



Integrazione cassetto denaro elettronico CM

Creato da: Luca Sassi

Ultimo aggiornamento: 13/11/2025 14:12:31

Modificato da: Luca Sassi

Versione Documento: 1.0

Indice del documento

1.	Introduzione	1
1.1.	Tecnologie utilizzate	1
1.1.1.	Comunicazione tramite Web Socket.....	1
1.1.2.	Comunicazione tramite API REST	1
1.2.	Documentazione	1
2.	Flusso di comunicazione con Web Socket.....	2
2.1.	Processo di autenticazione	2
2.1.1.	Dettagli della Richiesta POST	2
2.2.	Effettuare una connessione Web Socket.....	3
2.3.	Comporre un comando per il Web Socket	3
2.3.1.	Corpo del Messaggio (Body)	3
2.3.2.	Autenticazione (Header)	3
2.4.	Messaggi inviati dal client al cassetto denaro elettronico	3
2.4.1.	Messaggio di presentazione (handshake).....	4
2.4.2.	Comando.....	4
2.5.	Messaggi inviati dal cassetto denaro elettronico al client	4
2.5.1.	Gestione delle Risposte e degli Eventi	4
2.5.2.	Formato e Classificazione dei Messaggi.....	4
2.5.3.	Messaggio "start_command_event"	5
2.5.4.	Messaggio "end_command_event"	5
2.5.5.	Messaggio "event"	5
2.5.6.	Messaggio "error".....	5
2.5.7.	Classificazione dell'Errore.....	6
2.5.8.	Messaggio "status_event_obj"	6
3.	Elenco dei comandi che si possono inviare	7
4.	Presentazione (handshake)	8
5.	Reset cassetto	9
6.	Eroga importo	10
7.	Eroga per denominazioni	11

8.	Vendita	14
9.	Annullo vendita.....	15
10.	Avvio caricamento denaro	16
11.	Termine caricamento denaro	17
12.	Versa importo nella cashbox.....	18
13.	Versa importo nella cashbox per denominazioni	19
14.	Riporta ai minimi.....	22
15.	Svuota cassetto (completo)	23
16.	Svuota cassetto (solo monete).....	24
17.	Svuota cassetto (solo banconote)	25
18.	Svuota cashbox	26
19.	Stato cassetto.....	27
20.	Appendice	28
20.1.	Elenco comandi	28
20.2.	Posizione delle denominazioni.....	28
20.3.	Stato del cassetto	29
20.4.	Elenco Errori.....	29
20.5.	Elenco subErrors	32
20.6.	Elenco eventi.....	33
20.7.	Tipo denominazioni.....	34
20.8.	Stato denominazioni	35
20.9.	Modalità inventario.....	35
21.	Cronologia delle versioni	36

1. Introduzione

Questo manuale **descrive** i flussi logici **necessari** per implementare l'**integrazione a basso livello** del cassetto denaro elettronico **CM**. L'**entità principale** del cassetto, denominata "**Business Logic**", gestisce a basso livello i **dispositivi meccanici** installati (meccanismi di gestione monete e banconote) e **esponde** una serie di comandi specifici per **l'inserimento e il prelievo** di denaro.

1.1. Tecnologie utilizzate

Il **dialogo** con la Business Logic richiede l'implementazione di due canali di comunicazione: **Web Socket** e **API REST**.

1.1.1. Comunicazione tramite Web Socket

La comunicazione tramite Web Socket prevede:

- **Creazione del canale:** Instaurazione di un canale diretto tra client e server.
- **Presentazione del client:** Autenticazione iniziale del client verso il server.
- **Canale persistente:** Una volta completata la presentazione, il canale rimane permanentemente aperto, garantendo una **comunicazione bidirezionale** continua.
- **Invio comandi:** Il client può inviare comandi al server.
- **Ricezione eventi (Push):** Il client può ricevere **eventi** dal server in modo asincrono, anche in assenza di una richiesta esplicita.

1.1.2. Comunicazione tramite API REST

La comunicazione tramite **API REST** si basa sull'invio di richieste **POST** e **GET** per:

- Recuperare **informazioni** relative al cassetto denaro elettronico.
- Ottenere il **token** di autenticazione.

