

Documentation CSV-Syncer



Version 1.0

27.01.2021



Kasse Speedy

© mtMax GmbH
Bannholzweg 13
D-69168 Wiesloch
Deutschland

Tel. +49 – 6222 – 38254 – 0

<https://www.kasse-speedy.de>

info@mtMax.de

Table of Contents

INTRODUCTION	5
INSTALLATION	6
PREREQUISITES	6
ENABLE API ACCESS	6
INSTALLATION CSV-SYNCER	6
CONFIGURATION	7
BASIC CONFIGURATION	7
CONFIGURATION FOR CSV OUTPUT	7
FIRST SYNC RUN	8
CONFIGURATION FOR CSV INPUT	8
ADVANCED CONFIGURATION	9
TIPS AND TRICKS	10
CSV FORMAT	10
FULL SYNC VERSUS DELTA SYNC	10
FIRST SYNC / FULL SYNC TAKES VERY LONG	10
OPTIONAL: TRANSMITTING DATA USING TIMESTAMPS	10
CONFIGURATION CHANGES	10
PROBLEMS WITH MICROSOFT EXCEL	10
DATA TABLES	12
MASTER DATA	12
SD_CUSTOMERGROUPS	12
SD_CUSTOMERS	13
SD_DISCOUNTS	14
SD_PRODUCTGROUPS	15
SD_PRODUCTS	15
SD_BALANCES	18
SD_USERGROUPS	19
SD_USERPERMISSIONS	20
SD_USERS	21
TRANSACTIONAL DATA	22
SD_RECEIPTS	22
SD_RECEIPTPOS	23
SD_RECEIPTPOSDISCOUNTS	24
SD_BALANCEPOS	25
SD_CLOSINGRUNS	26
VERSION HISTORY	29

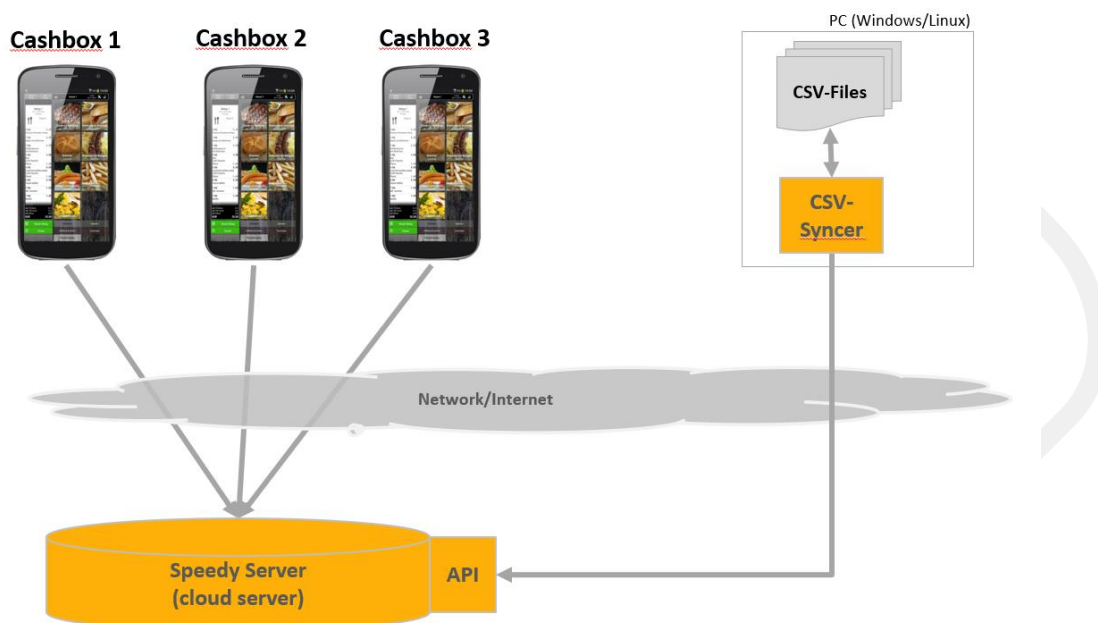


Introduction

The Android cashbox system „Kasse Speedy“ runs locally on any Android device. All cash data are stored on a local database.

By using the add-on „Network Sync“ (german “Netzwerkkopplung”) all cashbox data can be transmitted to a central database server (cloud server) back and forth. The data transmission is implemented as a two-way asynchronous data exchange. The transport protocol between the cashboxes and the server is HTTPS.

The CSV-Syncer is a small program which makes it easy to read and write such server data. The following picture gives an overview:



The CSV-Syncer is able to:

- read and write master data like: product groups, products, discounts, customer groups, customers, user groups, user group permissions, users, balances (warehouses)
- read transactional data like: receipts, receipt positions, closing runs (german: Endabrechnungen), balance positions (warehouse stock)

This document describes the installation and configuration of the CSV-Syncer.

