

## iTaxi - Shared Taxi Business on Blockchain

### 1. Deficiencies:

- At sometimes, Web3 Provider not support some browsers. I have faced some trouble while I executed it on other browser such as Chrome, Opera. I have browsed it but I can not find absolute solution. According to internet, they just offer execute it on Internet Explorer.
- I can send ether from address to contract with payable fallback function, but I can not send ether from contract to address so Contract Balance is not change after GetSalary and GetDividend methods as you see at Run part. I have browsed it and make constructor to payable is offered as a solution. Although it works on Localhost VM, it does not work on Web3 Provider.
- Before each demonstration, you should refresh ganache-cli and smart contract address.

### 2. Configuration:

- To demonstration, firstly we need to install ganache-cli as a Web3 Provider. Below command can be used to installation.  
*npm install -g ganache-cli*
- After install ganache-cli, we can start it with below command. We use `-a` and `-e` parameters to change account number and balance amount we use in demonstration. We generate 130 accounts and each account has 200 ether as default balance.  
*ganache-cli -a 130 -e 200*
- After install and run ganache-cli, we need to deploy our Smart Contract and make change on ABI and Smart Contract address at code. We need to open .sol file at Remix IDE then it should be build with version 0.4.23 as you see in code.
- After build .sol file we deploy Smart Contract and copy Smart Contract's address from Remix IDE to AppContractJSON.js file in js directory. In AppContractJSON.js file we have 2 variables that named are ContractAddress and ABI. ABI does not need to any configuration but ContractAddress need configuration whenever we deploy Smart Contract.

### 3. Run:

- After above configuration steps, we can execute main.html file on Internet Explorer. I have faced some trouble while I executed it on other browser such as Chrome, Opera. I have browsed it but I can not find absolute solution. According to internet, they just offer execute it on Internet Explorer.
- When execute main.html, Info tab welcome. Info tab's screenshot placed below. As you see, we can initiate Manager, CarDealer, and Driver's info but it can change at this tab.
  - Driver Cap : This is monthly salary of Driver and only Manager can alter it.
  - Ticket Price : This is taxi ticket price that is used by customer.

*SetDriver, SetManager, SetCarDealer* is used at here.

## My Smart Contract

Info

Demo

Driver Address

0x67e83e0fa633ebefd4a1ba835ab361cd5505e5fa

Driver Cap

20

Ticket Price

2

Update

	Address
Account	0x4ae6ec7bc66095cbcf2f77230390dde838c559b0
<div>Change</div> Manager	0x4ae6ec7bc66095cbcf2f77230390dde838c559b0
<div>Change</div> Car Dealer	0x3f3f96621a9a1af4b20357f5a63c454372392543
Driver	0x67e83e0fa633ebefd4a1ba835ab361cd5505e5fa

- After all initialization, we can select Demo tab. Demo tab's screenshot placed below. With **Add 100 Participants** button that is yellow is shortcut to easily add 100 participants to contract.  
*Join* method is used at 100 times.

## My Smart Contract

Info

Demo

Add 100 Participants

Start Demo

- Start Demo button is activated after adding 100 participants. It initiates 1 year simulation that is keep all operations you give in project template.

## My Smart Contract

Info
Demo

Add 100 Participants
Start Demo

- A few times later, demonstration is finished and all logs are appended to screen as a table as below. It can takes 7-8 seconds and logs append at finish not simultaneously.

## My Smart Contract

Date	Operation	Amount	Contract Balance
13/1/2019	100 participants joined!	10000 ETH	10000 ETH
13/1/2019	Car Dealer set car propose! CarID : 00083290386130592628289163215554 Price : 2320 Valid Time : 23/1/2019		10000 ETH
26/1/2020	Manager purchased the proposal car!	2320 ETH	7680 ETH
13/1/2019	Customer paid ticket!	2 ETH	7682 ETH
22/1/2019	Customer paid ticket!	2 ETH	7684 ETH
31/1/2019	Customer paid ticket!	2 ETH	7686 ETH
9/2/2019	Customer paid ticket!	2 ETH	7688 ETH
18/2/2019	Manager release salary to Taxi Driver's account!	20 ETH	7688 ETH

- At the beginning of demonstration, Car Dealer sets car propose and Manager purchases the proposal car. Then the demonstration continue with Customer pays.  
*CarPropose, PurchaseCar* are used. Also *GetCharge* is used while customer payment.

