

Base URLs

The base URL for requests to the API is: <https://sandbox.compassway.org/api/>

HTTP Verbs

Standard HTTP verbs are used to indicate the API request method.

VERB	FUNCTION
GET	To retrieve a resource or a collection of resources
POST	To create a resource
PATCH	To modify an existing resource
PUT	To replace an existing resource
DELETE	To delete a resource

HTTP response status codes

HTTP response status codes indicate whether a specific [HTTP](#) request has been successfully completed. Responses are grouped in five classes:

1. [Informational responses](#) (100 – 199)
2. [Successful responses](#) (200 – 299)
3. [Redirection messages](#) (300 – 399)
4. [Client error responses](#) (400 – 499)
5. [Server error responses](#) (500 – 599)

Authorization

Login into the system

We call this endpoint to login into the system. The same endpoint is used for a client, lender, guarantor, etc. After a successful login authorization token is generated, it will be required for further actions.

Endpoint: </api/auth/login/>

Method: **POST**

Headers:

Content-Type: application/json

Request body:

```
{
  "username": "string", //User email (like
[addressname]@[domain].[top_domain])
  "password": "string" // user password
}
```

Response example:

```
{
  "expiry": "string(date-time)", //date of token expiration (like 2024-05-31T13:27:44.946561+03:00)
  "token": "string" //authorization token
}
```

Status codes:

Decision making

Automated

To use an automated option for decision-making, credit policy related to the application has to have a value false in the field “manual_verification“. (Check [Credit policy](#))

Verifier

A verifier is a single person who makes decisions (approve, reject, request for the guarantor) on the application

Get/send verifiers

We use this endpoint to receive or send information about verifiers for a given user;

Endpoint: */api/loan_routing/verifiers/*

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "id": integer, //Application Id
  "user": {
    "id": integer, //User Id
    "email": "string", //User email (like [addressname]@[domain].[top_domain])
    "last_name": "string", //User last name
    "first_name": "string" //User first name
  },
}
```

```
    "loan_amount_from": number, //Loan lower bound
    "loan_amount_to": number, //Loan upper bound
    "fact_addr_regions": ["string"], //List of regions where given verifiers
can make decisions
    "processed_applications": integer, //number of processed applications
    "credit_policies": [integer], //List of credit policies id related to
the verifier
    "shops": [integer] //List of shop IDs related to the verifier
}
```

Status codes:

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "loan_amount_from": number, //Loan lower bound
  "loan_amount_to": number, //Loan upper bound
  "fact_addr_regions": ["string"], //List of regions where given verifiers can
make decisions
  "user": integer, //User Id
  "processed_applications": integer, //number of processed applications
  "credit_policies": [integer], //List of credit policies id related to the
verifier
  "shops": [integer] //List of shop IDs related to the verifier
}
```

Response example:

```
{
  "id": integer, //Application Id
  "user": {
    "id": integer, //User Id
    "email": "string", //User email (like
[addressname]@[domain].[top_domain])
    "last_name": "string", //User last name
    "first_name": "string" //User first name
  },
  "loan_amount_from": number, //Loan lower bound
  "loan_amount_to": number, //Loan upper bound
  "fact_addr_regions": ["string"], //List of regions where given verifiers
can make decisions
  "processed_applications": integer, //number of processed applications
  "credit_policies": [integer], //List of credit policies id related to
the verifier
  "shops": [integer] //List of shop IDs related to the verifier
}
```

Get/change/delete verifier by given ID

We use this endpoint to receive or change information about verifiers for a given verifier ID;

Endpoint: `/api/loan_routing/verifiers/{id}/`

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "id": integer, //Application Id
  "user": {
    "id": integer, //User Id
    "email": "string", //User email (like
[addressname]@[domain].[top_domain])
    "last_name": "string", //User last name
    "first_name": "string" //User first name
  },
  "loan_amount_from": number, //Loan lower bound
  "loan_amount_to": number, //Loan upper bound
  "fact_addr_regions": ["string"], //List of regions where given verifiers
can make decisions
  "processed_applications": integer, //number of processed applications
  "credit_policies": [integer], //List of credit policies id related to
the verifier
  "shops": [integer] //List of shop IDs related to the verifier
}
```

Method: **PATCH**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "loan_amount_from": number, //Loan lower bound
  "loan_amount_to": number, //Loan upper bound
  "fact_addr_regions": ["string"], //List of regions where given verifiers can
make decisions
  "user": integer, //User Id
  "processed_applications": integer, //number of processed applications
  "credit_policies": [integer], //List of credit policies id related to the
verifier
  "shops": [integer] //List of shop IDs related to the verifier
}
```

