

Parking

Generated by Doxygen 1.8.14

Contents

1	Parking Smart Contract	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Data Structure Documentation	7
4.1	Posto Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Field Documentation	7
4.1.2.1	coordinate	7
4.1.2.2	prezzo	7
4.1.2.3	idSosta	8
4.2	Sosta Struct Reference	8
4.2.1	Detailed Description	8
4.2.2	Field Documentation	8
4.2.2.1	ora	8
4.2.2.2	durata	8
4.2.2.3	targa	9
4.2.2.4	nextSosta	9
4.2.2.5	idPosto	9

5 File Documentation	11
5.1 Parking.c File Reference	11
5.1.1 Detailed Description	12
5.1.2 Function Documentation	12
5.1.2.1 constructor()	12
5.1.2.2 checkDisp()	12
5.1.2.3 pago()	13
5.1.2.4 acquisto()	13
5.1.2.5 prenotazione()	14
5.1.2.6 ricercaPosto()	14
5.1.2.7 estensione()	14
5.1.3 Variable Documentation	15
5.1.3.1 nPosti	15
5.1.3.2 targaProva	15
5.1.3.3 maxTime	15
5.1.3.4 maxDurata	15
5.1.3.5 ether1	15
5.1.3.6 addr	15
5.1.3.7 ownAddr	15
Index	17

Chapter 1

Parking Smart Contract

Parcheggio per il quale il pagamento è effettuato tramite la rete ethereum. I posti sono numerati e sono conosciute le coordinate di ciascun posto, inoltre ogni zona, in cui le strisce verdi sono situate, ha un prezzo che varia in base alla centralità. I servizi che si vogliono offrire sono:

- Ricerca dei posti disponibili in città
- Pagamento, in loco, del parcheggio
- Estensione del periodo di sosta
- Prenotazione in anticipo del parcheggio

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Posto	Struttura del posto	7
Sosta	Struttura della sosta	8

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

[Parking.c](#)

Questo è il contratto per un parcheggio sulla rete Ethereum. Il file definisce tutte le funzioni necessarie al funzionamento

11

Chapter 4

Data Structure Documentation

4.1 Posto Struct Reference

struttura del posto

Data Fields

- string [coordinate](#)
- uint [prezzo](#)
- uint [idSosta](#)

4.1.1 Detailed Description

struttura del posto

4.1.2 Field Documentation

4.1.2.1 coordinate

```
string Posto::coordinate
```

coordinate geografiche

4.1.2.2 prezzo

```
uint Posto::prezzo
```

prezzo in ether

4.1.2.3 idSosta

```
uint Posto::idSosta
```

indice della sosta acquistata più di recente

The documentation for this struct was generated from the following file:

- [Parking.c](#)

4.2 Sosta Struct Reference

struttura della sosta

Data Fields

- uint [ora](#)
- uint [durata](#)
- string [targa](#)
- uint [nextSosta](#)
- uint [idPosto](#)

4.2.1 Detailed Description

struttura della sosta

4.2.2 Field Documentation

4.2.2.1 ora

```
uint Sosta::ora
```

ts dell'ora in cui è comincia la sosta

4.2.2.2 durata

```
uint Sosta::durata
```

durata, in secondi, della sosta

4.2.2.3 targa

```
string Sosta::targa
```

targa della macchina che si vuole parcheggiare

4.2.2.4 nextSosta

```
uint Sosta::nextSosta
```

indice della prossima sosta su quel posto

4.2.2.5 idPosto

```
uint Sosta::idPosto
```

indice del posto

The documentation for this struct was generated from the following file:

- [Parking.c](#)

Chapter 5

File Documentation

5.1 Parking.c File Reference

Questo è il contratto per un parcheggio sulla rete Ethereum. Il file definisce tutte le funzioni necessarie al funzionamento.

Data Structures

- struct [Posto](#)
struttura del posto
- struct [Sosta](#)
struttura della sosta

Functions

- [constructor](#) (address _payAccount) public
Costruttore del contratto, viene chiamato solo al momento del deploy.
- function [checkDisp](#) (uint _tsOwn, uint _dOwn, uint _headSoste)
Controlla la disponibilità di un posto per un certo orario.
- function [pago](#) (uint _prezzo, uint _idSosta)
Permette di pagare la sosta.
- function [acquisto](#) (uint _dOwn, uint _idPosto, string _targa)
Permette di acquistare una sosta in questo momento.
- function [prenotazione](#) (uint _tsOwn, uint _dOwn, uint _idPosto, string _targa)
Permette di acquistare una sosta per un momento futuro.
- function [ricercaPosto](#) (uint _tsOwn, uint _dOwn)
Permette di ricercare tutti i posti disponibili per un particolare orario.
- function [estensione](#) (uint _idSosta, uint _dOwn)
Permette di estendere una sosta già acquistata.

Variables

- uint `nPosti` = 10
- string `targaProva` = "AB000CD"
- uint `maxTime` = 10000000000
- uint `maxDurata` = 86400
- uint `ether1` = 1000000000000000000
- address `addr`
- address `ownAddr`

5.1.1 Detailed Description

Questo è il contratto per un parcheggio sulla rete Ethereum. Il file definisce tutte le funzioni necessarie al funzionamento.

