

Meldescheine aktualisieren

Dokumentation unvollständig.

Allgemein

Dieser Aufruf aktualisiert einen existierenden elektronischen Meldeschein im System.

Es kann pro Aufruf immer nur ein (1) Meldeschein aktualisiert werden.
Wenn im unten erläuterten Array mehrere Meldescheine aufgeführt sind, wird der erste Meldeschein (Index:0) aktualisiert.

Existierende Meldescheine müssen immer mindestens einen Gast besitzen. Es ist nicht möglich einen Meldeschein zu "leeren", indem alle Gäste gelöscht werden.

Beschränkungen

Ein Meldeschein kann nur verändert werden, solange sich sein Status in "Entwurf" oder "gedruckt" befindet. Ist ein Meldeschein abgereist, sind keine weiteren Änderungen möglich.

"abgereist" ist wie folgt definiert: Alle Gäste des Meldescheins sind abgereist. Ein automatischer Prozess auf System-Ebene prüft alle aktuellen Meldescheine und Gäste anhand ihres Abreisedatums. Sind alle Gäste abgereist, so erhält der Meldeschein den Status "abgereist". Der Meldeschein gilt somit für die Gemeinde als "bereit zur Abrechnung."

Änderungen an diesen Meldescheinen können nur über die Gemeinde angefordert werden.

Parameter

```
POST import/meldeschein/aktualisieren/
```

Name	Datentyp	Verwendung
------	----------	------------

meta	object	Meta-Objekt
meldescheine	array	Array mit Meldescheindaten
meldescheine[]	object	Die zu buchenden Daten des Meldescheins
meldescheine[].personen	array	Auflistung aller Gäste

Der Aufbau der Anfrage entspricht exakt des Aufrufs "buchen" mit dem Unterschied, dass im Meldeschein-Objekt die ID aus dem Aufruf "buchen" mit übergeben werden muss, um den zu aktualisierenden Meldeschein eindeutig zu identifizieren.

Zum aktualisieren von Gästen muss das Feld "id" mit übergeben werden

Server Antwort

Die "aktualisieren" Antwort enthält den aktualisierten Meldeschein als vollständiges Objekt. Die Antwort ist unabhängig der ausgeführten Aktion immer im gleichen Format und spiegelt den aktuellen Status des Meldescheins im System wider.

Da nur derzeit ein Meldeschein aktualisiert werden kann, befindet sich der aktualisierte Meldeschein im Objekt `response.meldescheine[0]`

Die Antwort gibt den aktuellen Status des Meldescheins im System zurück - hier können Sie ggfls. prüfen, ob Änderungen erfolgreich übernommen wurden.

```
{
  "meta": {
    "requestId": "afc8673a-2a4c-11eb-b5a3-c04a00212a69",
    "timestamp": "2020-11-19T10:50:41.996029",
    "user": {
      "id": 38,
      "username": "DemoV",
      "alias": "Lisa Mustermann"
    }
  },
}
```

```
"gemeinde": 1
},
"status": "success",
"response": {
  "meldescheine": [
    {
      "id": 688,
      "personen": [
        {
          "id": 2015,
          "tarif_id": 1,
          "meldeschein_id": 688,
          "gast_type": "person",
          "created": "2020-11-19T10:48:47.228504+01:00",
          "modified": "2020-11-19T10:50:42.311202+01:00",
          "anzahl": 1,
          "arrival_date": "2019-07-14",
          "departure_date": "2019-07-17",
          "CSID": "6b59af3c-2a4c-11eb-aa07-c04a00212a69-WLclSDown",
          "kurtaxe_calc": 6.0,
          "Gast_Vorname": "aktualisiert?",
          "Gast_Name": null,
          "Gast_Geburtsdatum": null,
          "Gast_Strasse": null,
          "Gast_Postleitzahl": null,
          "Gast_Wohnort": null,
          "Gast_Land": null,
          "Gast_Email": null,
          "Gast_Ausweisnummer": null,
          "Gast_Staatsangehoerigkeit": null,
          "address_id": null,
          "address_lat": null,
          "address_lng": null,
          "address_label": null
        },
        {
          "id": 2016,
          "tarif_id": 1,
          "meldeschein_id": 688,
```

```
"gast_type": "person",
"created": "2020-11-19T10:48:47.571441+01:00",
"modified": null,
"anzahl": 1,
"arrival_date": "2019-07-14",
"departure_date": "2019-07-17",
"CSID": "6b59af3c-2a4c-11eb-aa07-c04a00212a69-WLclSDown",
"kurtaxe_calc": 6.0,
"Gast_Vorname": "TEST_BF_2",
"Gast_Name": null,
"Gast_Geburtsdatum": null,
"Gast_Strasse": null,
"Gast_Postleitzahl": null,
"Gast_Wohnort": null,
"Gast_Land": null,
"Gast_Email": null,
"Gast_Ausweisnummer": null,
"Gast_Staatsangehoerigkeit": null,
"address_id": null,
"address_lat": null,
"address_lng": null,
"address_label": null
},
{
  "id": 2017,
  "tarif_id": 1,
  "meldeschein_id": 688,
  "gast_type": "person",
  "created": "2020-11-19T10:48:47.611475+01:00",
  "modified": null,
  "anzahl": 1,
  "arrival_date": "2019-07-14",
  "departure_date": "2019-07-17",
  "CSID": "6b59af3c-2a4c-11eb-aa07-c04a00212a69-WLclSDown",
  "kurtaxe_calc": 6.0,
  "Gast_Vorname": "TEST_BF_3",
  "Gast_Name": null,
  "Gast_Geburtsdatum": null,
  "Gast_Strasse": null,
```

```

        "Gast_Postleitzahl": null,
        "Gast_Wohnort": null,
        "Gast_Land": null,
        "Gast_Email": null,
        "Gast_Ausweisnummer": null,
        "Gast_Staatsangehoerigkeit": null,
        "address_id": null,
        "address_lat": null,
        "address_lng": null,
        "address_label": null
    },
    {
        "id": 2018,
        "tarif_id": 1,
        "meldeschein_id": 688,
        "gast_type": "person",
        "created": "2020-11-19T10:48:47.655957+01:00",
        "modified": null,
        "anzahl": 1,
        "arrival_date": "2019-07-14",
        "departure_date": "2019-07-17",
        "CSID": "6b59af3c-2a4c-11eb-aa07-c04a00212a69-WLclSDowcn",
        "kurtaxe_calc": 6.0,
        "Gast_Vorname": "TEST_BF_4",
        "Gast_Name": null,
        "Gast_Geburtsdatum": null,
        "Gast_Strasse": null,
        "Gast_Postleitzahl": null,
        "Gast_Wohnort": null,
        "Gast_Land": null,
        "Gast_Email": null,
        "Gast_Ausweisnummer": null,
        "Gast_Staatsangehoerigkeit": null,
        "address_id": null,
        "address_lat": null,
        "address_lng": null,
        "address_label": null
    }
],

