

KASSA.AT

API-Dokumentation

v1.0.0

KASSA.AT
Faxonline GmbH

E-Mail: info@kassa.at
Telefon: 0800 802104

Dokument-Version: 1.0.0
API-Version: 1
Veröffentlicht: 12.01.2021

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1 Generelle Informationen	4
1.1 Kontakt	4
1.2 Server	4
2 Ressourcen	5
2.1 /authentications	5
2.1.1 /create	5
2.1.2 /authenticate	7
2.2 /warehouses	8
2.2.1 /index	8
2.3 /inventories	9
2.3.1 /index	9
2.3.2 /stock (GET)	11
2.3.3 /stock (POST)	13
2.4 /movements	15
2.4.1 /create	15
3 Error Codes	18
3.1 1000er	18
3.2 2000er	19
3.3 3000er	20

Inhaltsverzeichnis

4	Globale Fehler	21
4.1	Status-Code 400: Bad Request (Parametervalidierung fehlgeschlagen)	21
4.2	Status-Code 401: Unauthorized (Kein Token übermittelt)	21
4.3	Status-Code 401: Unauthorized (Falscher Token übermittelt)	22
4.4	Status-Code 401: Unauthorized (Falsche oder keine Subdomain)	22
5	Glossar	23
5.1	Parameter-Glossar	23
6	Was die Zukunft bringt	24

1 Generelle Informationen

1.1 Kontakt

Bei Fragen bezüglich der API kontaktieren Sie uns am Besten mit einer Email an unsere Adresse info@kassa.at oder Sie rufen uns unter der Telefonnummer **0800 802104** an.

1.2 Server

Sandbox-Server:

Zur Zeit steht kein Sandbox-Server zur Verfügung.

Live-Server:

Der Live Server ist unter <https://kassa.at/api/v1/...> zu erreichen. Die spezifischen URLs sind in der Dokumentation jedes Aufrufs vermerkt.

2 Ressourcen

2.1 /authentications

Die authentication Aufrufe sind zur Erstellung der Token da.

2.1.1 /create

HTTP-Methode: `POST`

URL: `https://dev.kassa.at/api/v1/authentications/`

Parameter:

<code>company_dns</code>	<code>string</code>		Die Subdomain der Firma
<code>username</code>	<code>string</code>		Der Username eines Admins
<code>password</code>	<code>string</code>		Das Passwort eines Admins
<code>description</code>	<code>string</code>	<code>optional</code>	Welches System nutzt den Token

Beispiel CURL-Befehl:

```
curl -X POST https://dev.kassa.at/api/v1/authentications/  
-H 'Content-Type: application/json'  
-d '{"company_dns":"testfirma", "username":"admin", "password":"test-password",  
"description":"system XY"}'
```

Antworten:

Status-Code 201: Created

```
{  
  "status": 201,  
  "key": "cf96552b-187c8503d2d6ec08"  
}
```

Status-Code 400: Bad Request (Firmen-DNS nicht gefunden)

```
{
  "status": 400,
  "error_code": 3000,
  "error": "Object not found!",
  "details": [
    {
      "error_code": 2000,
      "error": "Company missing!",
      "error_message": "No company with given dns_name!"
    }
  ]
}
```

Status-Code 400: Bad Request (Logindaten fehlerhaft)

```
{
  "status": 400,
  "error_code": 2008,
  "error": "User invalid!",
  "details": [
    {
      "error_code": 2002,
      "error": "User invalid!",
      "error_message": "Username or password invalid or not an admin!"
    }
  ]
}
```

2.1.2 /authenticate

Info: Dieser Aufruf ist der Einzige, der mit Views arbeitet. Im Falle, dass die Login Daten nicht auf einer externen Seite eingegeben werden sollen, steht Ihnen folgende Methode zur Verfügung:

Erstellen Sie ein Formular, das den User auf unsere Webseite leitet. Dort gibt der User seine Login Daten ein und der Token wird generiert. Der User wird dann mit dem Token zurück zu Ihrem Service geschickt, wo der Token schließlich von Ihnen ausgelesen werden kann.

