

Verwendung der PayJoe®-API zum externen Upload von Belegen und Zahlungen

Allgemeine Informationen

Um über die PayJoe®-API Belege bzw. Zahlungen hochladen zu können, müssen Sie zunächst einen entsprechenden [Zugang anlegen](#), bei dessen Zugangsart Sie API_Belege bzw. API_Zahlungen auswählen. Nach dem Sie den Zugang gespeichert und neu geöffnet haben, sehen Sie die benötigten Zugangsdaten für die API-Verwendung:

- **UserName:** Ihre E-Mail-Adresse, mit der Sie sich auch sonst bei PayJoe® anmelden
- **API-Key:** Zugangsschlüssel für die Authentifizierung gegenüber der API
- **ZugangID:** Die ID dieses Zugangs, in den die Daten hochgeladen werden sollen

Wie Sie an diese Informationen gelangen, sehen Sie in unserem [Onlinehandbuch](#) in „Zugangsdaten eines Zahlungs- oder Belegzugangs anpassen“.

Mit diesen Informationen können Sie nun POST-Requests an die folgenden URLs richten:

<https://api.payjoe.de/belegupload>

<https://api.payjoe.de/zahlungsupload>

Als Content-Type wird JSON erwartet, setzen Sie also im Header des Aufrufs den Tag:

```
Content-Type: application/json
```

Upload von Belegen

Die Information, ob ein Feld zwingend benötigt wird (m = muss / o = optional) und die maximale Länge der String-Inhalte stehen jeweils als Kommentar hinter den Feldern.

Jeweils eines der folgende Objekte muss im Body des Requests enthalten sein (hier als c#-Klasse dargestellt):

```
public class BelegUpload
{
    public string UserName { get; set; } // m - 100
    public string APIKey { get; set; } // m - 100
    public int ZugangID { get; set; } // m
    public PayJoeBeleg[] Belege { get; set; } // m
}
```

Darin sind die oben genannten Zugangsdaten zu setzen sowie ein Array von PayJoeBeleg -Objekten zu übermitteln.

Feldinformationen PayJoe® Belege

Folgende Inhalte können als PayJoeBeleg übertragen werden:

```
public class PayJoeBeleg
{
    public int BelegBerichtID { get; set; } // o
    public string BelegIDExtern { get; set; } // o - 30
    public string BelegMandantenRefID { get; set; } // o - 50
    public DateTime Belegdatum { get; set; } // m
    public BelegTyp Belegtyp { get; set; } // m
    public double BelegBetrag { get; set; } // m
    public Waehrung BelegWaehrung { get; set; } // m
    public string ZahlungsArt { get; set; } // o - 50
    public string BelegVorname { get; set; } // o - 100
    public string BelegNachname { get; set; } // o - 100
    public string BelegFirma { get; set; } // o - 100
    public string BelegExterneBestellNr { get; set; } // o - 40
    public string BelegKundenNr { get; set; } // o - 30
    public string BelegDebitorenNr { get; set; } // o - 30
    public string BelegEmail { get; set; } // o - 100
    public string BelegNummer { get; set; } // m - 30
    public string BelegReferenz1 { get; set; } // o - 200
    public string BelegReferenz2 { get; set; } // o - 200
    public string BelegReferenz3 { get; set; } // o - 200
    public string BelegReferenz4 { get; set; } // o - 200
    public string BelegReferenz5 { get; set; } // o - 200
    public string BelegReferenz6 { get; set; } // o - 200
    public string BelegReferenz7 { get; set; } // o - 200
    public string BelegReferenz8 { get; set; } // o - 200
    public string BelegReferenz9 { get; set; } // o - 200
    public string BelegReferenz10 { get; set; } // o - 200
}
```

Dabei ist Belegtyp = 0 für Rechnung und = 1 für Gutschrift.

Als Waehrung wird der internationale numerische Währungs-Code verwendet: Euro ist dabei beispielsweise 978, US-Dollar 840 und das Britische Pfund 826. Weitere Informationen zur Waehrung finden Sie [hier](#).