1.2. Documentazione

Di seguito sono indicati gli URL per poter accedere alla documentazione completa delle API sia per il Web Socket che per API REST.

[Documentazione Web Socket](#)

[Documentazione REST API](#)

2. Flusso di comunicazione con Web Socket

Le fasi operative per stabilire la comunicazione tramite **Web Socket** sono:

1. **Autenticazione (Login):** Esecuzione del processo di autenticazione (login) tramite l'API REST dedicata.
2. **Connessione:** Stabilire la connessione del client al cassetto tramite il canale Web Socket.
3. **Handshake:** Invio del messaggio di presentazione del client (**handshake**).
4. **Invio Comandi:** Trasmissione di comandi al server.
5. **Gestione Messaggi:** Ricezione e gestione dei messaggi inviati dal server (eventi o risposte).

2.1. Processo di autenticazione

Per inviare comandi al cassetto, il client deve prima eseguire l'**autenticazione**.

Questa operazione viene gestita tramite l'API REST dedicata al **login** (auth/login).

2.1.1. Dettagli della Richiesta POST

Il client deve inviare una richiesta **POST** con le seguenti specifiche:

- **URL dell'API:** /auth/login
- **Corpo della richiesta (Payload):** Assente (nessun corpo).
- **Header:** Deve includere un unico parametro HTTP: Authorization .
- **Formato del Valore:** Il valore del parametro Authorization deve essere una stringa **codificata in Base64**, ottenuta dalla concatenazione di **utente** e **password** separati dal carattere pipe (|).

Esempio:

utente **cassa**

password **cassa**

La stringa concatenata diventa **cassa|cassa**

La stringa concatenata e codificata in Base 64 diventa **Y2Fzc2F8Y2Fzc2E=**

Utente e password di default sono "cassa" e "cassa", ma possono essere modificati.

La risposta a questa API REST conterrà il token rilasciato dal cassetto durante la fase di autenticazione.

Questo token dovrà essere utilizzato per ogni comando inviato al cassetto denaro elettronico.

2.2. Effettuare una connessione Web Socket

Il **client** deve stabilire la connessione con il **Web Socket** del cassetto denaro elettronico.

Le specifiche metodologie di implementazione dipendono dal linguaggio di programmazione utilizzato per lo sviluppo dell'applicativo client.

2.3. Comporre un comando per il Web Socket

Ogni **comando** inviato tramite **Web Socket** è strutturato nelle seguenti due parti:

1. **Corpo del Messaggio (Body)**
2. **Header (per l'Autenticazione)**

2.3.1. Corpo del Messaggio (Body)

Il corpo del messaggio è un documento in formato **JSON** che contiene le chiavi specifiche per il comando destinato al cassetto denaro elettronico. Per i dettagli, fare riferimento alla documentazione completa del Web Socket.

2.3.2. Autenticazione (Header)

Per l'autenticazione, è necessario includere il parametro `Authorization` nell'**Header** del messaggio di comando.

Il valore del parametro deve essere valorizzato con lo schema `Bearer` seguito da uno spazio e dal **token** precedentemente ottenuto tramite l'autenticazione API REST:

`Authorization: Bearer [TOKEN]`

`Bearer 328wefih98y4fweiojfiojfihgwe7cwe90fjoiehf8ds78wd89whvweh8`

2.4. Messaggi inviati dal client al cassetto denaro elettronico

Il **client** può inviare al cassetto denaro elettronico i seguenti due tipi di messaggi:

1. **Messaggio di Handshake:** Utilizzato per la presentazione iniziale del client al server.
2. **Messaggio di Comando:** Utilizzato per richiedere operazioni specifiche al cassetto.

2.4.1. Messaggio di presentazione (handshake)

Il **messaggio di Handshake** è **fondamentale** in quanto abilita la comunicazione tra client e cassetto denaro elettronico.

Requisito: Deve essere inviato come **primo messaggio** assoluto.

In assenza di presentazione (Handshake), qualsiasi **messaggio di comando** inviato dal client genererà un **evento di errore**. L'invio di comandi non sarà abilitato finché il messaggio di Handshake non sarà stato trasmesso con successo.

2.4.2. Comando

Il **messaggio di Comando** viene utilizzato per **avviare** nuove operazioni.