Author

Umberto Coppola Bottazzi, Alfonso Fezza

Version

V1.3

Date

22-Ottobre-2018

5.1.2 Function Documentation

5.1.2.1 constructor()

```
constructor (  
    address _payAccount )
```

Costruttore del contratto, viene chiamato solo al momento del deploy.

Parameters

<code>_payAccount</code>	indirizzo dell'account che riceverà i pagamenti
--------------------------	---

5.1.2.2 checkDisp()

```
function checkDisp (  
    uint _tsOwn,
```



```
uint _dOwn,  
uint _headSoste )
```

Controlla la disponibilità di un posto per un certo orario.

Parameters

<code>_tsOwn</code>	timestamp dell'orario in cui si desidera il posto
<code>_dOwn</code>	durata della sosta
<code>_headSoste</code>	riferimento alla testa della lista delle soste di un particolare posto

Returns

- `_i`: indice della prima sosta che ha il ts maggiore di quello cercato
- `_iPrev`: indice della sosta precedente a quella trovata
- `_disp`: informazione sulla disponibilità del posto. Questo parametro può assumere i seguenti valori:
 - true, se il posto è disponibile
 - false, se il posto non è disponibile

5.1.2.3 pago()

```
function pago (  
    uint _prezzo,  
    uint _idSosta )
```

Permette di pagare la sosta.

Parameters

<code>_prezzo</code>	costo in wei della sosta
<code>_idSosta</code>	id della sosta appena generato, usato come ricevuta di pagamento

Returns

- `_res`: informazione sulla riuscita del pagamento. Questo parametro può assumere i seguenti valori:
 - true, se il pagamento è riuscito
 - false, se il pagamento non è riuscito

5.1.2.4 acquisto()

```
function acquisto (  
    uint _dOwn,  
    uint _idPosto,  
    string _targa )
```

Permette di acquistare una sosta in questo momento.

Parameters

<code>_dOwn</code>	durata della sosta
<code>_idPosto</code>	indice del posto su cui si vuole sostare
<code>_targa</code>	targa della macchina da parcheggiare

5.1.2.5 prenotazione()

```
function prenotazione (
    uint _tsOwn,
    uint _dOwn,
    uint _idPosto,
    string _targa )
```

Permette di acquistare una sosta per un momento futuro.

Parameters

<code>_tsOwn</code>	timestamp dell'orario in cui si desidera il posto
<code>_dOwn</code>	durata della sosta
<code>_idPosto</code>	indice del posto su cui si vuole sostare
<code>_targa</code>	targa della macchina da parcheggiare

5.1.2.6 ricercaPosto()

```
function ricercaPosto (
    uint _tsOwn,
    uint _dOwn )
```

Permette di ricercare tutti i posti disponibili per un particolare orario.

Parameters

<code>_tsOwn</code>	timestamp dell'orario in cui si desidera il posto
<code>_dOwn</code>	durata della sosta

Returns

vettore di posti liberi

5.1.2.7 estensione()

```
function estensione (
    uint _idSosta,
    uint _dOwn )
```

Permette di estendere una sosta già acquistata.

Parameters

<code>_idSosta</code>	indice della sosta già acquistata
<code>_dOwn</code>	nuova durata durata della sosta

5.1.3 Variable Documentation

5.1.3.1 nPosti

```
uint nPosti = 10
```

numero totale di posti nel parcheggio

5.1.3.2 targaProva

```
string targaProva = "AB000CD"
```

placeholder per la targa

5.1.3.3 maxTime

```
uint maxTime = 10000000000
```

timestamp usato come limite superiore

5.1.3.4 maxDurata

```
uint maxDurata = 86400
```

durata in secondi di un giorno, usato come limite superiore

5.1.3.5 ether1

```
uint ether1 = 1000000000000000000
```

valore in wei di 1 ether

5.1.3.6 addr

```
address addr
```

indirizzo del proprietario del parcheggio

5.1.3.7 ownAddr

```
address ownAddr
```

indirizzo del proprietario del parcheggio

Index

acquisto
 Parking.c, 13
addr
 Parking.c, 15

checkDisp
 Parking.c, 12
constructor
 Parking.c, 12
coordinate
 Posto, 7

durata
 Sosta, 8

estensione
 Parking.c, 14
ether1
 Parking.c, 15

idPosto
 Sosta, 9
idSosta
 Posto, 7

maxDurata
 Parking.c, 15
maxTime
 Parking.c, 15

nPosti
 Parking.c, 15
nextSosta
 Sosta, 9

ora
 Sosta, 8
ownAddr
 Parking.c, 15

pago
 Parking.c, 13
Parking.c, 11
 acquisto, 13
 addr, 15
 checkDisp, 12
 constructor, 12
 estensione, 14
 ether1, 15
 maxDurata, 15
 maxTime, 15

 nPosti, 15
 ownAddr, 15
 pago, 13
 prenotazione, 14
 ricercaPosto, 14
 targaProva, 15
Posto, 7
 coordinate, 7
 idSosta, 7
 prezzo, 7
prenotazione
 Parking.c, 14
prezzo
 Posto, 7

ricercaPosto
 Parking.c, 14

Sosta, 8
 durata, 8
 idPosto, 9
 nextSosta, 9
 ora, 8
 targa, 8

targa
 Sosta, 8
targaProva
 Parking.c, 15