13/1/2019	100 participants joined!	10000 ETH	10000 ETH
13/1/2019	Car Dealer set car propose! CarID : 00083290386130592628289163215554 Price : 2320 Valid Time : 23/1/2019		10000 ETH
26/1/2020	Manager purchased the proposal car!	2320 ETH	7680 ETH
13/1/2019	Customer paid ticket!	2 ETH	7682 ETH

- At each month, Manager releases salary to Taxi Driver's account just one time. As you see, Contract Balance is not changed because this amount not send to Taxi Driver's address yet.

*PaySalary* is used for release.

9/2/2019	Customer paid ticket!	2 ETH	7688 ETH
18/2/2019	Manager release salary to Taxi Driver's account!	20 ETH	7688 ETH
18/2/2019	Customer paid ticket!	2 ETH	7690 ETH

- At each six months, Manager pays to Car Dealer for Car Expenses and divides participants' profit to their account just one time. As you see, Contract Balance is changed just at Car Expenses, because amount is not send to participants' address yet. *CarExpenses*, *PayDividend* are used at here.

12/7/2019	Customer paid ticket!	2 ETH	7722 ETH
21/7/2019	Manager paid to Car Dealer for Car Expenses!	10 ETH	7712 ETH
21/7/2019	Manager divide participants' profit to their account!		7712 ETH
21/7/2019	Customer paid ticket!	2 ETH	7714 ETH

- At the end of the year, Car Dealer sets car purchase propose, and more than half of the participants approve purchase proposal and finally Car Dealer send purchase car price. *ApproveSellProposal* method is executed for each participant seperatly. *PurchasePropose*, *ApproveSellProposal*, *SellCar* are used as orderly.

26/1/2020	Car Dealer set car purchase propose! CarID : 00083290386130592628289163215554 Price : 430 Valid Time : 4/2/2020		7754 ETH
26/1/2020	Total 51 participants have approved purchase proposal!		7754 ETH
28/1/2020	Car Dealer send purchase car price!	430 ETH	8184 ETH

- After send the car, Participants get dividend and Taxi Driver get salary from their account to address. At each row, there is a button that named is Show Balance and show participants or taxi driver's balance on modal. *GetSalary, GetDividend* are used.

30/1/2020	All participants transfer money from their account to address!	Show Balance	8184 ETH
30/1/2020	Taxi Driver transfer money from their account to address!	Show Balance	8184 ETH

  

3/12/2019	Cus	2 ETH	7744 ETH
12/12/2019	Cus	2 ETH	7746 ETH
21/12/2019	Cus	2 ETH	7748 ETH
30/12/2019	Cus	2 ETH	7750 ETH
8/1/2020	Man	20 ETH	7750 ETH
8/1/2020	Cus	2 ETH	7752 ETH
17/1/2020	Cus	2 ETH	7754 ETH
26/1/2020	Car		7754 ETH

  

3/12/2019	Cus	2 ETH	7744 ETH
12/12/2019	Cus	2 ETH	7746 ETH
21/12/2019	Cus	2 ETH	7748 ETH
30/12/2019	Cus	2 ETH	7750 ETH
8/1/2020	Man	20 ETH	7750 ETH

In here, there is a problem I can not solve. I can send ether from address to contract with payable fallback function, but I can not send ether from contract to address so Contract Balance is not change after *GetSalary* and *GetDividend* methods as you see. I have browsed it and make constructor to payable is offered as a solution. Although it works on Localhost VM, it does not work on Web3 Provider.