Riferimento: La lista completa dei comandi disponibili è elencata nell'**Appendice** di questo documento.

2.5. Messaggi inviati dal cassetto denaro elettronico al client

La comunicazione tramite **Web Socket** è **bidirezionale**, consentendo uno scambio costante tra client e cassetto denaro elettronico.

2.5.1. Gestione delle Risposte e degli Eventi

A fronte della ricezione di un comando, del verificarsi di un errore, o in presenza di un semplice evento, il cassetto denaro elettronico **invia un messaggio di risposta** al client.

È responsabilità del client **analizzare** ogni messaggio ricevuto e implementare la logica necessaria per agire in base al tipo di messaggio identificato.

2.5.2. Formato e Classificazione dei Messaggi

Tutti i messaggi di risposta sono in formato **JSON** e aderiscono a uno schema preciso (i dettagli sono forniti nella documentazione del Web Socket).

Questi messaggi sono classificati nei seguenti tipi:

- start_command_event
- end_command_event
- error
- event
- status_event_obj

2.5.3. Messaggio "start_command_event"

Questo messaggio (tipo `start_command_event`) viene inviato dal cassetto denaro elettronico immediatamente dopo la ricezione di un nuovo comando.

La sua funzione è quella di confermare al client l'avvenuta "**presa in carico**" del comando.

Il messaggio include l'**inventario** attuale, ovvero la lista dei tagli di denaro (con le rispettive quantità) presenti all'interno del cassetto elettronico.

2.5.4. Messaggio "end_command_event"

Il messaggio di tipo `end_command_event` viene inviato dal cassetto denaro elettronico al termine dell'esecuzione di un comando.

Il messaggio fornisce diverse informazioni di *summary* post-esecuzione:

1. **Esito del Comando:** Indica il risultato finale dell'operazione.
2. **Importo Richiesto:** Specifica l'importo iniziale (rilevante, ad esempio, per le operazioni di vendita).
3. **Importo Inserito:** Quantità totale di denaro inserita dal cliente.
4. **Importo Erogato:** Quantità di denaro erogata (ad esempio, resto di una vendita o denaro prelevato manualmente).
5. **Inventario Finale:** Lista dei tagli di denaro e delle relative quantità presenti nel cassetto al termine del comando.

2.5.5. Messaggio "event"

Il messaggio di tipo `event` viene inviato dal cassetto in modo **asincrono** e in qualsiasi momento, anche in assenza di comandi attivi.

Questo messaggio di notifica include le seguenti informazioni:

- **Codice Evento:** Identificativo univoco dell'evento.
- **Descrizione Evento:** Testo descrittivo associato all'evento.
- **Codice Interno:** Codice evento originale inviato dal dispositivo meccanico.
- **Dati Dispositivo:** Informazioni sul dispositivo che ha generato l'evento (tipo, marca e modello).

Ad esempio, l'evento `coinInserted` viene inviato dal cassetto denaro elettronico al client ogni volta che viene inserita una moneta.

2.5.6. Messaggio "error"

Il messaggio di tipo `error` viene inviato dal cassetto denaro elettronico quando viene riscontrata un'anomalia durante l'esecuzione di un comando.

Il messaggio fornisce le seguenti informazioni dettagliate sull'errore:

- **Codice Errore:** Identificativo numerico dell'errore.
- **Descrizione Errore:** Testo esplicativo associato all'errore.
- **Sub-errore:** Eventuale codice e nome correlato all'errore primario.
- **Dati Dispositivo:** Informazioni sul dispositivo che ha generato l'errore (tipo, marca e modello).

2.5.7. Classificazione dell'Errore

Il codice di errore consente di determinare la gravità dell'anomalia, classificandola come "**bloccante**" o come semplice **avvertimento (warning)**.

Esempio bloccante: L'errore `busy` viene restituito qualora si tenti di inviare un secondo comando mentre un'operazione precedente è ancora in corso.

2.5.8. Messaggio "status_event_obj"

Il messaggio di tipo `status_event_obj` racchiude l'insieme completo delle informazioni relative allo **stato operativo** del cassetto denaro elettronico.