HTTP-Methode: `POST`

URL: `https://dev.kassa.at/api/v1/authentications/authentication`

Parameter:

<code>return_path</code>	<code>string</code>		Die Adresse, an die der Token geschickt werden soll
<code>description</code>	<code>string</code>	<code>optional</code>	Welches System nutzt den Token

Beispiel HTML-Form:

```
<form action="https://dev.kassa.at/api/v1/authentications/authenticate"
method="get">
  <input name="return_path" type="hidden" value="https://foo-bar.at/receive-token">
  <input name="description" type="hidden" value="System XY">
  <button type="submit" name="Submit">
    Verbinden
  </button>
</form>
```

Nachdem der Token von unserer Seite aus erstellt wurde, wird der user an die zuvor im Parameter "return_path" spezifizierte Seite weitergeleitet. Auf dieser Seite muss dann der Token aus den mitgelieferten Parametern extrahiert werden.

Bezogen, auf den oberen HTML-Snippet schaut der endgültige Link, auf den am Ende gewechselt wird so aus: "<https://foo-bar.at/receive-token?key=4fa3d741-ab62f0a3d5c79e11>"

2.2 /warehouses

Die Warehouses sind die Lager eines Accounts und werden verwendet um Lagerstände anzuzeigen. Eine Firma kann mehrere Lager haben.

2.2.1 /index

HTTP-Methode: `GET`

URL: `https://<company_dns>.kassa.at/api/v1/warehouses`

Parameter:

<code>deleted</code>	<code>mixed</code>	<code>optional</code>	Zeigt nur gelöschte / nicht-gelöschte Lager an
----------------------	--------------------	-----------------------	--

Parameter-Filterung:

Parameter	Mögliche Werte	Typ
<code>deleted</code>	<code>"true" / "false" / "TRUE" / "FALSE"</code> <code>true / false</code>	<code>string</code> <code>boolean</code>

Beispiel CURL-Befehl:

```
curl -X GET http://<company_dns>.kassa.at/api/v1/warehouses
-H 'Content-Type: application/json'
-H 'authorization: <api_token>'
-d '{"deleted":false}'
```

Antworten:

Status-Code 200: OK

```
{
  "status": 200,
  "details": [
    {
      "id": 1,
      "description": "Standard Lager",
      "deleted": false,
      "created_at": {
        "format": "DD.MM.YYYY hh:mm:ss",
        "localized": "01.01.2021 15:35"
      },
      "updated_at": {
        "format": "DD.MM.YYYY hh:mm:ss",
        "localized": "12.01.2021 09:45"
      }
    }
  ],
  "length": 1
}
```


2.3 /inventories

Inventories sind eingelagerte Artikel. Ein Artikel in einem Lager ist ein inventory. Ein Artikel kann auch in mehreren Lagern eingelagert sein und ein Lager kann auch mehrere Artikel eingelagert haben.

2.3.1 /index

HTTP-Methode: GET

URL: `https://<company_dns>.kassa.at/api/v1/warehouses/<n>/inventories`

Parameter:

article_number	string	optional	Zeigt nur inventories mit dieser Artikelnummer an
article	integer	Optional	Zeigt nur inventories mit dieser Artikel-ID an
active	mixed	optional	Zeigt nur aktive /nicht aktive inventories an
inspect_stock	mixed	optional Default: false	Zeige zusätzlich zu den inventories auch die eingelagerte Anzahl an

Parameter-Filterung:

Parameter	Mögliche Werte	Typ
active	"true" / "false" / "TRUE" / "FALSE" true / false	string boolean
inspect_stock	"true" / "false" / "TRUE" / "FALSE" true / false	string boolean

Beispiel CURL-Befehl:

```
curl -X GET http://<company_dns>.kassa.at/api/v1/warehouses/<n>/inventories
-H 'Content-Type: application/json'
-H 'authorization: <api_token>'
-d '{"active":true}'
```

Antworten:**Status-Code 200: OK**

```
{
  "status": 200,
  "details": [
    {
      "id": 3,
      "warehouse": {
        "id": 1,
        "description": "Standard Lager",
        "deleted": false,
        "created_at": {
          "format": "DD.MM.YYYY hh:mm:ss",
          "localized": "01.01.2020 08:32"
        },
        "updated_at": {
          "format": "DD.MM.YYYY hh:mm:ss",
          "localized": "01.01.2020 12:52"
        }
      },
      "article": {
        "id": 5,
        "company_article_id": 5,
        "line_item_name": "Dies ist ein Testartikel",
        "default_quantity": "1.0",
        "sales_tax_percentage": "20.0",
        "gross_amount_per_item": "34.99",
        "article_number": null,
        "product_name": null,
        "deleted": false
      },
      "active": true,
      "stock": "Send parameter 'display_stock' to inspect the stock-amount",
      "lower_limit": 5,
      "created_at": {
        "format": "DD.MM.YYYY hh:mm:ss",
        "localized": "05.07.2018 08:35"
      },
      "updated_at": {
        "format": "DD.MM.YYYY hh:mm:ss",
        "localized": "05.07.2018 08:35"
      }
    }
  ],
  "length": 1
}
```

