Meldescheine buchen

```
POST  import/meldeschein/buchen/
```

Dieser Aufruf erzeugt einen oder mehrere elektronischen Meldeschein im System.

Allgemein

Es ist möglich mehrere Meldescheine auf einmal zu buchen, indem weitere Objekte dem `meldescheine` Array hinzugefügt werden.

Parameter

Name	Datentyp	Verwendung
meta	object	Meta-Objekt
params	object	Optionale Parameter
meldescheine	array	Array mit Meldescheindaten
meldescheine[]	object	Die zu buchenden Daten des Meldescheins
meldescheine[].personen	array	Auflistung aller Gäste
meldescheine[].ms_type	string	Immer mit dem Wert "ems" zu befüllen

Pro Person

Name	Datentyp	Verwendung
meldescheine[].personen[].tarif_id	int	Die Tarif-ID des Gasts
meldescheine[].personen[].gast_type	string	Gast-Art (`person`, `gruppe`)

Name	Datentyp	Verwendung
meldescheine[].personen[].arrival_date	date	Anreisedatum des Gasts
meldescheine[].personen[].departure_date	date	Abreisedatum des Gasts

Einzel-Person

Name	Datentyp	Verwendung
meldescheine[].personen[].*	string	Weitere definierte Felder. (nach Definition)

> Die Definition der Felder hängt von der Gemeinde ab und können über den Aufruf `/get_current_user` abgefragt werden.

Reisegruppe

Name	Datentyp	Verwendung
meldescheine[].personen[].anzahl	int	Anzahl der Gäste

> Wenn das Feld meldescheine[].personen[].anzahl größer 1 ist ,wird der Meldeschein als Gruppenmeldeschein verbucht. Eventuell übergebene Einzelinformationen für den Gast (Vorname,Nachname etc.) werden nicht gespeichert.

Optionale Parameter (`params`)

Name	Datentyp	Beschreibung
append_pdf	bool	Für bei der Response jedem Meldeschein-Objekt den Parameter "pdf" .

`params.append_pdf`

Dieser Schalter ermöglicht es, jedem Meldeschein-Objekt direkt das PDF anzufügen.
In der Server-Antwort wird das Feld `pdf` angefügt, welches den Base64-kodierten Meldescheine enthält.

Das aktivieren dieses Schalters gilt als Druckvorgang und wird entsprechend protokolliert - siehe Erklärung hier.

Beispiel

```
{
  "meta": {
    "gemeinde": 1,
    "requestId": "123-456-789",
    "idempotent": true,
    "objekt": 2049
  },
  "params": {
    "append_pdf": true
  },
    ....
 }
```

```
{
    "meta": {
        "gemeinde": 1,
        "requestId": "123-456-789",
        "idempotent": true,
        "objekt": 2049,
        "timestamp": "2020-12-18T11:51:41.154820",
        "user": {
            "id": 38,
            "username": "DemoV",
            "alias": "Lisa Mustermann"
        }
    },
    "status": "success",
    "meldescheine": [
        {
            "meldeschein": {
                "id": 28,
              //....
            },
            "id": 28,
```

```
            "pdf": "JVBERi0xLjMKMSAwIG9iago8PAov....."
        }
    ]
}
```

Server Antwort

Die Serverantwort gibt im Erfolgsfall ein Array aus `meldeschein` Objekten mit der dem gesamten Meldeschein-Objekt zurück.

> Der Anwort-Aufbau entspricht dem aus dem Meldeschein aktuaklisieren /get Aufruf

> Die Reihenfolge der Personen ist zwischen Anfrage und Antwort immer gleichbleibend. Somit können Sie die die ID's der Personen in Ihrem System für spätere Aktualisierungen mitführen

Response:

JSON


```
{
  "meta": {
    "requestId": "5ca0b885-11e3-4abd-9e60-cb8265ea9899",
    "timestamp": "2019-10-24T14:06:10.673840",
    "user": {
      "id": 14,
      "username": "api_demo",
      "first_name": "API",
      "last_name": "Demo",
      "email": "api_demo@apidemo.de",
      "is_active": true,
      "user_type": "beherberger",
      "alias": "API Demo",
      "last_login": "2019-10-24T11:35:18.015947+02:00"
    },
    "gemeinde": 1
  },
  "status": "success",
  "meldescheine": [
```

```
    {
      "id": 396,
      //.....
      "personen": [
        {
          "id":3334,
          "tarif_id": 1,

          "Gast_Vorname": "Max",
          "Gast_Name": "Mustermann",
          "arrival_date": "2019-01-01",
          "departure_date": "2019-01-15"
        }
      ]
    }
  ]
}
```

Beispiel-Buchungen

Beispiele für verschiedene Szenarien

Personen-Meldeschein

Folgende Beispiele zeigen jeweils die Buchung eines Scheins. Da die Rückantwort des Servers ist unabhängig des Typs immer gleich.

Request:

JSON

```
{
  "meta": {
    "vermieter": 1,
    "gemeinde": 1,
    "objekt": 1
  },
  "meldescheine": [
    {
      "input_id": "test-1",
      "personen": [
        {
          "gast_type": "person",
          "tarif_id": 1,
          "Gast_Vorname": "TEST_BF",
          "arrival_date": "2019-07-14",
          "departure_date": "2019-07-17"
        }
      ]
    }
  ]
}
```


XML

Gruppen-Meldeschein

Request:

JSON

```
{
  "meta": {
    "vermieter": 1,
    "gemeinde": 1,
    "objekt": 1
  },
  "meldescheine": [
    {
      "input_id": "test-1",
      "personen": [
        {
          "gast_type": "gruppe",
          "anzahl": 5,
          "tarif_id": 1,
          "arrival_date": "2019-07-14",
          "departure_date": "2019-07-17"
        }
      ]
    }
  ]
}
```

XML

Code-Beispiele

Folgendes Beispiel authentifiziert die Anfrage und bucht einen Meldeschein

Python 3

```
# This example uses the "requests" library for HTTP requests
(https://requests.readthedocs.io/en/latest/)
# You can install it via
# $ pip3 install requests
import requests
payload = {
    'username':'api_demo',
    'password':'demodemo',
```

```
}

r = requests.post('https://apiv2.meldescheine.de/api/token/',data=payload)
server_response_json = r.json()



jwt_token = server_response_json.get('token') # Aus dem server_response objekt den String-Wert
aus "token" extrahieren

# Jetzt buchen
meldeschein_data = {} # Leer für Test
headers = {
 'Authorization':  "JWT %s" % jwt_token
}
r =
requests.post("https://apiv2.meldescheine.de/api/import/meldeschein/buchen/",data=meldeschein_
data,headers=headers)
```

Revision #14
Created 28 July 2020 11:26:50 by .Admin
Updated 23 March 2023 09:08:27 by .Admin