Installation

Prerequisites

You need the following things to start:

- Any Android device, where Kasse Speedy is installed. APK download from <https://www.kasse-speedy.de/download>
- Import some demo data in Kasse Speedy: Settings menu → Import demo data → e.g. gastronomy.
- Enable the addon module “Network Sync” (dt. Netzwerkkopplung): Settings menu → Basic settings → License → Enable all switches (demo mode). The demo mode ends after an hour. You can enable the switches again (unlimited). Alternatively you may enter and activate a license key, if available.
- Go to settings menu → Network sync and create a new network account. Enable the Sync Service.
- To install and run the CSV-Syncer you need any PC, Laptop or server (Windows or Linux).

Enable API access

- Write an email to technik@mtmax.de and ask for API access. Provide your network account name (as created in the previous steps). Don't tell us your password! We will send you an API Key (“credential string”)

Installation CSV-Syncer

The CSV-Syncer may be installed on a normal PC, laptop or server (Windows or Linux).

- Make sure you have a Java runtime (JRE/JDK) installed on your PC or server. Download: <https://openjdk.java.net/install/>
- Copy the CSV-Syncer JAR file and the config.txt file to *any* directory on your PC or server.

Configuration

Open the config.txt file with any standard plain-text editor. Configuration lines starting with # are comments and are ignored during processing. All configuration parameters are written like: name=value

Basic configuration

Configuration	Comments
<code>accountName=my@account.de</code>	Name of your network sync account (server account), as created in chapter "Prerequisites"
<code>accountPassword=myPassword1234!</code>	Password of your server account, as created in chapter "Prerequisites".
<code>credential=xxxxxxxxxx</code>	The credential string (API key), as described above.
<code>serverAddress=https://kassenspeicher.de/server/service</code>	This is the fix server address.
<code>syncInterval=600</code>	Optional, default = 0. Synchronization interval in seconds. E.g. 600 seconds = 10 minutes. Possible values: 0, empty or not specified: the sync runs only once. You must start the syncer by yourself. 0 < x < 30: not allowed. The minimum time is 30 seconds. >=30: The syncer runs in a infinite loop and performs the sync every x seconds.

Configuration for CSV output

The following configuration options control the CSV output (receiving server data). **The syncer will only receive such data which are specified here. If you don't specify "outputPath" at all, nothing will be received!**

Configuration	Comments
<code>outputPath=c:/temp/output;sd_productgroups</code> <code>outputPath=c:/temp/output;sd_products</code> <code>outputPath=c:/temp/output;sd_customergroups</code> <code>outputPath=c:/temp/output;sd_customers</code>	Specifies which server data (=table name, see appendix) shall be written to which directory. You can specify the same directory for all data or specify different directories for each data. The directory path can be absolute or relative.

First sync run

Now we are ready to receive the first data. Start the syncer (double-click or execute the JAR file). The syncer automatically calls the server and reads the server data. You should get a couple of CSV files generated. Dependant on the configuration "syncInterval" the syncer now runs once or forever (until you kill the process).

Everytime you change something in the Android cashbox system, a new CSV file will be written to your local PC (e.g. change the sales price of one or more products in Kasse Speedy and you will receive a corresponding CSV file with the changed product data).

Configuration for CSV input

The following configuration options control the CSV input (sending CSV data to the server):

Configuration	Comments
<code>inputPath=c:/temp/input/products;sd_products</code> <code>inputPath=c:/temp/input/customers;sd_customers</code>	Specifies which server data (=table name, see appendix) shall be read from which directory.
<code>inputPath=c:/temp/input/</code>	<u>Alternatively</u> you may specify 1 central directory for all data. In this case, the CSV file names must start with the table name (see appendix), e.g. file name „sd_products.csv“ or „sd_products_xxxxx.csv“.
<code>successfullyTransmitted=c:/temp/transmitted</code>	Successfully transmitted CSV files are moved to this directory.
<code>notTransmitted=c:/temp/failed</code>	In case the data transmission fails (due to wrong CSV data) the CSV file is moved to this directory. Please check the error log to find out the reason.
<code>removeAfter=1440</code>	Optional. Time in minutes. Successfully transmitted CSV files (which are moved to the „successfullyTransmitted“ directory) are automatically deleted after the specified time. Possible values: 0: files are never deleted. > 0: any positive value in minutes. If omitted, the default is 1440 minutes = 24h.

Advanced configuration

For fine-tuning you may add these configuration options:

Configuration	Comments
<code>accountPasswordB=bXlQYXNzd29yZDEyMzQh</code>	Optional, password of your server account base64 encoded . To be used alternatively to <code>accountPassword</code> . This is not a safe encryption, but avoids clear-text password reading. It's your choice.
<code>selectLimit=100</code>	Optional, default = 500. Valid range: 1-500. The syncer will read only the specified number of data entries from the server. The created CSV files will not be larger than the specified number.
<code>logMode=E</code>	Optional, default = E. Controls which (error or warning) messages are written to the log file. Possible values: E: only errors W: errors and warnings I: errors, warnings and informational messages D: debug mode, every message is written. Do not use this permanently!



Tips and tricks

CSV format

The CSV file must follow these conventions:

- Columns are separated by a semicolon ;
- Text fields must be surrounded by double quotes “
- Each data entry is a single text line followed by a line break (CR)
- The column sequence is not guaranteed! Do *not* rely on a certain column sequence. When parsing/reading CSV files better rely on the header row (first row). The column names are fix.