Response example:

```
{
  "id": integer, //Application Id
  "user": {
    "id": integer, //User Id
    "email": "string", //User email (like
[addressname]@[domain].[top_domain])
    "last_name": "string", //User last name
    "first_name": "string" //User first name
  },
  "loan_amount_from": number, //Loan lower bound
  "loan_amount_to": number, //Loan upper bound
  "fact_addr_regions": ["string"], //List of regions where given verifiers
can make decisions
  "processed_applications": integer, //number of processed applications
  "credit_policies": [integer], //List of credit policies id related to
the verifier
  "shops": [integer] //List of shop IDs related to the verifier
}
```

Method: DELETE

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (response **status code** must be **204** if the verifier was successfully deleted)

Approve application

This endpoint is used to approve the application.

Endpoint: /api/client/applications/{application_id}/approve/

Method: POST

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (**status code** must be **201** if the application was successfully approved)

Reject application

This endpoint is used to reject applications.

Endpoint: `/api/client/applications/{application_id}/reject/`

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (**status code** must be **201** if the application was successfully rejected)

Committee

A credit committee is a group of people who make a decision on an application. It contains of supervisor and other members. Decision options are the same as for the verifier + sent to the supervisor (for a regular committee member).

Get/send a committee

The lender uses this endpoint to get or create a committee of verifiers.

Endpoint: `/api/loan_routing/credit_committees/`

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "id": integer, //Application Id
  "head": {
    "id": integer, //User Id
    "email": "string", //User email (like
[addressname]@[domain].[top_domain])
    "last_name": "string", //User last name
    "first_name": "string" //User first name
  },
  "users": {
    "id": integer, //User Id
```

```

        "email": "string", //User email (like
[addressname]@[domain].[top_domain])
        "last_name": "string", //User last name
        "first_name": "string" //User first name
    },
    "decision_time_mins": integer, //Time for voting in minutes
    "loan_amount_from": number, //Loan lower bound
    "loan_amount_to": number, //Loan upper bound
    "fact_addr_regions": ["string"], //List of regions where given verifiers
can make decisions
    "processed_applications": integer, //number of processed applications
    "credit_policies": [integer], //List of credit policies id related to
the verifier
    "shops": [integer] //List of shop IDs related to the verifier
}

```

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```

{
    "name": "string", //Committee name
    "head": integer, //Head id
    "users": {
        "id": integer, //User Id
    },
    "decision_time_mins": integer, //Time for voting in minutes
    "loan_amount_from": number, //Loan lower bound
    "loan_amount_to": number, //Loan upper bound
    "fact_addr_regions": ["string"], //List of regions where given verifiers
can make decisions
    "processed_applications": integer, //number of processed applications
    "credit_policies": [integer], //List of credit policies id related to
the verifier
    "shops": [integer] //List of shop IDs related to the verifier
}

```

Response example:

```

{
    "id": integer, //Application Id
    "head": {
        "id": integer, //User Id
        "email": "string", //User email (like
[addressname]@[domain].[top_domain])
        "last_name": "string", //User last name
        "first_name": "string" //User first name
    },
    "users": {
        "id": integer, //User Id

```

```

        "email": "string", //User email (like
[addressname]@[domain].[top_domain])
        "last_name": "string", //User last name
        "first_name": "string" //User first name
    },
    "decision_time_mins": integer, //Time for voting in minutes
    "loan_amount_from": number, //Loan lower bound
    "loan_amount_to": number, //Loan upper bound
    "fact_addr_regions": ["string"], //List of regions where given verifiers
can make decisions
    "processed_applications": integer, //number of processed applications
    "credit_policies": [integer], //List of credit policies id related to
the verifier
    "shops": [integer] //List of shop IDs related to the verifier
}

```

Committee member vote

Committee member uses this endpoint to vote

Endpoint: `/api/loan_routing/credit_committee_votings/{voting_id}/vote/`

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```

{
    "is_positive_vote": boolean, // true if positive, false otherwise
    "comment": "string" //comment from a voter
}

```

Response example:

```

{
    "id": integer, //Voting id
    "application": integer, //Application id
    "credit_committee": integer, //Committee id
    "is_active": boolean, //true if active, false otherwise
    "expire": "string(date-time)", //time of expiration (like 2024-05-
31T13:27:44.946561+03:00)
    "votes": [
        {
            "id": integer, // committee member vote id
            "user": {
                "id": integer, //user id
                "email": "string", //User email (like
[addressname]@[domain].[top_domain])
                "last_name": "string", //user last name
                "first_name": "string" //user name
            },

```

```
        "is_positive_vote": boolean, // true if positive, false otherwise
        "comment": "string", //comment from a voter
        "created_at": "string(date-time)", //creation date (like 2024-05-
31T13:27:44.946561+03:00)
        "updated_at": "string(date-time)", //update date (like 2024-05-
31T13:27:44.946561+03:00)
        "credit_committee_voting": id //committee voting id
    }
},
    "created_at": "string(date-time)", //creation date (like 2024-05-
31T13:27:44.946561+03:00)
    "updated_at": "string(date-time)" //update date (like 2024-05-
31T13:27:44.946561+03:00)
}
```

Endpoint: `/api/loan_routing/credit_committees/{id}/`

Method: **DELETE**

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (response **status code** must be **204** if the verifier was successfully deleted)

Shop Request

Creating shop request

This endpoint is used for creating shop requests. It's the initial phase of creating an application for a loan.

Endpoint: `/api/api_integration/shop_request/`

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
    "product_category_id": integer, //Product category id
    "shop_id": integer, //Shop id
    "credit_policy_id": integer, //Credit policy id
    "region": "string", //Region
    "product_name": "string", //Product name
}
```

```

    "product_id": integer, //Product id
    "product_price": number, //List of shop IDs related to the verifier
    "client_name": "string", //Client name
    "client_phone": "string", //Client phone
    "loan_term": integer, // Loan term in days
    "promo_code": "string", //Promo code
    "prepayment_amount": integer //Prepayment amount
}

```

Response example:

```

{
  "uuid": "string", // UUID
  "credit_policy": {
    "id": integer, // credit policy id
    "collaterals": [
      {
        "id": integer, // Collateral id
        "collateral_type": {
          "id": integer, // id related to list in lower cell
          "name": "string" // Collateral type (current options - "Deposit",
"Goods purchased at the expense of a loan", "Real estate", "Automobile",
"Guarantee", "Property surety", "Financial Guarantee")
        },
        "monitoring_frequency": integer, // monitoring frequency in days
        "required": boolean // true if collateral is required, false otherwise
      }
    ],
    "verifier_type": {
      "id": integer, // verifier id
      "name": "string", // One the possible options: "Credit committee",
"Verifier".
      "name_en": "string", // Name [en]
      "name_uk": "string", // Name [uk]
      "name_ru": "string", // Name [ru]
      "name_pl": "string" // Name [pl]
    },
    "name": "string", // Credit policy name
    "regions": ["string"], // List of regions where credit policy is applicable
    "is_prepayment": boolean, // true if there is a prepayment, false otherwise
    "loan_amount_limit_from": number, // loan lower bound
    "loan_amount_limit_to": number, // loan upper bound
    "loan_term_limit_from": number, // loan duration lower bound (days)
    "loan_term_limit_to": number, // loan duration upper bound (days)
    "interest": number, // Interest in percents
    "interest_delay_value": integer, // Interest delay value
    "interest_delay_unit": integer, // Interest delay unit; 1 - days, 2 - weeks,
3 - months, 4 - years
    "overdue_interest": number, // Overdue interest in percentage
    "overdue_interest_delay_value": integer, // Overdue interest delay value
    "overdue_interest_delay_unit": integer, // Overdue interest delay unit; 1 -
days, 2 - weeks, 3 - months, 4 - years
    "overdue_interest_counting_type": ["string"], // Overdue interest counting
type
    "overdue_interest_delay_value_min": integer, // Overdue interest delay value
min
    "overdue_interest_delay_value_max": integer, // Overdue interest delay value
max

