 在 `./frontend` 中启动前端程序，运行如下的命令：

```
npm run start
```

功能实现分析

1. 查看自己拥有的汽车列表。查看当前还没有被借用的汽车列表。
维护了两个映射：

```
mapping(uint256 => Car) public cars;  
mapping(address => uint256[]) private ownedCars;
```

- 查看拥有的汽车列表时，通过传入当前账户的地址返回拥有的车辆序列：
`ownedCars[msg.sender]`
- 查看当前还没有被借用的汽车列表时，遍历 `cars` 映射集，返回一个映射 `availableCars`，其包含 `cars` 中所有 `borrower` 属性为 `address(0)`，即无人借用的汽车。

2. 查询一辆汽车的主人，以及该汽车当前的借用者（如果有）。

维护上述 `cars` 映射，然后实现以下函数：

```
function getOwner(uint256 carId)
function getBorrower(uint256 carId)
```

传入汽车的 ID，返回 `cars[ID]` 的 `owner` 和 `borrower` 属性。

3. 选择并借用某辆还没有被借用的汽车一定时间。

实现 `borrowCar` 函数：

```
function borrowCar(uint256 carTokenId, uint256 duration) external { // 传入车辆
ID, 借用时长
    require(ownerOf(carTokenId) != msg.sender, "You cannot borrow your own
car"); // 自己的车
    require(cars[carTokenId].borrower == address(0), "Car is already borrowed");
// 已被借用
    uint256 rentalFee = duration / 60; // 一分钟1个自来水币
    require(tapWaterCoin.balanceOf(msg.sender) >= rentalFee, "Insufficient
balance"); // 余额不足
    tapWaterCoin.transferFrom( msg.sender, cars[carTokenId].owner, rentalFee); //
交易
    cars[carTokenId].borrower = msg.sender; // 写入借用者
    cars[carTokenId].borrowUntil = block.timestamp + duration; // 借用到期时间
    emit CarBorrowed(carTokenId, msg.sender, block.timestamp, duration); // 向区
块链发送交易事件
}
```

4. 使用自己发行的积分（ERC20）完成付费租赁汽车的流程。

实现一个基于 ERC20 的合约 `TapWaterCoin`，发行自来水币（某种拧一下水龙头就会到处都是的廉价货币），并通过上述 `BorrowYourCar` 合约中的 `borrowCar` 函数实现交易中的费用支出。

项目运行截图

（若图片不能正常查看请参考 readme.pdf 文件。）

- 进入汽车租赁系统，此时尚未连接账户。

汽车租赁系统

领取自来水币空投

连接钱包

当前用户：无用户连接

当前用户拥有自来水币数量：0

添加新车

我的车辆

空闲车辆

更新空闲车辆列表

ID:

查询车辆

ID:

借用时间 (min) :

借用

- 使用小狐狸连接到钱包

汽车租赁系统

领取自来水币空投

当前用户：0x916c3d14CDBCE867a6bD3522946828b42D74093a

当前用户拥有自来水币数量：0

添加新车

我的车辆

空闲车辆

更新空闲车辆列表

ID:

查询车辆

ID:

借用时间 (min) :

借用

- 在当前账户下添加若干车辆用以测试，刷新页面后显示拥有车辆和空闲车辆

添加新车

我的车辆

ID: 1



ID: 2



ID: 3



空闲车辆

更新空闲车辆列表

ID: 1



ID: 2



ID: 3



- 领取自来水币空投，刷新页面后显示货币数量

汽车租赁系统

领取自来水币空投

当前用户: 0x916c3d14CDBCE867a6bD3522946828b42D74093a

当前用户拥有自来水币数量: 1000000

添加新车

- 查询车辆信息

ID: 1



localhost:3000 显示

ID: 2
车主: 0x916c3d14CDBCE867a6bD3522946828b42D74093a
状态: 空闲
借用者: 无

确定

ID: 3



空闲车辆

更新空闲车辆列表

ID: 1



ID: 2



ID: 3



ID: 2

查询车辆

- 更换账户后, 借用空闲车辆

当前用户: 0x6dE52Fa7129B88d4EDF6b1Cee401638997a49399

当前用户拥有自来水币数量: 1000000


添加新车

我的车辆


空闲车辆

更新空闲车辆列表


ID: 1



ID: 2



ID: 3



ID:


查询车辆

ID:

借用时间 (min):

借用

http://localhost:3000


的支出上限请求
验证第三方详情

网站请求的支出上限 [编辑](#)
0.000000000000000003 TWC

[建议的网站 >](#)
0.00024103
燃料 (估计) 0.00024103 UNKNOWN
很可能在 15 秒以内 最大费用: 0.00024103 UNKNOWN
[查看详情](#)

拒绝

批准

localhost:3000 显示

借用成功。

确定

添加新车

我的车辆

空闲车辆

更新空闲车辆列表

ID: 1



ID: 2



ID: 3



ID:

查询车辆

ID:

2

借用时间 (min) :

3

借用

- 刷新页面，发现拥有的自来水币减少，空闲车辆状态更新

汽车租赁系统

领取自来水币空投

当前用户: 0x6dE52Fa7129B88d4EDF6b1Cee401638997a49399

当前用户拥有自来水币数量: 999997

添加新车

我的车辆

空闲车辆

更新空闲车辆列表

ID: 1



ID: 3



- 查询已被借用的车辆

localhost:3000 显示

ID: 2
车主: 0x916c3d14CDBCE867a6bD3522946828b42D74093a
状态: 已被借用
借用者: 0x6dE52Fa7129B88d4EDF6b1Cee401638997a49399

确定

我的车辆

空闲车辆

更新空闲车辆列表

ID: 1



ID: 3



ID: 2

查询车辆

- 借用已被借用的车辆

localhost:3000 显示

车辆已被借用

确定

添加新车

我的车辆

空闲车辆

更新空闲车辆列表

ID: 1



ID: 3



ID:

查询车辆

ID:

借用时间 (min) :

借用

- 租用时间 (3min) 结束后，更新空闲车辆列表，发现车辆已归还

汽车租赁系统

领取自来水币空投

当前用户: 0x6dE52Fa7129B88d4EDF6b1Cee401638997a49399

当前用户拥有自来水币数量: 999997

添加新车

我的车辆

空闲车辆

更新空闲车辆列表

•

ID: 1



•

ID: 2



•

ID: 3



参考内容

- 课程的参考Demo见: [DEMOS](#)。
- ERC-4907 [参考实现](#)