Full sync versus delta sync

The CSV syncer will only receive server data which are recently modified. Example: you change the sales price of 3 products in Kasse Speedy. As a consequence the CSV syncer will create a CSV file with 3 entries. You will *not* get all products. This is called “delta sync” – only changed data are transmitted.

The delta sync is controlled by timestamps which are stored the file “cache.txt”.

If you delete the file “cache.txt” you can enforce a “full sync”. Then *all* data are retrieved from the server.

First sync / full sync takes very long

If you run your first sync cycle (or if you enforce a “full sync” by deleting the “cache.txt” file), it may happen that it takes pretty much time to received all your server data.

This depends on 2 configuration settings: selectLimit and syncInterval.

Example: Imagine you have 1000 products on the server. You have configured selectLimit=100 and syncInterval=60. Then it will take $1000 / 100 * 60 = 600$ seconds = 10 minutes to receive all 1000 products.

Optional: transmitting data using timestamps

In each CSV file you may specify a column “ChangedDateTimeUTC”. Here you can indicate *when* the data set has been changed (independent on the transmission point in time). If you omit this column the CSV-Syncer will send the current timestamp (transmission point in time).

The server data are updated only if the given timestamp is *newer* than the server timestamp.

Configuration changes

The syncer will read the configuration file for each sync cycle. If you change the configuration, you don't have to stop and start the syncer. It will read the new configuration automatically.

Problems with Microsoft Excel

If you open CSV files with Microsoft Excel it may happen that numbers are crunched!

- Long numbers may be converted to a floating point number (exponential notation) like 1,2345E12.
- A leading zero may be discarded, e.g. for telephone numbers or ZIP codes.

These are problems of Microsoft Excel and are not CSV problems! Please open CSV data like this:

- Either use a plain-text editor, e.g. Windows Notepad.
- Or use another spreadsheet software like LibreOffice.
- Or use the data import assistant of Microsoft Excel.
- Or do handle the CSV file manually and read/parse the file programmatically (using a CSV parsing library).



Data Tables

Underlined field names are primary keys.

Master Data

Client applications have read/write access to all master data tables.

sd_customergroups

This table contains all defined customer groups. The external API grants read/write access.

Field name	Type	Description
<u>CustomerGroupID</u>	bigint(20)	Unique ID of the customer group
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
CustomerGroupText	varchar(100)	Description of the customer group
BgColor	varchar(10)	Background color, e.g. „ffa7b600“. <ul style="list-style-type: none"> - ff: transparency (always ff) - a7: value for red - b6: value for green - 00: value for blue
TextColor	varchar(10)	Text color, e.g. „ffa7b600“. <ul style="list-style-type: none"> - ff: transparency (always ff) - a7: value for red - b6: value for green - 00: value for blue
SortIndex	int(11)	Sort index, any number which is used for sorting (ascending order)
Status	tinyint(4)	Status: 0 = active, 1 = inactive, 2 = invisible, 3 = deleted
DiscountID	bigint(20)	Unique ID of discount (foreign key to sd_discounts). -1 = no discount
DisplayMode	tinyint(4)	Display/sort mode: <ul style="list-style-type: none"> • 0 = manually sorted • 1 = alph. sorted by Name1 • 2 = alph. sorted by Name2 • 3 = graphical display (Tischplan) • 4 = sorted by rating • 5 = alph. sorted by customer number
DisplaySettings	text	Data for graphical display (Tischplan)

Cashboxes	text	List of cashbox names (CR separated), on which this entity shall be displayed, empty = all cashboxes (default)
------------------	------	--

sd_customers

This table contains all defined customers (customer master). The external API grants read/write access.

Field name	Type	Description
CustomerID	bigint(20)	Unique ID of customer
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
CustomerNumber	varchar(100)	External number(s) of customer. Multiple numbers are separated by line breaks.
CustomerGroupID	bigint(20)	Unique ID of customer group (foreign key to sd_customergroups)
CustomerImagePath	varchar(100)	File path to customer image (local file path on client device)
CustomerTitle	varchar(50)	Title (e.g. Mr./Mrs)
CustomerAcademicTitle	varchar(50)	Academic title (e.g. „Dr.“)
CustomerName	varchar(100)	First name
CustomerName2	varchar(100)	Last name
CustomerAddress	varchar(100)	Street and house number
PostalCode	varchar(20)	Postal code
City	varchar(50)	City
Country	varchar(50)	Country
SortIndex	int(11)	Sort index, any number which is used for sorting (ascending order)
Status	tinyint(4)	Status: 0 = active, 1 = inactive, 2 = invisible, 3 = deleted
MemoText	varchar(500)	Internal memo text
DiscountID	bigint(20)	Unique ID of discount (foreign key to sd_discounts). -1 = no discount
CustomerDefault	tinyint(4)	Flag, if this customer is the default customer. 0=false, -1=true
PhoneNumberPrivate	varchar(50)	Private phone number

PhoneNumberOffice	varchar(50)	Office phone number
PhoneNumberMobile	varchar(50)	Mobile phone number
Email	varchar(50)	Email of customer
Homepage	varchar(100)	Homepage of customer
TaxNumber	varchar(50)	Tax number of customer
Birthday	date	Birthday of customer
Cashback	tinyint(4)	Flag, if customer uses cashback. 0=false, -1=true
PaymentMethods	varchar(100)	List of payment methods allowed for this customer (separated by ";")
Cashboxes	text	List of cashbox names (CR separated), on which this entity shall be displayed, empty = all cashboxes (default)

sd_discounts

This table contains all defined product groups. The external API grants read/write access.

