

LV CC API partner API documentation

- Introduction :
 - API partner process flow
 - Frequently asked questions
 - 1) How to present final loan offer to the customer on API Partner side?
 - 2) How to recognise if {"err":{"loan-offer":"data-not-found"}} is because offer was yet not created or we fail to create an offer due to low score?
- Required methods :
 - 1. registerCustomer:
 - 2. getLatestLoanOfferExternal
 - The request must consist of:
 - The function can return the following errors:
 - POST Request:
 - Response:
 - Latest Loan Offer response explanation
 - Loan Offer Statuses
 - 3. setCustomerPassword
 - The request must consist of:
 - The function can return the following errors:
 - POST Request:
 - Response:
 - 4. refreshExpiredLoanOfferDate:
 - The request must consist of:
 - The function can return the following errors:
 - POST Request:
 - Response:
- Examples:
 - Empty request.
 - Missing password.
 - Missing data section.
 - Missing customer-id
 - Request with customer-id, that is not in our database.
 - Request for refresh loan offer data, with customer-id assigned to customer without expired and not open last loan offer.
 - Request for refresh loan offer data, with customer-id assigned to customer with active loan offer.
 - Request for refresh loan offer, correct case.
 - Request for loan offer data, correct case.
 - Request for loan offer data, correct case, but with some problems with loan offer settings/

Introduction :

<https://platon.docs.apiary.io/#introduction>

API partner process flow

We created a mapper from API partner fields to platon required fields. You should send the bank statement data (Nordigen data) directly with the customer registration data, under key bank-statement-data.

So we expect request body like in example below:

```
{
  "data": {
    "bank-statement-data": {
      "data": {
        "attributes": {
          "categorisation": {
            "accounts": [
              {
                "account_number": "LV13 HABA 0000 0000 0",
                "bank": "Swedbank",
                "country": "lv",
                "currencies": [
                  {
                    "code": "EUR",
                    "credit_turnover": 1000,
                    "debit_turnover": 1000,
                    "end_balance": 2.27,
                    "start_balance": 61.61,
                    "transactions": [
                      {
                        "amount": 12,
                        "category_id": 733,
                        "date": "2019-06-03",
                        "info": "Komand\u0113juma",
                        "partner": "SERTIKON",
                        "title": "Other incoming",
                        "transaction_id": "a2679e"
                      },
                      {
                        "amount": -1.09,
                        "category_id": 26,
                        "date": "2019-06-04",
                        "info": "PIRKUMS 557",
                        "partner": "Asnini 2-ve",
                        "title": "Groceries",
                        "transaction_id": "28"
                      },
                      {
                        "amount": -1.15,
                        "category_id": 26,
                        "date": "2019-06-04",
                        "info": "PIRKUMS 55736",
                        "partner": "Asnini 2-vei",
                        "title": "Groceries",
                        "transaction_id": "860d40"
                      }
                    ]
                  }
                ]
              }
            ]
          },
          "holders": [
            {
              "name": "Name Surname"
            }
          ],
          "iban": {
            "bank_code": "HABA",
            "bank_name": "SWEDBANK AS",
            "bic": "HABALV22",
            "country_code": "LV",
            "iban": "LV13HABA000000000000"
          },
          "is_joint_ownership": false,
          "is_shared_ownership": false,
          "period_end": "2020-01-15",
          "period_start": "2019-06-03"
        }
      ],
      "category_tree_version": null
    },
    "observations": {
      "GAMBLING": {
        "description": "Person has made and/or received",
        "severity": "RED"
      },
      "INTERMEDIATE_FINAL_SALARY": {
        "description": "Person has received final "
      }
    }
  }
}
```

```

        "severity": "PINK"
      },
      "TRANSFER_TO_OWN_ACCOUNT": {
        "description": "Transfer to own account",
        "severity": "RED"
      }
    },
    "status": "completed"
  },
  "type": "report processing status"
}
},
"customer-personcode": "150291-14338",
"customer-phone": "29424189",
"customer-email": "testtest@test.lv",
"customer-firstname": "Test",
"customer-lastname": "Test",
"customer-birthday": "1977-02-06",
"customer-monthly-income": 0,
"customer-gender": 0,
"language": "lv",
"loan-amount": 5000,
"loan-term": 36,
"new-registration": true,
"use-products": "1",
"product": 2
},
"login": "API partner username",
"password": "password"
}

```

2. getLatestLoanOfferExternal

The `getLatestLoanOfferExternal` function returns a recent loan Offer for requested customer .