Status-Code 400: Bad Request (Warehouse-ID nicht gefunden)

```
{
  "status": 400,
  "error_code": 3000,
  "error": " Object not found!",
  "details": [
    {
      "error_code": 3008,
      "error": " Object not found!",
      "error_message": " No warehouse found with given parameter!"
    }
  ]
}
```

2.3.2 /stock (GET)

HTTP-Methode: GET

URL: `https://<company_dns>.kassa.at/api/v1/warehouses/<n>/inventories/<n>/stock`

Beispiel CURL-Befehl:

```
curl -X GET
http://<company_dns>.kassa.at/api/v1/warehouses/<n>/inventories/<n>/stock
-H 'Content-Type: application/json'
-H 'authorization: <api_token>'
```

Antworten:

Status-Code 200: OK

```
{
  "status": 200,
  "data": [
    {
      "stock": "-4.0",
      "inventory": {
        "id": 15,
        "article": {
          "id": 5,
          "line_item_name": "Dies ist ein Testartikel",
          "article_number": null
        },
        "warehouse": {
          "id": 1,
          "description": "Standard Lager"
        },
        "warning_at": 5,
        "created_at": {
          "format": "DD.MM.YYYY hh:mm:ss",
          "localized": "05.07.2018 11:25"
        },
        "updated_at": {
          "format": "DD.MM.YYYY hh:mm:ss",
          "localized": "05.07.2018 11:25"
        }
      }
    }
  ]
}
```

Status-Code 400: Bad Request (Warehouse-ID nicht gefunden)

```
{
  status: 400,
  error_code: 3000,
  error: "Object not found!",
  details: [
    {
      error_code: 3008,
      error: "Object not found!",
      error_message: "No warehouse found with given parameter!"
    }
  ]
}
```

Status-Code 400: Bad Request (Inventory-ID nicht gefunden)

```
{
  status: 400,
  error_code: 3000,
  error: "Object not found!",
  details: [
    {
      error_code: 3007,
      error: "Object not found!",
      error_message: " No inventory found with given parameter!"
    }
  ]
}
```

2.3.3 /stock (POST)

HTTP-Methode: POST

URL: `https://<company_dns>.kassa.at/api/v1/warehouses/<n>/inventories/<n>/stock`

Beispiel CURL-Befehl:

```
curl -X POST
http://<company_dns>.kassa.at/api/v1/warehouses/<n>/inventories/<n>/stock
-H 'Content-Type: application/json'
-H 'authorization: <api_token>'
-d '{"quantity":13.42, "text":"Lagerstandsprüfung: Mai"}
```

Parameter:

quantity	float		Die absolute Anzahl, wie oft der Artikel im Lager ist.
text	string	Optional	Gibt eine Beschreibung für das update an

Antworten:

Status-Code 200: OK

```
{
  "status": 200,
  "data": [
    {
      "inventory": {
        "id": 15,
        "article": {
          "id": 2,
          "line_item_name": "Dies ist ein Testartikel ",
          "article_number": null
        },
        "warehouse": {
          "id": 1,
          "description": "Standard Lager"
        },
        "warning_at": 5,
        "created_at": {
          "format": "DD.MM.YYYY hh:mm:ss",
          "localized": "05.07.2018 11:25"
        },
        "updated_at": {
          "format": "DD.MM.YYYY hh:mm:ss",
          "localized": "05.07.2018 11:25"
        }
      },
      "old_stock": "-4.0",
      "new_stock": "13.42"
    }
  ]
}
```

Status-Code 400: Bad Request (Warehouse-ID nicht gefunden)

```
{
  status: 400,
  error_code: 3000,
  error: "Object not found!",
  details: [
    {
      error_code: 3008,
      error: "Object not found!",
      error_message: "No warehouse found with given parameter!"
    }
  ]
}
```

Status-Code 400: Bad Request (Inventory-ID nicht gefunden)

```
{
  status: 400,
  error_code: 3000,
  error: "Object not found!",
  details: [
    {
      error_code: 3007,
      error: "Object not found!",
      error_message: " No inventory found with given parameter!"
    }
  ]
}
```

2.4 /movements

Movements sind die einzelnen Transaktionsnachweise jedes inventories.