Field name	Type	Description
<u>DiscountID</u>	bigint(20)	Unique ID of discount
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
DiscountText	varchar(100)	Description of the discount
DiscountPercentage	decimal(5,2)	Percentage of the discount
DiscountType	tinyint(4)	Type of the discount: 0 = visible discount (is printed) 1 = hidden discount (is not printed)
DiscountUsage	int(11)	Usage of the discount defines in which context the discount can be used: - 3 = product (groups) - 4 = customer (groups) - 8 = receipt as a whole - 16 = single receipt position The values can be combined at will, e.g. 4 + 8 = 12.
SortIndex	int(4)	Sort index, any number which is used for sorting (ascending order)
Status	tinyint(11)	Status: 0 = active, 1 = inactive, 2 = invisible, 3 = deleted
MemoText	varchar(500)	Internal memo text

sd_productgroups

This table contains all defined product groups. The external API grants read/write access.

Field name	Type	Description
ProductGroupID	bigint(20)	Unique ID of the product group
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
ProductGroupText	varchar(100)	Description of the product group
BgColor	varchar(10)	Background color, e.g. „ffa7b600“. <ul style="list-style-type: none"> - ff: transparency (always ff) - a7: value for red - b6: value for green - 00: value for blue
TextColor	varchar(10)	Text color, e.g. „ffa7b600“. <ul style="list-style-type: none"> - ff: transparency (always ff) - a7: value for red - b6: value for green - 00: value for blue
SortIndex	int(11)	Sort index, any number which is used for sorting (ascending order)
Status	tinyint(4)	Status: 0 = active, 1 = inactive, 2 = invisible, 3 = deleted
DiscountID	bigint(20)	Unique ID of discount (foreign key to sd_discounts). -1 = no discount
ParentProductGroupID	bigint(20)	ID of the parent product group, -1 = no parent product group (top level, default)
DisplayMode	tinyint(4)	Display/sort mode: <ul style="list-style-type: none"> • 0 = manually sorted • 1 = alph. sorted by product text • 2 = alph. sorted by product number • 3 = sorted by rating
Cashboxes	text	List of cashbox names (CR separated), on which this entity shall be displayed, empty = all cashboxes (default)

sd_products

This table contains all defined products. The external API grants read/write access.

Field name	Type	Description
ProductID	bigint(20)	Unique ID of the product
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change

ProductNumber	varchar(200)	External number(s) of the product. Multiple numbers are separated by line breaks.
ProductTextShort	varchar(100)	Short (internal) description text of the product. This text is used on the cashbox user interface.
ProductTextLong	varchar(300)	Long (external) description text of the product. This text is used on receipts and printing.
ProductGroupID	bigint(20)	Unique ID of the product group (foreign key to sd_productgroups)
SortIndex	int(11)	Sort index, any number which is used for sorting (ascending order)
PurchasePrice	decimal(13,6)	Purchase price of the product
SalesPrice	decimal(13,6)	Sales price of the product
DepositPrice	decimal(13,6)	Deposit price of the product
PrintVoucher	tinyint(4)	Flag, if vouchers shall be printed for this product. 0=false, -1=true
PrintDepositVoucher	tinyint(4)	Flag, if deposit vouchers shall be printed for this product. 0=false, -1=true
Status	tinyint(4)	Status: 0 = active, 1 = inactive, 2 = invisible, 3 = deleted
TaxPercentage	decimal(5,2)	Tax percentage
DiscountID	bigint(20)	Unique ID of discount (foreign key to sd_discounts). -1 = no discount
TaxPercentage2	decimal(5,2)	Tax percentage. Only used for alternative taxes (in house/to go)
BookingMethod	tinyint(4)	Behaviour when booking (see Speedy documentation): <ul style="list-style-type: none"> - 0 = direct, single - 1 = direct, merge - 2 = popup, single - 3 = popup, merge
ProductImagePath	varchar(100)	File path to product image (local file path on client device)
QuantityUnit	varchar(50)	Quantity unit, e.g. "Stk", "kg", etc. As defined by the user.
MemoText	varchar(300)	Internal memo text
BookingTextTemplates	varchar(300)	Booking text / variant texts (see example* below and Speedy documentation). Each value is separated by line breaks.