JSON-Beispiel Beleg

Der Inhalt einer JSON-Nachricht sieht dann beispielsweise für zwei Belege folgendermaßen aus:

```
{
  "UserName": "kunde@payjoe.de",
  "APIKey": "i5YStP:=k[!w",
  "ZugangID": 818,
  "Belege": [{
    "BelegBerichtID": 23,
    "BelegIDExtern": "8711",
    "BelegMandantenRefID": "F8",
    "Belegdatum": "2015-12-07T13:17:22.2426324+01:00",
    "Belegtyp": 0,
    "BelegBetrag": 29.99,
    "BelegWaehrung": 978,
    "ZahlungsArt": "Rechnungskauf",
    "BelegVorname": "Max",
    "BelegNachname": "Mustermann",
    "BelegFirma": null,
    "BelegExterneBestellNr": "479873",
    "BelegKundenNr": "C637732",
    "BelegDebitorenNr": "24338",
    "BelegEmail": "max.mustermann@t-online.de",
    "BelegNummer": "53427",
    "BelegReferenz1": null,
    "BelegReferenz2": null,
    "BelegReferenz3": null,
    "BelegReferenz4": null,
    "BelegReferenz5": null,
    "BelegReferenz6": null,
    "BelegReferenz7": null,
  }
  ]
}
```

```
"BelegReferenz8":null,
"BelegReferenz9":null,
"BelegReferenz10":null
},{
"BelegBerichtID":23,
"BelegIDExtern":"8712",
"BelegMandantenRefID":"F8",
"Belegdatum":"2015-12-07T15:47:02.2120334+01:00",
"Belegtyp":0,
"BelegBetrag":79.5,
"BelegWaehrung":978,
"ZahlungsArt":"PayPal",
"BelegVorname":"Monika",
"BelegNachname":"Mustermann",
"BelegFirma":null,
"BelegExterneBestellNr":"479874",
"BelegKundenNr":"C637711",
"BelegDebitorenNr":"24334",
"BelegEmail":"monika.mustermann@gmx.net",
"BelegNummer":"53453",
"BelegReferenz1":null,
"BelegReferenz2":null,
"BelegReferenz3":null,
"BelegReferenz4":null,
"BelegReferenz5":null,
"BelegReferenz6":null,
"BelegReferenz7":null,
"BelegReferenz8":null,
"BelegReferenz9":null,
"BelegReferenz10":null
}]
}
```

Erfolgsmeldung

Als Antwort bekommen Sie im Erfolgsfall untenstehende Objekte als JSON zurückgeliefert. Das Feld LetztesBuchungsdatum enthält das neueste Datum eines Belegs oder einer Zahlung, das in der Datenbank abgelegt ist. Auf diese Art können Sie den Upload in kleinere Pakete zerteilen. Es wird empfohlen nicht mehr als 500 Belege mit einem Aufruf zu übertragen.

```
public class BelegUploadReturn
{
    public bool Erfolgreich { get; set; }
    public DateTime LetztesBuchungsdatum { get; set; }
}
```

Hier der Inhalt der Antwort bei erfolgreichem Beleg-Upload:

```
{
    "Erfolgreich":true,
    "LetztesBuchungsdatum":"2018-01-07T14:50:55.26"
}
```

Upload von Zahlungen

Die Information, ob ein Feld zwingend benötigt wird (m = muss / o = optional) und die maximale Länge der String-Inhalte stehen jeweils als Kommentar hinter den Feldern.

Jeweils eines der folgende Objekte muss im Body des Requests enthalten sein (hier als c#-Klasse dargestellt):

```
public class ZahlungsUpload
{
    public string UserName { get; set; } // m - 100
    public string APIKey { get; set; } // m - 100
    public int ZugangID { get; set; } // m
    public PayJoeZahlung[] Zahlungen { get; set; } // m
}
```

Darin sind die oben genannten Zugangsdaten zu setzen sowie ein Array von PayJoeZahlung-Objekten zu übermitteln.