```

```
"overdue_tolerance_amount_percent": number, // Overdue tolerance in
percentage
"overdue_tolerance_amount_fixed": integer, // Overdue tolerance amount fixed
"interest_method": ["string"], // interest calculation method; 3 possible
options - 1 for "Declining Balance", 2 for "Declining Balance
(Equal Installments)", 3 for "Fixed Flat
"repayment_frequency_n": integer, // Repayment frequency N days
"credit_type": ["string"], // Credit type
"manual_verification": boolean, // Manual verification enabled/disabled
"visible_on_calculator": boolean, // Visible on calculator enabled/disabled
"is_guarantor_required": boolean, // Guarantor required enabled/disabled
"discount_market_value": number, // Discount amount to market value
"created_at": "string(date-time)", //created at (like 2024-05-
31T13:27:44.946561+03:00)
"updated_at": "string(date-time)", //updated at (like 2024-05-
31T13:27:44.946561+03:00)
"prepayment_min": number, // Prepayment min in percentage
"prepayment_max": number, // Prepayment max in percentage
"merchant_fee": "string", // Merchant fee
"interest_calculation_period": ["string"], // Interest calculation period
"only_interest_till_final_disbursement": boolean, // true if only interest
are considered till final disbursement, false otherwise
"is_tranches_enabled": boolean, // Is tranches enabled
"min_tranches": integer, // Min amount of tranches
"max_tranches": integer, // Max amount of tranches
"disbursement_amount_more_approved": boolean, // Disbursement amount more
approved enabled/disabled
"full_prepayment": boolean, // Full prepayment enabled/disabled
"prepayment_strategy": ["string"], // prepayment strategy; two possible
options - "No change in Installments", "Installment changes" (In the first case,
prepayment does not impact the total loan cost, while in the second case, it
does.)
"parent_credit_policy": integer, // Parent credit policy
"client_type": integer, // Client type 1 for new client, 2 for existed
client
"percent_accrual_schema": integer, // 1 for "The number of days in a month
and a year corresponds to the calendar", 2 for "The number of days in a month
corresponds to a calendar year 360 days", 3 for "The number of days in a month
30 days corresponds to 360 days in a year"
"repayment_frequency_value": integer, // Repayment frequency; 1 for "Equal
Periods", 2 for "One-time payment"
"repayment_frequency_unit": integer, // Repayment frequency unit
"scoring_model": integer // Selected scoring model
},
"url": "string", // Url
"is_expired": "string", // Is expired
"region": "string", // Region
"product_name": "string", // Product name
"product_id": "string", // Client product ID
"product_price": number, // Product price
"client_name": "string", // Client name
"client_phone": "string", // Client phone
"loan_term": integer, // Loan term in days
"promo_code": "string", // Promo code
"prepayment_amount": integer, // Prepayment amount
"shop": {
  "id": integer, // ID
```

```

"url": "string", // Shop URL
"name": "string", // Shop legal name
"iban": "string", // Shop IBAN
"registry_identifier": "string", // Shop registry identifier
"address": "string", // Shop address
"bank_name": "string", // Shop bank name
"bank_code": "string", // Shop bank registry code
"years_in_bussiness": integer, // Years in Business
"number_of_employees": integer, // Years in Business
"description": "string", // Description
"billing_address": "string", // Billing Address
"email": "string", //email (like [addressname]@[domain].[top_domain])
"phone_number": "string", // Phone number
"user": [integer] // List of current shop users ID
},
"category": {
  "id": integer, // id which relates to category names below
  "name": "string", // category name; one of the following options: "Money on
card", "Laptops and Computers", "Smartphones, TV and electronics", "Products for
gamers", "Appliances", "Household products", "Tools and car goods", "Plumbing
and repair", "Cottage, garden and vegetable garden", "Sports and hobbies",
"Clothes, shoes and jewelry", "Health and beauty", "Children`s goods",
"Stationery and books", "Alcoholic drinks and food", "Business goods and
services", "Tours and Leisure"
  "name_en": "string", // Category name [en]
  "name_uk": "string", // Category name [uk]
  "name_ru": "string", // Category name [ru]
  "name_pl": "string" // Category name [pl]
}
}
}

```

Credit policy

Create credit policy

This endpoint is used to create credit policies (credit products). It can be used only after the lender authorizes into the system.