AlterSalesPrice	tinyint(4)	Boolean <ul style="list-style-type: none"> 0 = Sales price is not editable during booking 1 = Sales price is editable during booking
AlterText	tinyint(4)	Boolean <ul style="list-style-type: none"> 0 = Product text is not editable during booking 1 = Product text is editable during booking
AlterMemoText	tinyint(4)	Boolean <ul style="list-style-type: none"> 0 = Memo text is not editable during booking 1 = Memo text is editable during booking
QuantityDecimalPlaces	tinyint(4)	Number of decimal places for product quantities (0 – 6)
Rating	tinyint(4)	Rating, for favorites list (0 – 5)
ProductType	tinyint(4)	Type of product: <ul style="list-style-type: none"> 0 = normal product (default) 1 = inventory managed product 2 = coupon 3 = customer credit 4 = text/memo
Balances	text	List of balance ID's (CR separated). Contains currently exactly 1 balance ID, multiple ID's reserved for future. A balance may be a warehouse, coupon or customer credit.
Cashboxes	text	List of cashbox names (CR separated), on which this entity shall be displayed, empty = all cashboxes (default)

* Remarks to field "BookingTextTemplates": this field contains product variants, stored in CSV format.

Example:

```
"with extra cheese";0.5;-1;0;1;#ff000000;
"with French Fries";2.0;-1;0;1;#ff000000;
      1           2 3 4 5 6
```

CSV-position (column)	Example	Meaning	Consequences when booking a receipt
1	with French Fries	Text of variant	Variant text will be put to the receipt booking text (either „PosTextShort“ or “MemoText”, dependant on CSV column 4)
2	0.5	Price of variant	Variant price will be added to SalesPricePerUnit (multiplied by the booked

			quantity)
3	-1	-1 = no selection group 1 = selection group 1 2 = selection group 2 etc.	Variants of the same selection group can be booked alternately only (either-or).
4	0	0 = normal 1 = internal note only	0 = normal: variant text will be put to receipt field "PosTextShort" 1 = internal note: variant text will be put to receipt field "MemoText"
5	0	0 = can be booked only once 1 = can be booked multiple times	
6	#ff000000	Color of variant on cash register UI. Value corresponds to ARGB standard (alpha, red, green, blue) with # as prefix	

sd_balances

This table contains all defined "balances". The external API grants read/write access.

A "Balance" may be, dependant on the BalanceType, either a warehouse (dt. "Lager"), a coupon (dt. "Gutschein") or a customer credit (dt. "Kundenguthaben").

Field name	Type	Description
BalanceID	bigint(20)	Unique ID of the balance
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
BalanceType	tinyint(4)	The type of the balance: 0 = warehouse 1 = coupon 2 = customer credit
BalanceNumber	varchar(200)	External number of the balance. Dependant on the balance type it may be: <ul style="list-style-type: none"> warehouse number coupon number (not used for customer credits)
BalanceText	varchar(100)	Description text of the balance. Dependant on the balance type it may be: <ul style="list-style-type: none"> warehouse name coupon text (not used for customer credits)
Status	tinyint(4)	Status: 0 = active, 1 = inactive, 2 = invisible, 3 = deleted
SortIndex	int(11)	Sort index, any number which is used for sorting (ascending order)
MemoText	varchar(300)	Internal memo text. Currently used for warehouses only.
ParentBalanceID	bigint(20)	ID of the parent balance (-1 = no parent), if balances have a hierarchy. Currently not used!
CustomerID	bigint(20)	Unique ID of an assigned customer (-1 = no

		customer assigned). Used for customer credits only.
ImagePath	varchar(100)	File path to an image (local file path on client device). Used for warehouses only.
ValidFrom	datetime	Valid-from date. Currently not used.
ValidTo	datetime	Valid-to date. Used for coupons only.

The current balance value (warehouse stock, coupon value or value of the customer credit) is stored in table `sd_balancepos`. See chapter “transactional data” below.

sd_usergroups

This table contains all defined user groups. The external API grants read/write access.

Field name	Type	Description
UserGroupID	bigint(20)	Unique ID of the user group
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
UserGroupText	varchar(100)	Description text of the user group
SortIndex	int(11)	Sort index, any number which is used for sorting (ascending order)
Status	tinyint(4)	Status: 0 = active, 1 = inactive, 2 = invisible, 3 = deleted
MemoText	varchar(500)	Internal memo text.
IsAdmin	tinyint(4)	Boolean <ul style="list-style-type: none"> 0 = Normal user group with permissions as defined in table <code>sd_userpermissions</code>. -1 = Admin user group. This user group has <u>all</u> permissions. Permissions entries in <code>sd_userpermissions</code> are ignored.
Cashboxes	text	List of cashbox names (CR separated), on which this entity shall be displayed, empty = all cashboxes (default)

sd_userpermissions

This table contains all defined permissions of user groups. The external API grants read/write access. Remark: It is not possible to assign permissions to single users. Permissions are assigned to user groups always.

Field name	Type	Description
<u>UserGroupID</u>	bigint(20)	Unique ID of the user group, as defined in table sd_usergroups
<u>UserID</u>	bigint(20)	Reserved for future use. Currently -1 always.
<u>PermissionName</u>	varchar(100)	Name of the permission. To get a list of all possible permissions, please performs these steps: <ul style="list-style-type: none"> • Install the cashbox software on your Android device • Import the demo data (or define your own user groups) • Export the user group(s) • Open the export file as spreadsheet and check the permissions column
<u>ChangedDateTimeUTC</u>	datetime(3)	UTC timestamp of the last data change
<u>PermissionValue</u>	tinyint(4)	A value between 0 (no permission/forbidden) and 100 (full permission/allowed). Some permissions also allow intermediate values. To get the relevant values, please perform these steps: <ul style="list-style-type: none"> • Install the cashbox software on your Android device • Import the demo data (or define your own user groups) • Manipulate the user group permissions using the sliders. • Export the user group(s) • Open the export file as spreadsheet and check the permissions column • Repeat the export for other permission slider positions

sd_users

This table contains all defined users. The external API grants read/write access.