Wenn ein Update des Lagerstandes gemacht wird, wird ein neues Movement erstellt.

2.4.1 /create

HTTP-Methode: `POST`

URL: `https://<company_dns>.kassa.at/api/v1/warehouses/<n>/inventories/<n>/movements`

Beispiel CURL-Befehl:

```
curl -X POST
http://<company_dns>.kassa.at/api/v1/warehouses/<n>/inventories/<n>/movements
-H 'Content-Type: application/json'
-H 'authorization: <api_token>'
-d '{"quantity":-2, "text":"Ich hatte hunger"}
```

Parameter:

quantity	float		Die absolute Anzahl, wie oft der Artikel im Lager ist.
text	string	optional	Gibt eine Beschreibung für das update an
type*	string	optional	Lagerstand erhöhen oder verringern

*: Es gibt zwei Wege ein movement als Abzug zu machen. Entweder eine negative quantity, oder eine positive quantity und "decrease" als type.

Parameter-Filterung:

Parameter	Mögliche Werte	Typ
type	"increase" / "decrease"	string

Antworten:**Status-Code 200: OK**

```
{
  "status": 201,
  "data": [
    {
      "id": 3010,
      "inventory": {
        "id": 15,
        "article": {
          "id": 2,
          "line_item_name": "Dies ist ein Testartikel"
        },
        "warehouse": {
          "id": 1,
          "description": "Standard Lager"
        },
        "old_stock": "13.42",
        "new_stock": "11.42"
      },
      "quantity": "-2.0",
      "description": "Ich hatte hunger",
      "created_at": {
        "format": "DD.MM.YYYY hh:mm:ss",
        "localized": "11.01.2021 11:49"
      },
      "updated_at": {
        "format": "DD.MM.YYYY hh:mm:ss",
        "localized": "11.01.2021 11:49"
      }
    }
  ]
}
```

Status-Code 400: Bad Request (Warehouse-ID nicht gefunden)

```
{
  status: 400,
  error_code: 3000,
  error: "Object not found!",
  details: [
    {
      error_code: 3008,
      error: "Object not found!",
      error_message: "No warehouse found with given parameter!"
    }
  ]
}
```


Status-Code 400: Bad Request (Inventory-ID nicht gefunden)

```
{
  status: 400,
  error_code: 3000,
  error: "Object not found!",
  details: [
    {
      error_code: 3007,
      error: "Object not found!",
      error_message: " No inventory found with given parameter!"
    }
  ]
}
```

3 Error Codes

3.1 1000er

Die Fehler der 1000er Reihe beschreiben Parametervalidierungsfehler.

error_code	error	error_message
1000	Parametervalidation failed!	Parametervalidation failed!
1002	Parameter is not set!	Parameter 'company_dns' is not set!
1003	Parameter is not set!	Parameter 'username' is not set!
1004	Parameter is not set!	Parameter 'password' is not set!
1007	Parameter is invalid!	Parameter 'company_dns' includes forbidden characters (a-z A-Z 0-9 - _)!
1008	Parameter is invalid!	Parameter 'company_dns' doesn't match expected length (3..32)!
1010	Parameter is invalid!	Parameter 'username' doesn't match expected length (3..64)!
1011	Parameter is invalid!	Parameter 'password' doesn't match expected length (8..72)!
1014	Parameter is not set!	Parameter 'quantity' is not set!
1028	Parameter is invalid!	Parameter 'quantity' is not the right format. Check the documentation to see the right format!
1032	Parameter is invalid!	Parameter 'article' must be an integer (0-9)!
1035	Parameter is invalid!	Parameter 'deleted' must be 'true' or 'false'!
1036	Parameter is not set!	Parameter 'warehouse_id' is not set!
1037	Parameter is invalid!	Parameter 'warehouse_id' must be an integer (0-9)!

Kapitel: 3 Error Codes

1038	Parameter is invalid!	Parameter 'active' must be 'true' or 'false'!
1039	Parameter is not set!	Parameter 'inventory_id' is not set!
1040	Parameter is invalid!	Parameter 'inventory_id' must be an integer (0-9)!
1041	Parameter is invalid!	Parameter 'text' doesn't match expected length (0..100)!
1042	Parameter is invalid!	Parameter 'type' must be 'increase' or 'decrease'!
1043	Parameter is invalid!	Parameter 'inspect_stock' must be true or false!
1044	Parameter is invalid!	Parameter 'description' doesn't match expected length (<250)!
1045	Parameter is invalid!	Parameter 'article_number' doesn't match expected length (0..60)!
1046	Parameter is invalid!	Parameter 'quantity' can't be 0!