Questo messaggio include i seguenti dati chiave:

1. **Stato del Dispositivo:** La condizione operativa attuale del cassetto.
2. **Comando Attivo:** Indicazione di un eventuale comando in corso di esecuzione.
3. **Inventario:** L'elenco aggiornato dei tagli di denaro e delle relative quantità.

3. Elenco dei comandi che si possono inviare

Nelle pagine seguenti, saranno illustrati tutti i comandi disponibili:

- Presentazione (handshake)
- Reset cassetto
- Eroga importo
- Eroga per denominazioni
- Vendita
- Annulla vendita
- Avvio caricamento denaro
- Termine caricamento denaro
- Versa importo nella cashbox
- Versa importo nella cashbox per denominazioni
- Riporta ai minimi
- Svuota cassetto (completo)
- Svuota cassetto (solo monete)
- Svuota cassetto (solo banconote)
- Svuota cashbox
- Stato cassetto
- Reset giacenza

4. Presentazione (handshake)

Questo messaggio serve per presentarsi al server. Se non si effettua la presentazione dopo la connessione, ogni comando sarà rifiutato.

```
{
    "type": "handshake",
    "websocketClientType": "client",
    "clientCode": "5",
    "clientName": "",
    "clientUserCode": "",
    "clientUserName": "",
    "force": true
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre handshake	String
websocketClientType	Tipo di client: sempre client	String
clientCode	Codice del client che si connette. È importante che non vengano eseguite presentazioni con lo stesso codice client da macchine diverse	String
clientName	Identificativo della macchina client che si presenta	String
clientUserCode	Codice utente che sta usando il client	String
clientUserName	Nome utente che sta usando il client	String
force	Se impostato a “true” viene forzata la presentazione	Boolean

5. Reset cassetto

Questo comando avvia il reset del cassetto denaro elettronico.

```
{  
    "type": "command",  
    "code": 255,  
    "name": "nome client",  
    "parameters": {}  
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 255	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri (indicarla con una mappa vuota)	Map

6. Eroga importo

Questo comando serve per erogare denaro indicando un importo. Il cassetto denaro elettronico sceglierà autonomamente i tagli da erogare in base alla sua giacenza attuale e alle impostazioni di minimi e massimi.

```
{
    "type": "command",
    "code": 20,
    "name": "nome client",
    "parameters": {
        "amount": 200
    }
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 20	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri	Map
amount	Importo da erogare espresso in centesimi (Es: 1,00 Euro → 100)	Int

7. Eroga per denominazioni

Questo comando serve per erogare denaro indicando gli specifici tagli da utilizzare.

Prestare attenzione: indicare solo ed esclusivamente i tagli realmente presenti nel cassetto, altrimenti si riceverà un errore di impossibilità di erogazione denaro per insufficienza dei tagli.

```
{
    "type": "command",
    "code": 20,
    "name": "nome client",
    "parameters": {
        "amount": 2500,
        "denominationList": [
            {
                "denominationType": "NOTE",
                "valueInt": 1000,
                "currencyCode": "EUR",
                "valueString": "10,00EUR",
                "recyclingEnabled": true,
                "active": true,
                "stock": 2,
                "stock_recycler": 0,
                "stock_cashbox": 0,
                "channel": -1,
                "totalAmount": 2000,
                "totalAmountString": "20,00EUR"
            },
            {
                "denominationType": "NOTE",
                "valueInt": 500,
                "currencyCode": "EUR",
                "valueString": "5,00EUR",
                "recyclingEnabled": true,
                "active": true,
                "stock": 1,
                "stock_recycler": 0,
                "stock_cashbox": 0,
                "channel": -1,
                "totalAmount": 500,
                "totalAmountString": "5,00EUR"
            }
        ]
    }
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String