Field name	Type	Description
UserID	bigint(20)	Unique ID of the user
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
UserNumber	varchar(100)	External number(s) of the user. Multiple numbers are separated by line breaks.
UserName	varchar(100)	Description/name of the user
UserGroupID	bigint(20)	ID of the user group, as defined in table sd_usergroups.
SortIndex	int(11)	Sort index, any number which is used for sorting (ascending order)
Status	tinyint(4)	Status: 0 = active, 1 = inactive, 2 = invisible, 3 = deleted
MemoText	varchar(500)	Internal memo text.
Password	varchar(100)	Encrypted user password. It is recommended to set the password via the cashbox user interface.
UserImagePath	varchar(100)	File path to user image (local file path on client device)
Birthday	datetime	Reserved for future use.
Cashboxes	text	List of cashbox names (CR separated), on which this entity shall be displayed, empty = all cashboxes (default)

Transactional Data



Attention!

Client applications have **read-only** access to all transactional data tables. Due to financial regulations never try to alter these data in any way! Any attempt of data modification may be unveiled by fiscal authorities and may be penalized. The API prohibits such data modification.

sd_receipts

This table contains the receipt headers (1 row for each receipt). The external API grants read-only access.

Field name	Type	Description
<u>ReceiptID</u>	bigint(20)	Unique ID of the receipt
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
ReceiptDateTime	datetime(3)	Local timestamp of the receipt
ReceiptStatusID	tinyint(4)	Status of the receipt: 2=open, 3=payed, 4=cancelled, 5=deleted
CustomerID	bigint(20)	ID of the assigned customer
ClosingStatus	tinyint(4)	Closing status of the receipt (dt. Endabrechnung): 0=not closed, 1=closed
ReceiptNumber	varchar(50)	External receipt number (the number which is visible to the user and which is printed on the receipt)
PosCount	int(11)	Number of receipt positions
AmountSumDeposit	decimal(13,4)	Total amount of deposit (dt. Pfand)
AmountSumNet	decimal(13,4)	Total net amount (dt. Nettosumme)
AmountSumGross	decimal(13,4)	Total gross amount (dt. Bruttosumme)
PaymentMethod	varchar(50)	Payment method (dt. Zahlungsart) empty= cash (BAR) or any value defined by the user
AmountGiven	decimal(13,4)	Amount given by the customer
AmountChange	decimal(13,4)	Change amount (given back to the customer)
TaxMode	tinyint(4)	Tax mode („im Haus“ / „außer Haus“) 1 = tax value 1 is used (default) 2 = alternative tax value 2 is used
CertificateID	bigint(20)	ID of the certificate, which has been used for the digital signature
Signature	varchar(255)	Signature string of the digital signature

ChangedUserID	bigint(20)	Unique ID of the user (foreign key to sd_users), who changed the receipt at last
CashboxName	varchar(50)	Name of the cashbox, where the receipt belongs to.
SignatureStatus	tinyint(4)	Status of digital signature <ul style="list-style-type: none"> • -1 = none, unused, default • 1 = signature of start receipt • 2 = signature of subsequent receipts • 101 = signature error
InvoicePrintedQuantity	tinyint(4)	Counter for printed invoices
CustomerNumber	varchar(100)	Customer number which was used during receipt booking
MemoText	varchar(300)	Receipt memo text. Currently this text cannot be entered by the user. It is used by the system to annotate the receipt in special situations (receipt handover, etc.).
PaymentTransactionData	text	JSON string, contains transaction data of various non-cash payments, e.g. credit card payments
ReceiptType	tinyint(4)	Type of receipt: <ul style="list-style-type: none"> • 0 = normal sales receipt (default) • 2 = payment in/out (Einlage/Entnahme) • 10 = warehouse-in receipt • 11 = warehouse-out receipt • 12 = warehouse-move receipt • 13 = inventory receipt

sd_receiptpos

This table contains the receipt positions (0..n rows for each receipt). The external API grants read-only access.

Field name	Type	Description
<u>ReceiptID</u>	bigint(20)	Unique ID of the receipt (foreign key to sd_receipts)
<u>ReceiptPosID</u>	bigint(20)	ID of the receipt position, ID is unique within one receipt
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
ProductID	bigint(20)	Unique key of the product (foreign key to sd_products)
Quantity	decimal(13,6)	Booked quantity