3.2 2000er

Die Fehler dieser Nummern beschreiben einen Fehler bei der authentifizierung

error_code	error	error_message
2000	Company missing!	No company with given dns_name!
2002	User invalid!	Username or password invalid or not an admin!
2003	Unauthorized!	No token given!
2004	Unauthorized!	Token is invalid!
2008	User invalid!	User invalid!
2009	Unauthorized!	Unauthorized!

3.3 3000er

Fehler der 3000er Serie zeigen an, dass durch eingegebene Daten kein Ergebnis gefunden wurde, zB.: eine id in der URL zeigt auf keinen existierenden Eintrag.

error_code	error	error_message
3000	Object not found!	Object not found!
3007	Object not found!	No inventory found with given parameter!
3008	Object not found!	No warehouse found with given parameter!

4 Globale Fehler

Diese Fehler können bei jedem Aufruf, wenn nicht anders beschrieben, auftreten.

4.1 Status-Code 400: Bad Request (Parametervalidierung fehlgeschlagen)

```
{
  status: 400,
  error_code: 1000,
  error: "Parametervalidation failed!",
  details: [
    {
      error_code: 10xx,
      error: "Parameter is not set!",
      error_message: "Parameter '<parametername>' is not set!"
    },
    {
      error_code: 10xx,
      error: "Parameter is invalid!",
      error_message: "Parameter '<parametername>' doesn't match expected length (<10)!"
    },
  ]
}
```

Diese Fehler sind die häufigsten. Sie treten auf wenn übermittelte Daten nicht akzeptiert werden.

Die `error_codes` sind aus der [1000er Reihe](#). Der `error`-Part ist dabei eine grobe beschreibung des Problems. Die `error_message` spezifiziert das vorhandene Problem dann genauer.

Wenn mehrere Fehler bei den Parametern auftreten werden diese wie im Beispiel gezeigt mit Kommas getrennt.

4.2 Status-Code 401: Unauthorized (Kein Token übermittelt)

```
{
  status: 401,
  error_code: 2009,
  error: "Unauthorized!",
  details: [
    {
      error_code: 2003,
      error: "Unauthorized!",
      error_message: "No token given!"
    }
  ]
}
```

4.3 Status-Code 401: Unauthorized (Falscher Token übermittelt)

```
{
  status: 401,
  error_code: 2009,
  error: "Unauthorized!",
  details: [
    {
      error_code: 2004,
      error: "Unauthorized!",
      error_message: "Token is invalid!"
    }
  ]
}
```

4.4 Status-Code 401: Unauthorized (Falsche oder keine Subdomain)

```
{
  "status": 401,
  "error_code": 2001,
  "error": "Company missing!",
  "details": [
    {
      "error_code": 2001,
      "error": "Company missing!",
      "error_message": "No company in subdomain!"
    }
  ]
}
```

Jeder Aufruf (ausgenommen Tokenerstellung) benötigt eine Subdomain durch die wir die Firma feststellen können. Sollte keine Subdomain gegeben sein oder die Subdomain schlichtweg falsch sein, wird dieser Fehler zurückgegeben.

5 Glossar

5.1 Parameter-Glossar

<code><company_dns></code>	Die Subdomain der Firma. Diese wird bei Erstellung der Firma bekanntgegeben und kann (momentan nur über die Webseite, nicht per API) in den Firmeneinstellungen eingesehen und verändert werden.
<code><api_token></code>	Der generierte Token muss immer mitgegeben werden. (Tokengenerierung finden Sie hier)
<code><n></code>	Bezeichnet die ID des Objekts, dass in der Adresse davor steht. Bsp <code>/api/v1/warehouses/<n>/inventories</code> . Hier ist <code><n></code> die ID von warehouse, in dem nach inventories gesucht werden soll.

6 Was die Zukunft bringt

Neue Features:

- Datumsformate auch als Datumstyp übergeben
- Firmen erstellen / auslesen / bearbeiten
- Kassen erstellen / auslesen / bearbeiten
- Benutzer erstellen / auslesen / bearbeiten
- Rechnungen erstellen / auslesen / bearbeiten

Fixes:

- Datums-Format-String Sekunde rauslöschen
- Returns Integers in integers und floats in Floats zurück geben