



# *PHP Bibliothek für ratenkauf by easyCredit*

*Anleitung zur Nutzung der PHP Bibliothek*

## Inhaltsverzeichnis

1. Übersicht .....	4
1.1 Funktionsübersicht .....	4
2. Voraussetzung .....	4
2.1 Kompatibilität .....	4
3. Einbindung und Nutzung .....	4
3.1 Standard Rückgabewert.....	4
3.2 Instanziierung des EasyCreditClients .....	5
3.2.1 Parameter .....	5
3.3 Initialisierung eines Vorgangs .....	5
3.3.1 Parameter .....	5
3.3.2 Rückgabewert.....	6
3.4 Aktualisieren eines Vorgangs.....	6
3.4.1 Parameter .....	6
3.4.2 Rückgabewert.....	6
3.5 Durchführen einer Entscheidung .....	7
3.5.1 Parameter .....	7
3.5.2 Rückgabewert.....	7
3.6 Abfrage der Entscheidung.....	7
3.6.1 Parameter .....	7
3.6.2 Rückgabewert.....	7
3.7 Prüfung einer mTAN .....	7
3.7.1 Parameter .....	7
3.7.2 Rückgabewert.....	8
3.8 Erneutes Versenden einer mTAN.....	8
3.8.1 Parameter .....	8
3.8.2 Rückgabewert.....	8
3.9 Bestätigen eines Vorgangs.....	8
3.9.1 Parameter .....	8
3.9.2 Rückgabewert.....	8
3.10 Abruf der Rechtstexte.....	8
3.10.1 Parameter.....	9
3.10.2 Rückgabewert .....	9
3.11 Abruf der gespeicherten Daten .....	9
3.11.1 Parameter.....	9
3.11.2 Rückgabewert .....	9
3.12 Abruf der Finanzierungsdaten .....	9
3.12.1 Parameter.....	9

3.12.2	Rückgabewert .....	9
3.13	Abruf einer Beispiel-Finanzierung.....	10
3.13.1	Parameter.....	10
3.13.2	Rückgabewert .....	10
3.14	Abruf des günstigsten Ratenplans .....	10
3.14.1	Parameter.....	10
3.14.2	Rückgabewert .....	10
3.15	Abruf des Restbetrages der Ankaufobergrenze.....	10
3.15.1	Parameter.....	10
3.15.2	Rückgabewert .....	11
3.16	Prüfen ob Zahlart für aktuelle Bestellung zur Verfügung steht .....	11
3.16.1	Parameter.....	11
3.16.2	Rückgabewert .....	11

ÄNDERUNGSHISTORIE

<i>Datum</i>	<i>Autor</i>	<i>Änderung</i>
15.01.2019	<a href="mailto:support@senbyte.com">support@senbyte.com</a>	INITIALE VERSION

## 1. Übersicht

Die PHP Bibliothek für ratenkauf by easyCredit erleichtert die technische Integration der Zahlart ratenkauf by easyCredit in einen Onlineshop. Dieses Dokument beschreibt die Nutzung der Bibliothek.

### 1.1 Funktionsübersicht

Die Bibliothek unterstützt die Integrationsarten „Payment Page“ und „Service Integration“ der Zahlart und bietet den folgenden Funktionsumfang (abhängig von der Integrationsart):

- Initialisierung eines neuen Vorgangs
- Aktualisierung eines bestehenden Vorgangs
- Durchführen einer Entscheidung zu einem Vorgang (angenommen, abgelehnt)
- Abfrage der Entscheidung zu einem Vorgang
- Prüfung einer mTAN
- Erneutes Versenden einer mTAN
- Bestätigen (Abschluß) eines Vorgangs
- Abruf der Rechtstexte
- Abruf der gespeicherten Daten eines Vorgangs
- Abruf der Finanzierungsdaten eines Vorgangs
- Abruf einer Beispiel-Finanzierung (Modellrechnung)
- Abruf des günstigsten Ratenplans
- Abruf des Restbetrages der Ankaufobergrenze für eine Webshop

## 2. Voraussetzung

Für die Nutzung der Bibliothek ist neben PHP die Installation von Curl auf ihrem Server erforderlich.

### 2.1 Kompatibilität

Die PHP Bibliothek wurde auf PHP 7 entwickelt und unterstützt die PHP Versionen ab 7.0.x.