SalesPricePerUnit	decimal(13,6)	Sales NET price for quantity 1 (dt. Netto-Einzelpreis)
DepositPricePerUnit	decimal(13,6)	Deposit NET price for quantity 1 (dt. Netto-Pfandeeinzelpreis)
VoucherPrintedQuantity	decimal(13,6)	Number of printed vouchers
DepositVoucherPrintedQuantity	decimal(13,6)	Number of printed deposit vouchers
TaxPercentage	decimal(5,2)	Tax percentage
PosType	tinyint(4)	Type of the receipt position: 0 = normal product position 2 = payment position without product reference (dt. Einlage/Entnahme) 3 = text position without product reference
SortIndex	int(11)	Sort index, currently not used
ProductGroupID	bigint(20)	Unique ID of product group (foreign key to sd_productgroups)
PosTextShort	varchar(300)	Text of receipt position (as booked by the user, normally product text)
Status	tinyint(4)	Status of the receipt position: 2=open, 3=payed, 4=cancelled, 5=deleted
ChangedUserID	bigint(20)	Unique ID of the user (foreign key to sd_users), who changed the receipt position at last
QuantityUnit	varchar(50)	Quantity unit, e.g. "Stk", "kg", etc. As defined by the user.
MemoText	varchar(300)	Internal memo text of the receipt position, as entered by the user
ProductType	tinyint(4)	See sd_products.ProductType
BalanceID	bigint(20)	ID of the balance object (either a warehouse, coupon or customer credit). See explanations for table sd_balances.

sd_receiptposdiscounts

This table contains discount information for receipt positions (0..n rows for each receipt position). The external API grants read-only access.

Be aware: The discount information is just for informational purposes, e.g. to print it on receipts. The actual discount amount is already considered in the receipt position (sd_receiptpos.SalesPricePerUnit is already reduced by the discount).

Field name	Type	Description
<u>ReceiptID</u>	bigint(20)	Unique ID of the receipt (foreign key to sd_receipts)
<u>ReceiptPosID</u>	bigint(20)	ID of the receipt position (foreign key to sd_receiptpositions)
<u>DiscountID</u>	bigint(20)	Unique ID of the discount (foreign key to sd_discounts)
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
DiscountText	varchar(100)	Text of the discount
DiscountPercentage	decimal(5,2)	Percentage of the discount
DiscountType	tinyint(4)	Type of the discount: <ul style="list-style-type: none"> 0 = visible discount (is printed) 1 = hidden discount (not printed)
Status	tinyint(4)	Status: <ul style="list-style-type: none"> 5 = deleted

sd_balancepos

This table contains the current value (quantity or amount) for a specific “balance”. The external API grants read-only access.

A “Balance” may be, dependant on the BalanceType, either a warehouse (dt. “Lager”), a coupon (dt. “Gutschein”) or a customer credit (dt. “Kundenguthaben”).

Field name	Type	Description
<u>BalanceID</u>	bigint(20)	Unique ID of the balance
<u>ProductID</u>	bigint(20)	Unique ID of the assigned product
<u>ProductVariantName</u>	varchar(50)	Name of the product variant (currently not used yet)
Quantity	decimal(13,6)	Current quantity of the specific product in the specific balance. <ul style="list-style-type: none"> warehouse: this is the current warehouse stock coupons: not relevant customer credit: not relevant The quantity <u>unit</u> (piece, kg, liter,...) must be retrieved from sd_products.QuantityUnit.
Amount	decimal(13,6)	Current amount/currency value of the specific product in the specific balance. <ul style="list-style-type: none"> warehouse: not relevant coupon: current value of the coupon customer credit: current value of the

customer credit

sd_closingruns

This table contains all closing runs (dt. Endabrechnungen / Z-Bons). The external API grants read-only access.

Field name	Type	Description
ClosingRunID	bigint(20)	Unique ID of closing run
ChangedDateTimeUTC	datetime(3)	UTC timestamp of the last data change
ClosingRunDateTime	datetime(3)	Local date/time of closing run creation
ClosingRunNumber	varchar(50)	External number of closing run
PreviousClosingRunID	bigint(20)	Unique ID of previous closing run (foreign key to sd_closingruns)
UserID	bigint(20)	Unique ID of user (foreign key to sd_users), who created the closing run
ReceiptIDFrom	bigint(20)	Unique ID of receipt (foreign key to sd_receipts), which is the first receipt of this closing run
ReceiptIDTo	bigint(20)	Unique ID of receipt (foreign key to sd_receipts), which is the last receipt of this closing run
MemoText	varchar(300)	Internal memo text as entered by the user during closing run creation
Currency	varchar(10)	Currency, e.g. „EUR“
Status	tinyint(4)	Status: 5 = deleted
CountSales	int(11)	Number of sales receipt positions
SumSalesNet	decimal(13,4)	Net amount of sales
SumSalesGross	decimal(13,4)	Gross amount of sales
CountSalesCash	int(11)	Number of CASH sales receipt positions (dt. BAR-Verkäufe)
SumSalesCashNet	decimal(13,4)	Net amount of CASH sales
SumSalesCashGross	decimal(13,4)	Gross amount of CASH sales
CountReturns	int(11)	Number of returns (quantity < 0)