Endpoint: /api/lender/credit_policies/

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```

{
  "name": "string", // Credit policy name
  "regions": ["string"], // List of regions where credit policy is applicable
  "is_prepayment": boolean, // true if there is a prepayment, false otherwise
  "loan_amount_limit_from": number, // loan lower bound
}

```

```
"loan_amount_limit_to": number, // loan upper bound
"loan_term_limit_from": number, // loan duration lower bound (days)
"loan_term_limit_to": number, // loan duration upper bound (days)
"interest": number, // interest value (from 0 to 1)
"interest_delay_value": integer, // Interest delay value
"interest_delay_unit": integer, // Interest delay unit; 1 - days, 2 - weeks,
3 - months, 4 - years
"overdue_interest": number, // Overdue interest
"overdue_interest_delay_value": integer, // Overdue interest delay value
"overdue_interest_delay_unit": integer, // Overdue interest delay unit; 1 -
days, 2 - weeks, 3 - months, 4 - years
"overdue_interest_counting_type": ["string"], // Overdue interest counting
type; two possible options - "From First Overdue Date", "From Oldest Late
Payment"
"overdue_interest_delay_value_min": integer, // Overdue interest delay value
min
"overdue_interest_delay_value_max": integer, // Overdue interest delay value
max
"overdue_tolerance_amount_percent": number, // Overdue tolerance amount
percent
"overdue_tolerance_amount_fixed": integer, // The amount of debt in which
the loan isn't considered overdue
"interest_method": ["string"], // interest calculation method; 3 possible
options - 1 for "Declining Balance", 2 for "Declining Balance (Equal
Installments)", 3 for "Fixed Flat"
"repayment_frequency_n": integer, // Repayment frequency; 1 for "Payments
every N days in equal installments (loan body/quantity)", 2 for "Custom dates
and amounts"
"credit_type": "string", // one of the following options: "Money on card" or
"Consumer finance"
"manual_verification": boolean, // true if there is manual verification,
false otherwise
"visible_on_calculator": boolean, // True if visible, false otherwise
"is_guarantor_required": boolean, // true if guarantor is required, false
otherwise
"prepayment_min": number, // Prepayment min
"prepayment_max": number, // Prepayment max
"merchant_fee": "string", // Merchant fee
"interest_calculation_period": "string", // Interest calculation period; two
possible options - "After period", "On a Specified Date"
"only_interest_till_final_disbursement": boolean, // true if only interest
are considered till final disbursement, false otherwise
"is_tranches_enabled": boolean, // true if tranches are enabled, false
otherwise
"min_tranches": integer, // minimum amount of tranches
"max_tranches": integer, // maximum amount of tranches
"disbursement_amount_more_approved": boolean, // True if only interest is
applied till tranche issuance date
"full_prepayment": boolean, // true if full prepayment is enabled, false
otherwise
"prepayment_strategy": "string", // prepayment strategy; two possible
options - "No change in Installments", "Installment changes" (In the first case,
prepayment does not impact the total loan cost, while in the second case, it
does.)
"parent_credit_policy": integer, // Parent credit policy
"parent_credit_policy_id": integer, // Parent credit policy id
```

```

    "client_type": integer, // Client type 1 for new client, 2 for existed
client
    "percent_accrual_schema": integer, // 1 for "The number of days in a month
and a year corresponds to the calendar", 2 for "The number of days in a month
corresponds to a calendar year 360 days", 3 for "The number of days in a month
30 days corresponds to 360 days in a year"
    "repayment_frequency_value": integer, // Repayment frequency; 1 for "Equal
Periods", 2 for "One-time payment"
    "repayment_frequency_unit": integer, // 1 for "Days", 2 for "Weeks", 3 for
"Month", 4 for "Years"
    "scoring_model": integer, // 1 for new client, 2 for existed client
    "verifier_type": integer // 1 for "Verifier", 2 for "Credit committee"
}

```

Response example:

```

{
  "id": integer, // credit policy id
  "collaterals": [
    {
      "id": integer, // Collateral id
      "collateral_type": {
        "id": integer, // id related to list in lower cell
        "name": "string" // one of the following options: "Deposit", "Goods
purchased at the expense of a loan", "Real estate", "Automobile", "Guarantee",
"Property surety", "Financial Guarantee"
        "name_en": "string", // Name [en]
        "name_uk": "string", // Name [uk]
        "name_ru": "string", // Name [ru]
        "name_pl": "string" // Name [pl]
      },
      "monitoring_frequency": integer, // monitoring frequency in days
      "required": boolean // true if collateral is required, false otherwise
    }
  ],
  "verifier_type": {
    "id": integer, // verifier id
    "name": "string", // One the possible options: "Credit committee",
"Verifier"
    "name_en": "string", // Name [en]
    "name_uk": "string", // Name [uk]
    "name_ru": "string", // Name [ru]
    "name_pl": "string" // Name [pl]
  },
  "name": "string", // Credit policy name
  "regions": ["string"], // List of regions where credit policy is applicable
  "is_prepayment": boolean, // true if there is a prepayment, false otherwise
  "loan_amount_limit_from": number, // loan lower bound
  "loan_amount_limit_to": number, // loan upper bound
  "loan_term_limit_from": number, // loan duration lower bound (days)
  "loan_term_limit_to": number, // loan duration upper bound (days)
  "interest": number, // Interest
  "interest_delay_value": integer, // Interest delay value
  "interest_delay_unit": integer, // Interest delay unit; 1 - days, 2 - weeks,
3 - months, 4 - years
  "overdue_interest": number, // Overdue interest
  "overdue_interest_delay_value": integer, // Overdue interest delay value
}

```