## 3. Einbindung und Nutzung

Die Hauptklasse zur Nutzung ist „*EasyCreditClient*“. Fügen Sie die Datei „*EasyCreditClient.php*“ aus dem Hauptverzeichnis der Bibliothek ihrem Autoloader des Projektes zu oder importieren diese Datei mittels „*require\_once*“ in Ihr Skript.

### 3.1 Standard Rückgabewert

Alle Funktionen des EasyCreditClients liefern als Rückgabewert entweder ein Objekt der Klasse *EasyCreditClientAbstractResult* oder eine Erweiterung dieser Klasse zurück. Somit sind in jeder Antwort folgende Attribute enthalten:

- `$statusCode`: der http Status Code der REST api der Teambank
- `$error`: Flag zur Anzeige ob ein Fehler aufgetreten ist
- `$messages`: Evtl. von der REST api gemeldete Benachrichtigungen

## 3.2 Instanziierung des EasyCreditClients

Alle von der Bibliothek angebotenen Funktionen werden über die Klasse „*EasyCreditClient*“ zur Verfügung gestellt. Erzeugen Sie eine Instanz der Klasse:

```
require_once 'EasyCreditClient.php';

$client = new EasyCredit\EasyCreditClient(
    'myShopId',
    'myToken',
    'integrationType',
    'myLogFile',
    'logLevel'
);
```

### 3.2.1 Parameter

Der Konstruktor der Klasse erwartet folgende Parameter:

- myShopId: Ihre von der Teambank erhaltene Web Shop ID
- myToken: Ihr zu der Web Shop ID passender Token
- integrationType: ProcessInitialize::INTEGRATION\_TYPE\_SERVICE\_INTEGRATION oder ProcessInitialize::INTEGRATION\_TYPE\_PAYMENT\_PAGE
- myLogFile: vollständiger Pfad inkl. Dateiname der Logdatei. Wird null angegeben erfolgt kein Logging
- logLevel: LogLevel::Error oder LogLevel::Debug

## 3.3 Initialisierung eines Vorgangs

```
$response = $client->initialize(
    $customer,
    $billingAddress,
    $deliveryAddress,
    $orderTotal,
    $riskRelatedInfo,
    $cartInfos,
    $callbackUrls,
    $term,
    $technicalShopParams
);
```

### 3.3.1 Parameter

- \$customer: Klasse *EasyCredit\Transfer\Customer*
- \$billingAddress: Klasse *EasyCredit\Transfer\BillingAddress*
- \$deliveryAddress: Klasse *EasyCredit\Transfer\DeliveryAddress*
- \$orderTotal: Datentyp *float*

- \$riskrelatedInfo: Klasse *EasyCredit\Transfer\RiskRelatedInfo*
- \$cartInfos: Klasse *EasyCredit\Transfer\CartInfoCollection*
- \$callbackUrls: Klasse *EasyCredit\Transfer\CallbackUrls*
- \$term: Datentyp *int*
- \$technicalShopParams: Klasse *EasyCredit\Transfer\TechnicalShopParams*

### 3.3.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientInitializeResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- \$technicalProcessIdentifier: Datentyp *string*

## 3.4 Aktualisieren eines Vorgangs

```
$response = $client->update(
    $term,
    $employmentData,
    $contact,
    $agreement,
    $bankData,
    $birthDate
);
```

### 3.4.1 Parameter

- \$term: Datentyp *int*
- \$employmentData: Klasse *EasyCredit\Transfer\EmploymentData*
- \$contact: Klasse *EasyCredit\Transfer>Contact*
- \$agreement: Klasse *EasyCredit\Transfer\Agreement*
- \$bankData: Klasse *EasyCredit\Transfer\BankData*
- \$birthDate: Klasse *DateTime*

### 3.4.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientBaseResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- \$processed: Datentyp *bool*

### 3.5 Durchführen einer Entscheidung

```
$response = $client->decide();
```

#### 3.5.1 Parameter

Keine.