Chiave	Descrizione	Tipo dato
code	Codice del comando: sempre 20	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri	Map
amount	Importo da versare nella cashbox espresso in centesimi (Es: 1,00 Euro → 100)	Int
denominationList	Lista delle denominazioni da versare nella cashbox	List<Map>
denomination	Mappa della denominazione	Map
denominationType	Tipo di denominazione	String
valueInt	Valore facciale della denominazione espresso in centesimi (1,00 Euro → 100)	Int
currencyCode	Valuta della denominazione	String
valueString	Valore facciale della denominazione espresso in String	String
recyclingEnabled	Indica se il taglio indicato è "riciclabile" oppure no	Boolean
active	Indica se il taglio è attivo	Boolean
stock	Indica il numero di tagli da versare nella cashbox	Int
stock_recycler	Indicare sempre il valore 0	Int
stock_cashbox	Indicare sempre il valore 0	Int
totalAmount	Indicare il valore complessivo del taglio da versare nella cashbox	Int
totalAmountString	Indicare il valore complessivo del taglio da versare nella cashbox espresso in String	String

8. Vendita

Questo comando server per avviare una transazione di vendita.

```
{
    "type": "command",
    "code": 22,
    "name": "nome client",
    "parameters": {
        "amount": 500
    }
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 22	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri	Map
amount	Parametro del comando: indicare l'importo da accettare espresso in centesimi (Es: 1,00 Euro → 100)	Int

9. Annullo vendita

Questo comando serve per interrompere una transazione di vendita. Eventuale denaro inserito sarà espulso. Se lanciato senza una transazione di vendita in corso, si riceverà un errore.

```
{
    "type": "command",
    "code": 24,
    "name": "nome client",
    "parameters": {}
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 24	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri (indicarla con una mappa vuota)	Map

10. Avvio caricamento denaro

Questo comando serve per avviare il caricamento di denaro all'interno del cassetto denaro elettronico.

```
{  
    "type": "command",  
    "code": 30,  
    "name": "nome client",  
    "parameters": {}  
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 30	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri (indicarla con una mappa vuota)	Map

11. Termine caricamento denaro

Questo comando serve per interrompere il caricamento di denaro. Se lanciato senza un caricamento denaro in corso, si riceverà un errore.

```
{  
    "type": "command",  
    "code": 31,  
    "name": "nome client",  
    "parameters": {}  
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 31	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri (indicarla con una mappa vuota)	Map

12. Versa importo nella cashbox

Questo comando serve per depositare del denaro nelle cashbox indicando un importo (monete e banconote). Le monete e le banconote interessate dal versamento, saranno depositate nelle rispettive cassette poste al di sotto dei dispositivi.

```
{
  "type": "command",
  "code": 40,
  "name": "nome client",
  "parameters": {
    "amount": 1500
  }
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 40	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri	Map
amount	Importo da versare nella cashbox espresso in centesimi (Es: 1,00 Euro → 100)	Int

13. Versa importo nella cashbox per denominazioni

Questo comando serve per depositare del denaro nelle cashbox indicando un importo (monete e banconote) indicando i tagli esatti da depositare. Le monete e le banconote interessate dal versamento, saranno depositate nelle rispettive cassette poste al di sotto dei dispositivi.

Prestare attenzione: indicare solo ed esclusivamente i tagli realmente presenti nel cassetto, altrimenti si riceverà un errore di impossibilità di erogazione denaro per insufficienza dei tagli.

```
{
    "type": "command",
    "code": 40,
    "name": "nome client",
    "parameters": {
        "amount": 2500,
        "denominationList": [
            {
                "denominationType": "NOTE",
                "valueInt": 1000,
                "currencyCode": "EUR",
                "valueString": "10,00EUR",
                "recyclingEnabled": true,
                "active": true,
                "stock": 2,
                "stock_recycler": 0,
                "stock_cashbox": 0,
                "channel": -1,
                "totalAmount": 2000,
                "totalAmountString": "20,00EUR"
            },
            {
                "denominationType": "NOTE",
                "valueInt": 500,
                "currencyCode": "EUR",
                "valueString": "5,00EUR",
                "recyclingEnabled": true,
                "active": true,
                "stock": 1,
                "stock_recycler": 0,
                "stock_cashbox": 0,
                "channel": -1,
                "totalAmount": 500,
                "totalAmountString": "5,00EUR"
            }
        ]
    }
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String

