Winning Software Solution
winningsoftwaresolution@gmail.com

ShopChain
SyncLab

Analisi dei requisiti

Informazioni	
Redattori	Raffaele Oliviero
Versione	0.0.2
Uso	esterno

Descrizione
Lista Metodi

Versione	Data	Persona	Attivtà	Descrizione
0.0.2	5/3/2022	Raffaele Oliviero	Redazione	Inserimento metodi pubblici e stesura 2
0.0.1	4/3/2022	Andrea Volpe	Redazione	Inserimento metodi pubblici
0.0.0	2/2/2022	Raffaele Oliviero	Redazione	Strutturazione del documento

1 Introduzione

1.1 Scopo del documento

Il documento elenca i metodi pubblici di ogni classe, descrivendone i loro parametri di invocazione, i loro return e il comportamento.

2 Tipi usati frequentemente

Un breve elenco dei tipi comunemente usati e dei loro attributi

2.1 paymentEntry

Un **articolo** in vendita
Parametri:

`id` l'id dell'articolo
`seller` l'indirizzo del wallet del venditore
`price` il prezzo in centesimi di dollaro

2.2 settledPayment

Una **transazione**
Parametri:

`id` l'id della transazione
`paymentEntryID` l'id dell'articolo acquistato
`client` l'indirizzo del wallet dell'acquirente
`status` lo stato della transazione con i seguenti significati:

- `0` la transazione è stata cancellata
- `1` la transazione è stata pagata
- `2` i fondi sono stati sbloccati
- `3` la transazione è timed out

`created` la date di creazione della transazione
`confirmed` se la transazione è stata chiusa, ne indica la data di chiusura

2.3 payment

Una **transazione** in cui sono contenute più informazioni
Parametri:

`id` l'id della transazione
`buyer` l'indirizzo del wallet dell'acquirente
`seller` l'indirizzo del wallet del venditore
`price` il prezzo in centesimi di dollaro
`status` lo stato della transazione con i seguenti significati:

- `0` la transazione è stata cancellata
- `1` la transazione è stata pagata

`2` i fondi sono stati sbloccati
`3` la transazione è timed out

`created` la data di creazione della transazione
`confirmed` se la transazione è stata chiusa, ne indica la data di chiusura

3 Lista dei metodi pubblici

3.1 Persistence

class Persistence

3.1.1 getPaymentByBuyer(buyer): Promise<payment[ ]>

Restituisce le transazioni il cui acquirente corrisponde a `buyer`
Parametri
buyer: string
Valore Restituito
Promise<payment[ ]>

3.1.2 getPaymentBySeller(seller): Promise<payment[ ]>

Restituisce le transazioni il cui venditore corrisponde a `buyer`
Parametri
seller: string
Valore Restituito
Promise<payment[ ]>

3.1.3 getPaymentEntryByID(id): Promise<paymentEntry>

Restituisce il prodotto identificato da `it`
Parametri
id: bigint
Valore Restituito
Promise<paymentEntry>

3.1.4 getPaymentByID(id): Promise<payment>

Restituisce la transazione identificata da `id`
Parametri
id: bigint
Valore Restituito
Promise<payment>

3.2 ServerManager

class ServerManager

3.2.1 setContract(shopContract) : ServerManager

Imposta un contratto
Parametri
shopContract : ShopContract_Interface
Valore Restituito
ServerManager

3.2.2 setSQL(sql) : ServerManager

Imposta una connessione SQL
Parametri
sql : SQL_Interface
Valore Restituito
ServerManager

3.2.3 setPageCreator(page): ServerManager

Imposta un PageCreator
Parametri
page : PageCreator
Valore Restituito
ServerManager

3.2.4 start() : ServerManager

Crea e avvia un nuovo server
Parametri
Valore Restituito
ServerManager
Eccezioni
AssertionError?