#### 3.5.2 Rückgabewert

Der Rückgabewert ist vom Typ `EasyCredit\Client\Result\EasyCreditClientDecideResult` und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- \$decision: Datentyp `string`

### 3.6 Abfrage der Entscheidung

```
$response = $client->getDecision(  
    $technicalProcessIdentifier  
);
```

#### 3.6.1 Parameter

- \$technicalProcessIdentifier: Datentyp `string`

#### 3.6.2 Rückgabewert

Der Rückgabewert ist vom Typ `EasyCredit\Client\Result\EasyCreditClientDecisionResult` und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- \$decision: Klasse `EasyCredit\Transfer\Decision`

### 3.7 Prüfung einer mTAN

```
$response = $client->verifyMTan(  
    $mTan  
);
```

#### 3.7.1 Parameter

- \$mTan: Datentyp `string`



### 3.7.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientBaseResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- \$processed: Datentyp *bool*

## 3.8 Erneutes Versenden einer mTAN

```
$response = $client->resendMTan(  
    $technicalProcessIdentifier  
);
```

### 3.8.1 Parameter

- \$technicalProcessIdentifier: Datentyp *string*

### 3.8.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientBaseResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- \$processed: Datentyp *bool*

## 3.9 Bestätigen eines Vorgangs

```
$response = $client->confirm();
```

### 3.9.1 Parameter

Keine.

### 3.9.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientBaseResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- \$processed: Datentyp *bool*

## 3.10 Abruf der Rechtstexte

```
$response = $client->getLegislativeText();
```

### 3.10.1 Parameter

Keine.

### 3.10.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientLegislativeTextResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- `$legislativeText`: Klasse *EasyCredit\Transfer\LegislativeText*

## 3.11 Abruf der gespeicherten Daten

```
$response = $client->getCommonProcessData(  
    $technicalProcessIdentifier  
);
```

### 3.11.1 Parameter

- `$technicalProcessIdentifier`: Datentyp *string*

### 3.11.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientCommonProcessDataResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- `$commonProcessData`: Klasse *EasyCredit\Transfer\CommonProcessData*

## 3.12 Abruf der Finanzierungsdaten

```
$response = $client->getFinancingDetails(  
    $technicalProcessIdentifier  
);
```

### 3.12.1 Parameter

- `$technicalProcessIdentifier`: Datentyp *string*

### 3.12.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientFinancingDetailsResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- `$installmentPlan`: Klasse *EasyCredit\Transfer\InstallmentPlan*

### 3.13 Abruf einer Beispiel-Finanzierung

```
$response = $client->getModelCalculation(  
    $amount  
);
```

#### 3.13.1 Parameter

- \$amount: Datentyp *float*

#### 3.13.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientModelCalculationResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- \$modelCalculation: Klasse *EasyCredit\Transfer\ModelCalculation*

### 3.14 Abruf des günstigsten Ratenplans

```
$response = $client->getBestModelCalculation (  
    $amount  
);
```

#### 3.14.1 Parameter

- \$amount: Datentyp *float*

#### 3.14.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientBestModelCalculationResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- \$bestModelCalculation: Klasse *EasyCredit\Transfer\ModelCalculationShort*

### 3.15 Abruf des Restbetrages der Ankaufobergrenze

```
$response = $client->getResidualAmountPurchaseLimit();
```

#### 3.15.1 Parameter

Keine.

### 3.15.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientResidualAmountLimitResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- `$residualAmountLimit:` Datentyp *float*

## 3.16 Prüfen ob Zahlart für aktuelle Bestellung zur Verfügung steht

Diese Funktion führt keine Interaktion mit der REST api der Teambank aus, sondern prüft die grundlegenden Voraussetzungen für die Zahlart Ratenkauf bei easyCredit:

- Betrag liegt zwischen 200,00€ und 5.000,00€
- Rechnungsland ist Deutschland
- Lieferland ist Deutschland
- Rechnungs- und Lieferadresse sind identisch

Werden Rechnungs- oder Lieferadresse nicht mit übergeben, wird lediglich der Betrag geprüft.

```
$response = $client->isPaymentMethodAvailable(  
    $amount,  
    $billingAddress,  
    $deliveryAddress  
);
```

### 3.16.1 Parameter

- `$amount:` Datentyp *float*
- `$billingAddress:` Klasse *EasyCredit\Transfer\BillingAddress*
- `$deliveryAddress:` Klasse *EasyCredit\Transfer\DeliveryAddress*

### 3.16.2 Rückgabewert

Der Rückgabewert ist vom Typ *EasyCredit\Client\Result\EasyCreditClientResidualAmountLimitResult* und enthält zusätzlich zu den Standardwerten (Siehe Punkt 3.1):

- `$isAvailable:` Datentyp *bool*