Chiave	Descrizione	Tipo dato
code	Codice del comando: sempre 40	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri	Map
amount	Importo da versare nella cashbox espresso in centesimi (Es: 1,00 Euro → 100)	Int
denominationList	Lista delle denominazioni da versare nella cashbox	List<Map>
denomination	Mappa della denominazione	Map
denominationType	Tipo di denominazione	String
valueInt	Valore facciale della denominazione espresso in centesimi (1,00 Euro → 100)	Int
currencyCode	Valuta della denominazione	String
valueString	Valore facciale della denominazione espresso in String	String
recyclingEnabled	Indica se il taglio indicato è "riciclabile" oppure no	Boolean
active	Indica se il taglio è attivo	Boolean
stock	Indica il numero di tagli da versare nella cashbox	Int
stock_recycler	Indicare sempre il valore 0	Int
stock_cashbox	Indicare sempre il valore 0	Int
totalAmount	Indicare il valore complessivo del taglio da versare nella cashbox	Int
totalAmountString	Indicare il valore complessivo del taglio da versare nella cashbox espresso in String	String

14. Riporta ai minimi

Questo comando serve per espellere monete e/o banconote fino al raggiungimento dei minimi impostati su ciascun taglio.

```
{
  "type": "command",
  "code": 50,
  "name": "nome client",
  "parameters": {
    "device": "ALL"
  }
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 50	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri	Map
device	Indicare la tipologia di denominazioni (vedi qui)	String

15. Svuota cassetto (completo)

Questo comando serve per avviare uno svuotamento completo del denaro.

Le monete e le banconote saranno depositate nelle rispettive cassette poste al di sotto dei dispositivi.

```
{
  "type": "command",
  "code": 100,
  "name": "nome client",
  "parameters": {}
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 100	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri (indicarla con una mappa vuota)	Map

16. Svuota cassetto (solo monete)

Questo comando serve per svuotare il cassetto dalle monete.

Le monete saranno depositate nelle rispettive cassette poste al di sotto dei dispositivi.

```
{
    "type": "command",
    "code": 100,
    "name": "nome client",
    "parameters": {
        "device": "COIN"
    }
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 100	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri	Map
device	Indicare COIN	String

17. Svuota cassetto (solo banconote)

Questo comando serve per svuotare il cassetto dalle banconote.

Le banconote saranno depositate nelle rispettive cassette poste al di sotto dei dispositivi.

```
{
    "type": "command",
    "code": 100,
    "name": "nome client",
    "parameters": {
        "device": "COIN"
    }
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 100	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri	Map
device	Indicare NOTE	String

18. Svuota cashbox

Questo comando serve per azzerare l'inventario logico relativo ai tagli contenuti nella cashbox.

Da mandare quando si riceve l'errore di estrazione della cashbox

```
{  
    "type": "command",  
    "code": 101,  
    "name": "nome client",  
    "parameters": {}  
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 101	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri (indicarla con una mappa vuota)	Map

19. Stato cassetto

Questo comando serve per ottenere lo stato del cassetto e l'inventario attuale.