```

```
"travel_span": {
  "arrival_date": "2019-07-14T00:00:00",
  "departure_date": "2019-07-17T00:00:00",
  "travel_days": 3,
  "uebernachtungen_sum": 12
},
"objekt": {
  "id": 2049,
  "beherberger_id": 13455,
  "beherberger": {
    "firma": "Ferienhof Müller",
    "vorname": "Melissa",
    "nachname": "Müller",
    "comment": "<br>",
    "debitor_nr": "08154711",
    "alias": "Ferienhof Müller (08154711)",
    "id": 13455,
    "ansprechpartner": 37,
    "gemeinde_id": 1
  },
  "beherberger_alias": "Ferienhof Müller (08154711)",
  "ortsteil": {
    "id": 1,
    "alias": "Standard",
    "gemeinde": 1
  },
  "created": "2020-03-30T08:34:03.052153+02:00",
  "modified": "2020-07-21T13:16:53.520759+02:00",
  "anzahl_betten": 4,
  "alias": "Haus Sonne",
  "strasse": "Edgar-Anstett-Straße 4",
  "plz": "66978",
  "ort": "Merzalben",
  "tarifzone": 1,
  "unterkuntsart": 1
},
"meldescheinnummer": "688",
"objekt_id": 2049,
"beherberger_id": 13455,
```

```

    "verwendete_kurtaxe": 24.0,
    "creation_user": {
      "id": 38,
      "username": "DemoV",
      "alias": "Lisa Mustermann"
    },
    "precheck_errors": [],
    "state_display": "Entwurf",
    "faktura_date": null,
    "CSID": "6b59af3c-2a4c-11eb-aa07-c04a00212a69-WLclSDown",
    "Form_Id": null,
    "BatchNo": null,
    "BatchRDate": "2020-11-19",
    "BatchPgDta": null,
    "BatchTrack": null,
    "ms_type": "ems",
    "state": "draft",
    "created": "2020-11-19T00:00:00+01:00",
    "modified": null,
    "user_note": null,
    "is_storniert": false,
    "source": null,
    "storno_bestatigt": false,
    "kurtaxe_ist": null,
    "manueller_betrag": 0.0,
    "kz_manueller_betrag": false,
    "kurtaxe_calc": 24.0,
    "stat_alter_kind_1": null,
    "stat_alter_kind_2": null,
    "stat_alter_kind_3": null,
    "stat_alter_kind_4": null,
    "invoice": null
  }
]
}
}

```

Anlegen von zusätzlichen Gästen

Um dem Meldeschein weitere Gäste hinzuzufügen, können diese ohne Angabe des Feldes "id" in das "personen" Array mit übergeben werden.