SumReturnsNet	decimal(13,4)	Net amount of returns
SumReturnsGross	decimal(13,4)	Gross amount of returns
CountReturnsCash	int(11)	Number of CASH returns (quantity < 0)
SumReturnsCashNet	decimal(13,4)	Net amount of CASH returns
SumReturnsCashGross	decimal(13,4)	Gross amount of CASH returns
CountCancelations	int(11)	Number of cancelled receipt positions
SumCancelationsNet	decimal(13,4)	Net amount of cancelled receipt positions
SumCancelationsGross	decimal(13,4)	Gross amount of cancelled receipt positions
SumTax	decimal(13,4)	Total sum of taxes
SumTaxCash	decimal(13,4)	Total sum of taxes which were booked CASH
SumDeposit	decimal(13,4)	Total sum of deposit
SumDepositCash	decimal(13,4)	Total sum of deposit which were booked CASH
SumChangeCash	decimal(13,4)	Change amount which were booked CASH
CountPaymentsIn	int(11)	Number of inlays (dt. Einlagen)
SumPaymentsIn	decimal(13,4)	Total sum of inlays (dt. Einlagen)
CountPaymentsOut	int(11)	Number of withdrawals (dt. Entnahmen)
SumPaymentsOut	decimal(13,4)	Total sum of withdrawals (dt. Entnahmen)
SumFinalPaymentOut	decimal(13,4)	Amount of final withdrawal (dt. Entnahme/Abschöpfung) during creation of closing run
FinalPaymentOutReceipt ID	bigint(20)	Unique ID of the „finalPaymentOut“ receipt (foreign key to sd_receipts)
BalanceCalc	decimal(13,4)	Total balance amount (as calculated)
BalanceSet	decimal(13,4)	Total balance amount (as counted and entered by the user)
BalanceNew	decimal(13,4)	Total balance amount for the next day

CountDepositInvoiced	int(11)	Number of invoiced deposit receipt positions (quantity > 0)
SumDepositInvoicedNet	decimal(13,4)	Total net sum of invoiced deposit
SumDepositInvoicedGross	decimal(13,4)	Total gross sum of invoiced deposit
CountDepositReturned	int(11)	Number of returned deposit receipt positions (quantity < 0)
SumDepositReturnedNet	decimal(13,4)	Total net sum of returned deposit
SumDepositReturnedGross	decimal(13,4)	Total gross sum of returned deposit
CashboxName	varchar(50)	Name of the cashbox, where the closing run belongs to
CountTip	int(11)	Number of tip bookings
SumTip	decimal(13,4)	Total sum of tips
SumTipCash	decimal(13,4)	Total sum of tips which were booked CASH
MoneyCountProtocol	varchar(600)	Money count protocol (number of counted coins and

Version history

Version 1.0 (27.01.2021)

First version of this document



Urheberrecht, Haftungsausschluss, Impressum

Urheberrecht Das Werk einschließlich aller Inhalte ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Nachdruck oder Reproduktion (auch auszugsweise) in irgendeiner Form (Druck, Fotokopie, Scan oder anderes Verfahren) sowie die Einspeicherung, Verarbeitung, Vervielfältigung und Verbreitung mit Hilfe elektronischer Systeme jeglicher Art, gesamt oder auszugsweise, ist ohne ausdrückliche schriftliche Genehmigung des Autors untersagt. Alle Übersetzungsrechte vorbehalten. Genannte Firmennamen (wie Microsoft oder Epson) und genannte Produktnamen (wie Microsoft Excel oder Epson TM-T88) sind Marken oder eingetragene Marken ihrer jeweiligen Eigentümer.

Haftungsausschluss Die Benutzung des Programmes (Android-App) und dieses Handbuches sowie die Umsetzung der darin enthaltenen Informationen erfolgt ausdrücklich auf eigenes Risiko. Haftungsansprüche gegen den Autor für Schäden materieller oder ideeller Art, die durch die Nutzung oder Nichtnutzung der Informationen bzw. durch die Nutzung fehlerhafter und/oder unvollständiger Informationen verursacht wurden, sind grundsätzlich ausgeschlossen. Wir übernehmen keine Gewähr für die Fehlerfreiheit der Software oder der Dokumentation. Für durch Benutzung unseres Programmes entstehende Schäden oder Folgeschäden übernehmen wir keine Haftung. Insbesondere haften wir nicht für den Verlust von Daten. Rechts- und Schadenersatzansprüche sind daher ausgeschlossen. Das Werk inklusive aller Inhalte wurde unter größter Sorgfalt erarbeitet. Der Autor übernimmt jedoch keine Gewähr für die Aktualität, Korrektheit, Vollständigkeit und Qualität der bereitgestellten Informationen. Druckfehler und Falschinformationen können nicht vollständig ausgeschlossen werden. Der Autor übernimmt keine Haftung für die Aktualität, Richtigkeit und Vollständigkeit der Inhalte des Buches, ebenso nicht für Druckfehler. Es kann keine juristische Verantwortung sowie Haftung in irgendeiner Form für fehlerhafte Angaben und daraus entstandenen Folgen vom Autor übernommen werden. Für die Inhalte von den in diesem Buch abgedruckten Internetseiten sind ausschließlich die Betreiber der jeweiligen Internetseiten verantwortlich. Der Autor hat keinen Einfluss auf Gestaltung und Inhalte fremder Internetseiten. Der Autor distanziert sich daher von allen fremden Inhalten. Zum Zeitpunkt der Verlinkung waren keinerlei illegale Inhalte auf den Webseiten vorhanden.

Impressum © Copyright 2021 by:
mtMax GmbH
Bannholzweg 13
D - 69168 Wiesloch
Germany