The request must consist of:

customer-id -> id of the customer received from `registerCustomer` function.

The function can return the following errors:

loan-final-interest-rate/loan-offered-APR/loan-offered-monthly-payment: error-during-processing -> during interest rate calculation, error occurred,

loan-final-interest-rate/loan-offered-APR/loan-offered-monthly-payment: error-loan-offer-could-not-be-found -> if during interest rate calculation, loan offer could not be found,

loan-offer: data-not-found -> if the customer doesn't have a loan offer.

POST Request:

URL: <https://platon.cf-it.at/affiliate/getLatestLoanOfferExternal>

HEADERS: Content-Type:application/json

BODY:

```

{
  "login": "apiary",
  "password": "apiary",
  "data": {
    "customer-id": "3333"
  }
}

```

Response:

Status : 200

HEADERS: Content-Type:application/json

POSSIBLE BODYS:

```
{ "err": [ "login-missing", "pass-missing" ] }
```

```
{ "err": [ "pass-missing" ] }
```

```
{ "err": [ "unknown-customer" ] }
```

```
{ "err": { "loan-offer": "data-not-found" } }
```

```
{
  "ok": 1,
  "data": {
    "blocking-time": null,
    "loan-offer": {
      "created_at": "2020-09-23T15:59:53+03:00",
      "chosen_amount": "1000.00",
      "chosen_term": 36,
      "status": 5,
      "transaction_mapping": "AOWEFANU",
      "payout_mode": null,
      "expiry": "2020-11-23T14:59:53+02:00",
      "max_repayable": null,
      "additional_interest_rate": "4.00",
      "location_id": 60,
      "created_by_user_id": null,
      "fulfillment_date": "2020-09-23 08:25:34"
    },
    "loan-final-interest-rate": 24
  },
  "loan-offered-APR": 21.90943145751953,
  "loan-offered-monthly-payment": 185.92,
  "loan-maximum-principal": "4000.00"
}
```

```

{
  "ok": 1,
  "data": {
    "blocking-time": null,
    "loan-offer": {
      "created_at": "2020-10-01T12:18:17+03:00",
      "chosen_amount": "1000.00",
      "chosen_term": 36,
      "status": 2,
      "transaction_mapping": "AOWEFANU",
      "payout_mode": 2,
      "expiry": "2020-11-23T14:59:53+02:00",
      "max_repayable": null,
      "additional_interest_rate": "4.00",
      "location_id": 60,
      "created_by_user_id": 1,
      "fulfillment_date": "2020-09-23 08:25:34"
    },
    "loan-final-interest-rate": "error-during-processing",
    "loan-offered-APR": "error-during-processing",
    "loan-offered-monthly-payment": "error-during-processing",
  }
}

```

Latest Loan Offer response explanation

transaction mapping - its internal ID of the offer, so you don't need to care about that so much.

payout mode - means defined payout mode we will use to payout the loan, if its null then its default (in our case bank transaction) but you will also don't need this I think.

max_repayable - that's the maximum monthly amount customer can afford (coming form our calculations).

location ID - internal ID of our brand across all counties - you will work only with LV Creamcredit so it will be always the same for you.

created by user ID - its filled loan offer is created by our employee so we can recognise which one, you will also don't need this field.

fulfillment_date - date when original offer was created, we calculate 5 days when offer is active from it.

additional interest fee -> additional interest rate represents one part of final offered interest rate: interest rate = rate based on term + additional interest rate.

loan-final-interest-rate -> it's calculated final-interest-rate for a loan created from this loan offer.

loan-maximum-principal -> it's calculated maximum principal we can offer to customer for a loan created from this loan offer.

loan-offered-APR - it's calculated apr for a loan created from this loan offer.

loan-offered-monthly-payment - it's calculated monthly installment payment for a loan created from this loan offer.