3.2.5 closeServer() : void

Chiude la connessione con il server
Parametri
Valore Restituito
void

3.3 SQL_Interface

interface SQL_Interface

3.3.1 insertPaymentEntry(entry): Promise<void>

Inserisce un nuovo prodotto nel database
Parametri
entry: paymentEntry

Valore Restituito
Promise<void>

3.3.2 insertSettledPayment(entry): Promise<void>

Inserisce un nuovo pagamento nel database
Parametri
entry: settledPayment
Valore Restituito
Promise<void>

3.3.3 updateSettledPayment(id, status, timestamp): Promise<void>

Aggiorna un pagamento nel database
Parametri
id: bigint,
status: number,
timestamp: bigint
Valore Restituito
Promise<void>

3.3.4 getPaymentByBuyer(buyer): Promise<payment[ ]>

Restituisce i pagamenti di uno specifico acquirente
Parametri
buyer: string
Valore Restituito
Promise<payment[ ]>

3.3.5 getPaymentBySeller(seller): Promise<payment[ ]>

Restituisce i pagamenti di uno specifico venditore
Parametri
seller: string
Valore Restituito
Promise<payment[ ]>

3.3.6 getPaymentEntryByID(id): Promise<paymentEntry>

Restituisce uno specifico prodotto
Parametri
id: bigint
Valore Restituito
Promise<paymentEntry>

3.3.7 getPaymentByID(id): Promise<payment>

Restituisce uno specifico pagamento
Parametri
id: bigint

Valore Restituito
Promise<payment>

3.3.8 getLastSyncBlock(): Promise<number>

Restituisce l'ultimo blocco sincronizzato
Parametri
Valore Restituito
Promise<number>

3.3.9 setLastSyncBlock(block): Promise <void>

Imposta l'ultimo blocco sincronizzato al valore passato come parametro
Parametri
block: number
Valore Restituito
Promise<void>

3.4 SQL

class SQL implements SQL_Interface

3.4.1 closeConnection(): void

Chiude la connessione con il database
Parametri
Valore Restituito
void

3.4.2 insertPaymentEntry(entry): Promise<void>

Inserisce un nuovo prodotto nel database
Parametri
entry: paymentEntry
Valore Restituito
Promise<void>

3.4.3 insertSettledPayment(entry): Promise<void>

Inserisce un nuovo pagamento nel database
Parametri
entry: settledPayment
Valore Restituito
Promise<void>

3.4.4 updateSettledPayment(id, status, timestamp): Promise<void>

Aggiorna un pagamento nel database
Parametri
id: bigint,

status: number,
timestamp: bigint
Valore Restituito
Promise<void>

3.4.5 getPaymentByBuyer(buyer): Promise<payment[ ]>

Restituisce i pagamenti di uno specifico acquirente
Parametri
buyer: string
Valore Restituito
Promise<payment[ ]>

3.4.6 getPaymentBySeller(seller): Promise<payment[ ]>

Restituisce i pagamenti di uno specifico venditore
Parametri
seller: string
Valore Restituito
Promise<payment[ ]>

3.4.7 getPaymentEntryByID(id): Promise<paymentEntry>

Restituisce uno specifico prodotto
Parametri
id: bigint
Valore Restituito
Promise<paymentEntry>

3.4.8 getPaymentByID(id): Promise<payment>

Restituisce uno specifico pagamento
Parametri
id: bigint
Valore Restituito
Promise<payment>

3.4.9 getLastSyncBlock(): Promise<number>

Restituisce l'ultimo blocco sincronizzato
Parametri
Valore Restituito
Promise<number>

3.4.10 setLastSyncBlock(block): Promise <void>

Imposta l'ultimo blocco sincronizzato al valore passato come parametro
Parametri
block: number

Valore Restituito
Promise<void>

3.5 Server

class Server

3.5.1 close(): void

Chiude la connessione con il server
Parametri
Valore Restituito
void

3.6 PageCreator

class PageCreator

3.6.1 landPage(req, res, db): void

Renderizza e invia la pagina `landing page` al client
Parametri
req: Request,
res: Response,
db: Persistence
Valore Restituito
void

3.6.2 helpPage(req, res): void

Renderizza e invia la pagina `help` al client
Parametri
req: Request,
res: Response
Valore Restituito
void

3.6.3 mainPage(req, res): void

Renderizza e invia la pagina `principale` al client
Parametri
req: Request,
res: Response
Valore Restituito
void

3.6.4 confirmPage(req, res, db): void

Renderizza e invia la pagina `?` al client
Parametri

req: Request,
res: Response,
db: Persistence
Valore Restituito
void

3.6.5 paymentByBuyerPage(req, res, db): void

Renderizza e invia la pagina delle `transazioni effettuate` al client
Parametri
req: Request,
res: Response,
db: Persistence
Valore Restituito
void