Für das anlegen von zusätzlichen Gästen gelten die Regeln wie beim anlegen eines Meldescheins. Die Felder "tarif_id" , "arrival_date" sowie "departure_date" sind zwingend notwendig. Zusätzliche Felder analog zum buchen Aufruf und der eingerichteten Gemeinde.

Beispiel-Request: Anlegen eines zusätzlichen Gastes:

"Max Mustermann"

Tarif-ID "1" (Erwachsener)

Reisezeitraum: 15.10.2020 - 16.10.2020

in Meldeschein 650

```
{
  "meta": {
    "objekt": 2080,
    "gemeinde": 1,
    "requestId": "TEST"
  },
  "meldescheine": [
    {
      "id": 650,
      "personen": [
        {
          "arrival_date": "2020-10-15",
          "departure_date": "2020-10-16",
          "tarif_id": 1,
          "Gast_Vorname": "Max",
          "Gast_Name": "Mustermann"
        }
      ]
    }
  ]
}
```

Aktualisieren von vorhandenen Gästen

Um Gäste zu aktualisieren, da sich z.B. Reisedaten geändert haben ist es notwendig, die "id" des Gastes mit zu übergeben.

Diese kann z.B. über den "[get-data](#)" Aufruf abgerufen werden oder kann beim buchen der Meldescheine sofort gespeichert werden.

Beispiel: Ändern des im vorherigen Beispiel "Anlegen von zusätzlichen Gästen" erzeugten Gastes "Max Mustermann":.

In diesem Fall wird der Reisezeitraum geändert: 16.10.2020 bis 19.10.2020 sowie der Nachname des Gastes.

Es werden nur Felder aktualisiert, die übergeben werden und im System vorhanden sind. Achten Sie daher bitte darauf, nur die notwendigen Felder zu übergeben und achten Sie auf korrekte Feldbezeichnungen

Zu beachten die ID "1945" des Gastes im unteren Beispiel.

- Wird die ID angegeben: Gast aktualisieren
- Keine Angabe der ID: Gast anlegen

```
{
  "meta": {
    "objekt": 2080,
    "gemeinde": 1,
    "requestId": "TEST"
  },
  "meldescheine": [
    {
      "id": 650,
      "personen": [
        {
          "id": 1945, //<----- ID des Gastes (z.B. über /get-data abgerufen
          "arrival_date": "2020-10-16",
          "departure_date": "2020-10-18",
          "tarif_id": 1,
          "Gast_Vorname": "Max",
          "Gast_Name": "Mustermann ( verändert )"
        }
      ]
    }
  ]
}
```

Löschen von vorhandenen Gästen

Hinweis zum aktualisieren von Gästen:

Bitte verwenden Sie immer den Ablauf "aktualisieren" um Gäste zu verändern. Bitte löschen Sie nicht alle Gäste und legen diese neu an.

Dies ist technisch in einem kombinierten Aufruf (s.u.) zwar möglich - jedoch erschwert dies die Nachvollziehbarkeit auf System-Ebene, da zu jedem Gast eine Historie angelegt wird.

Gäste können gelöscht werden, in dem diese mit ihrer ID sowie dem Feld "delete:true" übergeben werden.

Beispiel-Request: Löschen des Gastes mit id 1937 aus Meldeschein mit ID 651

```
{
  "meta": {
    "objekt": 2080,
    "gemeinde": 1,
    "requestId": "TEST"
  },
  "meldescheine": [
    {
      "id": 651,
      "personen": [
        {
          "id": 1937,
          "delete": true //<----- Gibt an, dass der Gast beim Aufruf gelöscht werden soll
        }
      ]
    }
  ]
}
```

Wenn Sie versuchen einen Gast zu löschen, der nicht dem Schein zugeordnet ist, erhalten Sie eine entsprechende Fehlermeldung

Verschiedene Gast-Aktionen in einem Aufruf

Es ist möglich die oben beschriebenen Aktionen in einem Aufruf zu kombinieren:

Beispiel:

Eine Kombination aus den og. Beispielen

- Gast **1945** aktualisieren
- Gast **1937** löschen
- Einen neuen Gast anlegen

```
{
  "meta": {
    "objekt": 2080,
    "gemeinde": 1,
    "requestId": "TEST"
  },
  "meldescheine": [
    {
      "id": 650,
      "personen": [
        {
          "id": 1945,
          "arrival_date": "2020-10-16",
          "departure_date": "2020-10-18",
          "tarif_id": 1,
          "Gast_Vorname": "Max",
          "Gast_Name": "Mustermann ( verändert )"
        },
        {
          "id": 1937,
          "delete": true
        },
        {
          "arrival_date": "2020-10-15",
          "departure_date": "2020-10-16",
```

```
    "tarif_id": 1,  
    "Gast_Vorname": "Max",  
    "Gast_Name": "Mustermann"  
  }  
}  
]  
}
```

Revision #14

Created 16 September 2020 08:24:31 by .Admin

Updated 19 November 2020 09:55:33 by .Admin