Loan Offer Statuses

1. "Waiting" - waiting for evaluation to give an offer or reject it
2. "Open" - Loan offer was created (that means we are able to approve customers loan within calculated limits) and we are waiting if client accepts it
3. "Done" - Customer accepted Loan offer
4. "Rejected" - Rejected Loan offer, customer either rejected it or we did because we are not able to offer customer anything
5. "Expired" - Loan offer waited in "Open" status for more than 5 days. Status changed to Expired afterwards.
6. "Cancelled" - Canceled by customer or by Creamfinance or by API partner but mostly not used as usually it ends in Expired
7. "Pending fulfillment of requirements" - not used now but this status means that we are missing some information to evaluate offer

3. setCustomerPassword

The setCustomerPassword function sets a new password for the customer.

The request must consist of:

customer-id -> id of the customer received from registerCustomer function.

The function can return the following errors:

customer-password: regex-mismatch -> if password don't pass validation (minimum 5 sign, must contain letter and number),

customer-password: field-missing -> if no customer-password field was provided,

loan-offer: data-not-found -> if the customer don't have loan offer,

unknown-customer -> if customer-id is unknown for affiliate.



Password Constrains

Requirements for new password creation:

- Minimum 5 signs
- At least 1 Capital and 1 Lower Case Letter
- At least 1 Number

POST Request:

URL: <http://cream.test/affiliate/setCustomerPassword>

HEADERS: Content-Type:application/json

BODY:

```
{
  "login": "apiary",
  "password": "apiary",
  "data": {
    "customer-id": "3333",
    "customer-password": "password"
  }
}
```

Response:

Status : 200

HEADERS: Content-Type:application/json

POSSIBLE BODYS:

```
{ "err": [ "login-missing", "pass-missing" ] }
```

```
{ "err": [ "pass-missing" ] }
```

```
{ "err": [ "unknown-customer" ] }
```

```
{ "err": { "customer-password": "regex-mismatch" } }
```

```
{ "err": { "customer-password": "field-missing" } }
```

```
{ "ok": 1, "data": [] }
```

4. refreshExpiredLoanOfferDate:

This function allows you to refresh expired loan offers. Loan offer is valid only for 30 days after the original offer is created.

The request must consist of:

customer-id -> id of the customer received from registerCustomer function.

The function can return the following errors:

customer: has-active-loan-offer -> if the customer already has active loan offer

customer: last-loan-offer-is-not-expired -> if the customer's last loan offer is not open and has not expired

loan-offer: data-not-found -> if the customer don't have loan offer

loan-offer: too-old -> if the loan offer is older than 30 days, it cannot be refreshed.

POST Request:

URL: <https://platon.cf-it.at/affiliate/refreshExpiredLoanOfferDate>

HEADERS: Content-Type:application/json

BODY:

```
{
  "login": "apiary",
  "password": "apiary",
  "data": {
    "customer-id": "3333"
  }
}
```

Response:

Status : 200

HEADERS: Content-Type:application/json

POSSIBLE BODY:

```
{"err":["login-missing","pass-missing"]}
```

```
{"err":["login-failed"]}
```

```
{"err":["pass-missing"]}
```

```
{"err":["unknown-customer"]}
```

```
{"err":{"customer":"has-active-loan-offer"}}
```

```
{"err":{"loan-offer":"too-old"}}
```

```
{"err":{"customer":"last-loan-offer-is-not-expired"}}
```

```
{"err":{"loan-offer":"data-not-found"}}
```

```
{"ok":1,"data":[]}
```

Examples:

Empty request.

URL: <https://platon.cf-it.at/affiliate/refreshExpiredLoanOfferDate>

Request body:

```
{}
```

Response body:

```
{"err":["login-missing","pass-missing"]}
```

Missing password.

URL: <https://platon.cf-it.at/affiliate/refreshExpiredLoanOfferDate>

Request body:

```
{  
  "login": "AFFILIATE_NAME"  
}
```

Response body:

```
{"err":["pass-missing"]}
```

Missing data section.