3.6.6 detailPage(req, res, db): void

Renderizza e invia la pagina dei `dettagli transazione?` al client
Parametri
req: Request,
res: Response,
db: Persistence
Valore Restituito
void

3.6.7 paymentBySellerPage(req, res, db): void

Renderizza e invia la pagina delle `transazioni in ingresso` al client
Parametri
req: Request,
res: Response,
db: Persistence
Valore Restituito
void

3.7 ShopContract_Interface

interface ShopContract_Interface

3.7.1 getBlockTime(block) : Promise<bigint>

Restituisce il timestamp di `block`
Parametri
block: number

3.7.2 addedPaymentEntry(options) : EventEmitter

Emette un evento di articolo aggiunto
Parametri
options: pastEventOptions

3.7.3 paymentSettled(options) : EventEmitter

Emette un evento di transazione effettuata
Parametri
options: pastEventOptions

3.7.4 statusChange(options) : EventEmitter

Emette un evento di cambio di stato
Parametri
options: pastEventOptions

3.7.5 getSettledPayment(id) : Promise<settledPayment>

Restituisce la transazione corrispondente a `id`
Parametri
id: bigint

3.7.6 getPaymentEntry(id) : Promise<settledPayment>

Restituisce l'articolo in vendita corrispondente a `id`
Parametri
id: bigint

3.8 ShopContract (typescript)

ShopContract implementa ShopContract_Interface

3.8.1 getBlockTime(block) : Promise<bigint>

Restituisce il timestamp di `block`
Parametri
block: number

3.8.2 addedPaymentEntry(options) : EventEmitter

Emette un evento di articolo aggiunto
Parametri
options: pastEventOptions

3.8.3 paymentSettled(options) : EventEmitter

Emette un evento di transazione effettuata
Parametri
options: pastEventOptions

3.8.4 statusChange(options) : EventEmitter

Emette un evento di cambio di stato
Parametri
options: pastEventOptions

3.8.5 getSettledPayment(id) : Promise<settledPayment>

Restituisce la transazione corrispondente a `id`
Parametri
id: bigint

3.8.6 getPaymentEntry(id) : Promise<settledPayment>

Restituisce l'articolo in vendita corrispondente a `id`
Parametri
id: bigint

3.9 ShopContractEventManager

ShopContract implementa ShopContract_Interface

3.9.1 Contstructor

3.10 ShopContract (solidity)

3.10.1 getLatestPrice() : uint256

Restituisce il prezzo aggiornato

3.10.2 checkUpkeep(calldata) : (bool, bytes)

Restituisce `true` se è passata una quantità di tempo maggiore di 60 secondi dall'ultimo controllo
Parametri
calldata: byte

3.10.3 performUpkeep(calldata) : void

Aggiorna le transazioni che sono scadute, cambiandone lo stato, e aggiorna i dati dell'ultimo controllo effettuato
Parametri
calldata: byte

3.10.4 settlePayment(paymentEntryId) : void

Controlla che i soldi ricevuti siano sufficienti a pagare il costo dell'articolo identificato da `paymentEntryId` e, in caso positivo, aggiunge una transazione alla lista delle transazioni, impostandone lo stato a `1`
Parametri
paymentEntryId: uint256

3.10.5 unlockFunds(settledPaymentId) : void

Controlla che chi ha inviato il messaggio corrisponda all'acquirente della transazione identificata da `settledPaymentId` e che la transazione sia in stato `1`; in caso positivo, trasferisce i fondi al venditore e imposta lo stato della transazione a `2`
Parametri
settledPaymentId: uint256

3.10.6 cancelPayment(settledPaymentId) : void

Controlla che chi ha inviato il messaggio corrisponda al venditore della transazione identificata da `settledPaymentId` e che la transazione sia in stato `1`; in caso positivo, restituisce i fondi all'acquirente e imposta lo stato della transazione a `0`
Parametri
settledPaymentId: uint256

3.10.7 getPaymentEntry(paymentEntryId) : PaymentEntry

Se esiste, restituisce l'articolo il cui id corrisponde a `paymentEntryId`
Parametri
paymentEntryId: uint256

3.10.8 getSettledPayment(settledPaymentId) : SettledPayment

Se esiste, restituisce la transazione il cui id corrisponde a `settledPaymentId`
Parametri
settledPaymentId: uint256