Feldinformationen für PayJoe® Zahlungen

Folgende Inhalte können als PayJoeZahlung übertragen werden:

```
public class PayJoeZahlung
{
    public int BerichtID { get; set; } // o
    public string UniqueIdentifizier { get; set; } // m - 80
    public DateTime Buchungsdatum { get; set; } // m
    public DateTime BerichtEndedatum { get; set; } // o
    public DateTime BerichtDepositdatum { get; set; } // o
    public DateTime Valutadatum { get; set; } // m
    public Waehrung Waehrung { get; set; } // m
    public double Nettobetrag { get; set; } // o
    public Waehrung Nettowaehrung { get; set; } // o
    public double Bruttobetrag { get; set; } // m
    public Waehrung Bruttowaehrung { get; set; } // m
    public double GebuehrBetrag { get; set; } // o
    public Waehrung GebuehrWaehrung { get; set; } // o
    public string BankleitzahlZahlender { get; set; } // o - 10
    public string BIC { get; set; } // o - 11
    public string KontonummerZahlender { get; set; } // o - 20
    public string IBAN { get; set; } // o - 34
    public string NameZahlender1 { get; set; } // o - 100
    public string NameZahlender2 { get; set; } // o - 100
    public string Belegnummer { get; set; } // o - 30
    public string Bestellnummer { get; set; } // o - 30
    public string TransaktionsID { get; set; } // o - 100
    public string Kundennummer { get; set; } // o - 30
    public string Referenz1 { get; set; } // o - 50
    public string Referenz2 { get; set; } // o - 50
    public string Referenz3 { get; set; } // o - 50
    public string Referenz4 { get; set; } // o - 50
    public string Marktplatz { get; set; } // o - 50
    public NeZahlungsStatus Zahlungsstatus { get; set; } // o
}
```

```

public string Zahlungstyp { get; set; } // o - 50
public bool NichtSaldorelevant { get; set; } // o
public string Buchungstext { get; set; } // o - 50
}

```

Dabei steht Zahlungsstatus = 0 für unbekannt und = 1 für bezahlt.

Als Waehrung wird der internationale numerische Währungs-Code verwendet: Euro ist dabei beispielsweise 978, US-Dollar 840 und das Britische Pfund 826. Weitere Informationen zur Waehrung finden Sie [hier](#).

JSON Beispiel Zahlung

Der Inhalt einer JSON-Nachricht sieht dann beispielsweise für eine Zahlung folgendermaßen aus:

```

{
  "UserName": "kunde@payjoe.de",
  "APIKey": "3k5TStP?=k(!w",
  "ZugangID": 5697,
  "ZahlungIn": [
    {
      "BerichtID": 45,
      "UniqueIdentifizier": "1T46DG0575BX16SSD",
      "Buchungsdatum": "2018-01-05T14:25:55.2426324+01:00",
      "BerichtEndedatum": "2018-01-07T14:25:55.2426324+01:00",
      „BerichtDepositdatum“: "2018-01-09T14:25:55.2426324+01:00",
      „Valutadatum“: "2018-01-09T14:25:55.2426324+01:00",
      "Waehrung": 978,
      "NettoBetrag": "18.00",
      "NettoWaehrung": 978,
      " BruttoBetrag ": "20.00",
      " BruttoWaehrung ": 978,
      "GebuehrBetrag ": "2.00",
      "GebuehrWaehrung ": 978,
      "BankleitzahlZahlender ": "50070024",
      "BIC ": "DEUTDEDBFRA",
      "KontonummerZahlender ": "0123456789",
      "IBAN ": "DE5850070024012346789",
      "NameZahlender1 ": "monika.mustermann@gmx.net",
    }
  ]
}

```

```

        "NameZahlender2":"Monika Mustermann",
        "Belegnummer":"6948593",
        "Bestellnummer":"BN-654698",
        "TransaktionsID":"1T46DG0575BX16SSD",
        "Kundennummer":"2014",
        "Referenz1":"T-Shirt Doppelpack",
        "Referenz2":"aus Angebot 2365",
        "Referenz3":null,
        "Referenz4":null,
        "Marktplatz":"Amazon.fr",
        "Zahlungsstatus":1,
        "Zahlungstyp":"Payment",
        "NichtSaldorelevant":0,
        "Buchungstext":"Completed"
    ]]
}

```

Erfolgsmeldung

Als Antwort bekommen Sie im Erfolgsfall untenstehende Objekte als JSON zurückgeliefert. Das Feld `LetztesBuchungsdatum` enthält das neueste Datum einer Zahlung, das in der Datenbank abgelegt ist. Auf diese Art können Sie den Upload in kleinere Pakete zerteilen. Es wird empfohlen nicht mehr als 500 Zahlungen mit einem Aufruf zu übertragen.

```

public class ZahlungsUploadReturn
{
    public bool Erfolgreich { get; set; }
    public DateTime LetztesZahlungsdatum { get; set; }
}

```

Hier der Inhalt der Antwort bei erfolgreichem Zahlungs-Upload:

```

{
    "Erfolgreich":true,
    "LetztesZahlungsdatum":"2018-01-07T14:50:55.26"
}

```