URL: <https://platon.cf-it.at/affiliate/refreshExpiredLoanOfferDate>

Request body:

```
{  
  "login": "AFFILIATE_NAME",  
  "password": "AFFILIATE_PASSWORD"  
}
```

Response body:

```
{"err":["data-missing"]}
```

Missing customer-id

URL: <https://platon.cf-it.at/affiliate/refreshExpiredLoanOfferDate>

Request body:

```
{  
  "login": "AFFILIATE_NAME",  
  "password": "AFFILIATE_PASSWORD",  
  "data": {  
  
  }  
}
```

Response body:

```
{ "err": { "login-email": [ "field-missing" ], "login-password": [ "field-missing" ] } }
```

Request with customer-id, that is not in our database.

URL: <https://platon.cf-it.at/affiliate/getOpenLoanOfferExternal>

Request body:

```
{
  "login": "AFFILIATE_NAME",
  "password": "AFFILIATE_PASSWORD",
  "data": {
    "customer-id": "ID"
  }
}
```

Response body:

```
{ "err": [ "unknown-customer" ] }
```

Request for refresh loan offer data, with customer-id assigned to customer without expired and not open last loan offer.

URL: <https://platon.cf-it.at/affiliate/refreshExpiredLoanOfferDate>

Request body:

```
{
  "login": "AFFILIATE_NAME",
  "password": "AFFILIATE_PASSWORD",
  "data": {
    "customer-id": "2184745"
  }
}
```

Response body:

```
{ "err": { "customer": "last-loan-offer-is-not-expired" } }
```

Request for refresh loan offer data, with customer-id assigned to customer with active loan offer.

URL: <https://platon.cf-it.at/affiliate/refreshExpiredLoanOfferDate>

Request body:

```
{
  "login": "AFFILIATE_NAME",
  "password": "AFFILIATE_PASSWORD",
  "data": {
    "customer-id": "ID"
  }
}
```

Response body:

```
{ "err": { "customer": "has-active-loan-offer" } }
```

Request for refresh loan offer, correct case.

URL: <https://platon.cf-it.at/affiliate/refreshExpiredLoanOfferDate>

Request body:

```
{  
  
  "login": "AFFILIATE_NAME",  
  "password": "AFFILIATE_PASSWORD",  
  "data": {  
    "customer-id": "ID"  
  }  
}
```

Response body:

```
{ "ok": 1, "data": [] }
```

Request for loan offer data, correct case.

URL: <https://platon.cf-it.at/affiliate/getOpenLoanOfferExternal>

Request body:

```
{  
  
  "login": "AFFILIATE_NAME",  
  "password": "AFFILIATE_PASSWORD",  
  "data": {  
    "customer-id": "ID"  
  }  
}
```

Response body:

```
{  
  "ok": 1,  
  "data": {  
    "blocking-time": null,  
    "loan-offer": {  
      "created_at": "2020-09-23T15:59:53+03:00",  
      "chosen_amount": "1000.00",  
      "chosen_term": 36,  
      "status": 2,  
      "transaction_mapping": "AOWEFANU",  
      "payout_mode": null,  
      "expiry": "2020-11-23T14:59:53+02:00",  
      "max_repayable": null,  
      "additional_interest_rate": "4.00",  
      "location_id": 60,  
      "created_by_user_id": null,  
      "fulfillment_date": "2020-09-23 08:25:34"  
    },  
    "loan-final-interest-rate": 24,  
    "loan-maximum-principal": "300.00"  
  }  
}
```

Request for loan offer data, correct case, but with some problems with loan offer settings/

URL: <https://platon.cf-it.at/affiliate/getOpenLoanOfferExternal>

Request body:

```
{
  "login": "AFFILIATE_NAME",
  "password": "AFFILIATE_PASSWORD",
  "data": {
    "customer-id": "ID"
  }
}
```

Response body:

```
{
  "ok": 1,
  "data": {
    "blocking-time": null,
    "loan-offer": {
      "created_at": "2020-10-01T12:18:17+03:00",
      "chosen_amount": "1000.00",
      "chosen_term": 36,
      "status": 2,
      "transaction_mapping": "AOWEFANU",
      "payout_mode": 2,
      "expiry": "2020-11-23T14:59:53+02:00",
      "max_repayable": null,
      "additional_interest_rate": "4.00",
      "location_id": 60,
      "created_by_user_id": 1,
      "fulfillment_date": "2020-09-23 08:25:34"
    },
    "loan-final-interest-rate": "error-during-processing",
    "loan-maximum-principal": "300.00"
  }
}
```