```
{
    "type": "command",
    "code": 200,
    "name": "luca",
    "parameters": {
        "inventory_mode": "all"
    }
}
```

Chiave	Descrizione	Tipo dato
type	Tipo del comando: sempre command	String
code	Codice del comando: sempre 200	Int
name	Nome mnemonico del client che invia il comando	String
parameters	Mappa dei parametri (indicarla con una mappa vuota)	Map
inventory_mode	Indicare quale tipo di inventario ricevere (vedi qui)	String

20. Appendice

20.1. Elenco comandi

Codice	Nome
255	Reset cassetto
20	Eroga importo
22	Vendita
24	Annullo vendita
30	Avvio caricamento denaro
31	Termine caricamento denaro
40	Versa importo nella cashbox
50	Riporta ai minimi
100	Svuota cassetto
101	Svuota cashbox
200	Stato cassetto
500	Reset giacenza

20.2. Posizione delle denominazioni

Codice	Nome
0	Riciclatore

Codice	Nome
1	Cashbox

20.3. Stato del cassetto

Codice	Nome
-2	Sconosciuto
-1	Non pronto
0	Pronto
1	Inizializzazione
2	Attenzione
3	Errore
4	In accettazione denaro
5	Erogazione denaro
6	In fase di reset
7	Non inizializzato

20.4. Elenco Errori

Codice	Nome
-1	Riscontrata eccezione
-2	Sconosciuto

Codice	Nome
-3	Modello cassetto non conosciuto
-1001	Comando non previsto o non implementato
-1002	Comando non supportato
-1003	Numero di parametri comando errato
-1004	Valore dei parametri passati superiore al massimo previsto
-1005	Il comando non può essere processato ora
-1006	Errore software
-1007	Chiave di criptazione assente
-1008	Calcolo del comando errato
-1009	Cassetto impegnato (busy)
-1010	Mancata risposta del dispositivo meccanico
-1011	Dispositivo meccanico disconnesso
-1012	Ricevuta risposta vuota dal dispositivo meccanico
-1013	Errore sequenza di sincronizzazione
-1014	Canale di comunicazione interrotto
-1015	Comando non inviato
-1016	Mancata lettura della risposta del dispositivo meccanico
-1017	Errore CRC
-1018	Impossibile processare il comando

Codice	Nome
-1019	Dispositivo meccanico non inizializzato
-1020	Comando fallito
-1021	Inceppamento
-1022	Dispositivo meccanico in stallo
-1023	Cashbox piena
-1024	Cashbox rimossa
-1025	Rullo banconote pieno
-1026	Hopper monete pieno
-1027	Dispositivo meccanico aperto
-1028	Accettatore banconote aperto
-1031	Inceppamento banconota esterno
-1032	Tentativo di furto rilevato
-1033	Banconota in accettazione espulsa durante il reset del cassetto
-1034	Banconota in deposito nello stacket durante il reset del cassetto
-1035	Erogazione interrotta
-1036	Ripristino inceppamento
-1037	Errore durante erogazione
-1038	Banconota riportata all'esterno durante il reset del cassetto
-1039	Banconota inserita nella cashbox durante l'avvio del cassetto

Codice	Nome
-1040	Banconota erogata durante l'avvio del cassetto
-1041	Calibrazione fallita
-1042	Cassetto in errore
-3000	Importo non trovato nel cassetto
-3001	Importo negativo rilevato
-5000	Tipo messaggio errato
-5001	Codice Client non presentato
-5002	Client già presentato
-5003	Client non presentato
-5004	Formato comando non corretto
-5006	Disallineamento tagli

20.5. Elenco subErrors

Codice	Nome
100	Sincronizzazione dispositivo non avvenuta
101	Dispositivo non abilitato durante l'inizializzazione
102	Generazione chiave non riuscita
103	Recupero dataset dispositivo fallito

20.6. Elenco eventi

Codice	Nome
0	Idle
7001	Reset cassetto terminato
7002	Lettura banconota
7004	Rifiuto banconota
7005	Banconota rifiutata
7006	Immagazzinamento banconota
7007	Banconota immagazzinata
7008	Cashbox inserita
7011	Inizializzazione in corso
7012	Erogazione in corso
7013	Erogazione terminata
7014	Moneta accettata
7028	Banconota accettata
7015	Svuotamento cassetto in corso
7016	Svuotamento cassetto completato
7027	Banconota spostata nella cashbox
7029	Banconota inserita nel riciclatore
7033	Spostamento denaro nella cashbox

Codice	Nome
7034	Denaro spostato nella cashbox
7032	Cashbox svuotata
7017	Banconota spostata nella cashbox
7018	Banconota espulsa ma non ritirata
7019	Inserimento denaro in corso
7020	Manutenzione richiesta
7021	Espulsione moneta
7022	Cassetto inizializzato
7026	Cassetto non inizializzato
7100	Erogazione resto avviata

20.7. Tipo denominazioni

Codice	Nome
ALL	Monete e Banconote
COIN	Monete
NOTE	Banconote

20.8. Stato denominazioni

Codice	Nome
-1	Sconosciuto
0	Vuoto
1	Quasi vuoto
2	Ok
3	Quasi pieno
4	Pieno
5	Negativo
6	Sovraccarico

20.9. Modalità inventario

Codice	Nome
all	Tutte le denominazioni
recycling	Denominazioni riciclabili
cashbox	Denominazioni in cashbox
notRecycling	Denominazioni non riciclabili

21. Cronologia delle versioni

Versione	Data	Descrizione modifica	Autore
1.0		Creazione del documento	Luca Sassi