```

    "overdue_interest_delay_unit": integer, // Overdue interest delay unit; 1 -
days, 2 - weeks, 3 - months, 4 - years
    "overdue_interest_counting_type": ["string"], // Overdue interest counting
type
    "overdue_interest_delay_value_min": integer, // Overdue interest delay value
min
    "overdue_interest_delay_value_max": integer, // Overdue interest delay value
max
    "overdue_tolerance_amount_percent": number, // Overdue tolerance amount
percent
    "overdue_tolerance_amount_fixed": integer, // Overdue tolerance amount fixed
    "interest_method": ["string"], // Interest method
    "repayment_frequency_n": integer, // Repayment frequency N days
    "credit_type": ["string"], // Credit type
    "manual_verification": boolean, // Manual verification
    "visible_on_calculator": boolean, // Visible on calculator
    "is_guarantor_required": boolean, // Guarantor required
    "discount_market_value": number, // Discount amount to market value
    "created_at": "string(date-time)", //created at (like 2024-05-
31T13:27:44.946561+03:00)
    "updated_at": "string(date-time)", //updated at (like 2024-05-
31T13:27:44.946561+03:00)
    "prepayment_min": number, // Prepayment min
    "prepayment_max": number, // Prepayment max
    "merchant_fee": "string", // Merchant fee
    "interest_calculation_period": ["string"], // Interest calculation period
    "only_interest_till_final_disbursement": boolean, // true if only interest
are considered till final disbursement, false otherwise
    "is_tranches_enabled": boolean, // Is tranches enabled
    "min_tranches": integer, // Min tranches
    "max_tranches": integer, // Max tranches
    "disbursement_amount_more_approved": boolean, // Disbursement amount more
approved
    "full_prepayment": boolean, // Full prepayment
    "prepayment_strategy": ["string"], // prepayment strategy; two possible
options - "No change in Installments", "Installment changes" (In the first case,
prepayment does not impact the total loan cost, while in the second case, it
does.)
    "parent_credit_policy": integer, // Parent credit policy
    "client_type": integer, // Client type 1 for new client, 2 for existed
client
    "percent_accrual_schema": integer, // 1 for "The number of days in a month
and a year corresponds to the calendar", 2 for "The number of days in a month
corresponds to a calendar year 360 days", 3 for "The number of days in a month
30 days corresponds to 360 days in a year"
    "repayment_frequency_value": integer, // Repayment frequency; 1 for "Equal
Periods", 2 for "One-time payment"
    "repayment_frequency_unit": integer, // Repayment frequency unit
    "scoring_model": integer // Selected scoring model
}

```

Receive credit policy

This endpoint is used to receive information about existing credit policies using ID. Only lenders can perform such actions.

Endpoint: `/api/lender/credit_policies/{credit_policy_id}/`

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "id": integer, // credit policy id
  "collaterals": [
    {
      "id": integer, // Collateral id
      "collateral_type": {
        "id": integer, // id related to list in lower cell
        "name": "string" // one of the following options: "Deposit", "Goods
        purchased at the expense of a loan", "Real estate", "Automobile", "Guarantee",
        "Property surety", "Financial Guarantee"
        "name_en": "string", // Name [en]
        "name_uk": "string", // Name [uk]
        "name_ru": "string", // Name [ru]
        "name_pl": "string" // Name [pl]
      },
      "monitoring_frequency": integer, // monitoring frequency in days
      "required": boolean // true if collateral is required, false otherwise
    }
  ],
  "verifier_type": {
    "id": integer, // verifier id
    "name": "string", // One the possible options: "Credit committee",
    "Verifier"
    "name_en": "string", // Name [en]
    "name_uk": "string", // Name [uk]
    "name_ru": "string", // Name [ru]
    "name_pl": "string" // Name [pl]
  },
  "name": "string", // Credit policy name
  "regions": ["string"], // List of regions where credit policy is applicable
  "is_prepayment": boolean, // true if there is a prepayment, false otherwise
  "loan_amount_limit_from": number, // loan lower bound
  "loan_amount_limit_to": number, // loan upper bound
  "loan_term_limit_from": number, // loan duration lower bound (days)
  "loan_term_limit_to": number, // loan duration upper bound (days)
  "interest": number, // Interest
  "interest_delay_value": integer, // Interest delay value
  "interest_delay_unit": integer, // Interest delay unit; 1 - days, 2 - weeks,
  3 - months, 4 - years
  "overdue_interest": number, // Overdue interest
  "overdue_interest_delay_value": integer, // Overdue interest delay value
}
```

```

    "overdue_interest_delay_unit": integer, // Overdue interest delay unit; 1 -
days, 2 - weeks, 3 - months, 4 - years
    "overdue_interest_counting_type": ["string"], // Overdue interest counting
type
    "overdue_interest_delay_value_min": integer, // Overdue interest delay value
min
    "overdue_interest_delay_value_max": integer, // Overdue interest delay value
max
    "overdue_tolerance_amount_percent": number, // Overdue tolerance amount
percent
    "overdue_tolerance_amount_fixed": integer, // Overdue tolerance amount fixed
    "interest_method": ["string"], // Interest method
    "repayment_frequency_n": integer, // Repayment frequency N days
    "credit_type": ["string"], // Credit type
    "manual_verification": boolean, // Manual verification
    "visible_on_calculator": boolean, // Visible on calculator
    "is_guarantor_required": boolean, // Guarantor required
    "discount_market_value": number, // Discount amount to market value
    "created_at": "string(date-time)", //created at (like 2024-05-
31T13:27:44.946561+03:00)
    "updated_at": "string(date-time)", //updated at (like 2024-05-
31T13:27:44.946561+03:00)
    "prepayment_min": number, // Prepayment min
    "prepayment_max": number, // Prepayment max
    "merchant_fee": "string", // Merchant fee
    "interest_calculation_period": ["string"], // Interest calculation period
    "only_interest_till_final_disbursement": boolean, // true if only interest
are considered till final disbursement, false otherwise
    "is_tranches_enabled": boolean, // Is tranches enabled
    "min_tranches": integer, // Min tranches
    "max_tranches": integer, // Max tranches
    "disbursement_amount_more_approved": boolean, // Disbursement amount more
approved
    "full_prepayment": boolean, // Full prepayment
    "prepayment_strategy": ["string"], // prepayment strategy; two possible
options - "No change in Installments", "Installment changes" (In the first case,
prepayment does not impact the total loan cost, while in the second case, it
does.)
    "parent_credit_policy": integer, // Parent credit policy
    "client_type": integer, // Client type 1 for new client, 2 for existed
client
    "percent_accrual_schema": integer, // 1 for "The number of days in a month
and a year corresponds to the calendar", 2 for "The number of days in a month
corresponds to a calendar year 360 days", 3 for "The number of days in a month
30 days corresponds to 360 days in a year"
    "repayment_frequency_value": integer, // Repayment frequency; 1 for "Equal
Periods", 2 for "One-time payment"
    "repayment_frequency_unit": integer, // Repayment frequency unit
    "scoring_model": integer // Selected scoring model
}

```

Change credit policy

This endpoint is used to configure existing credit policies. Only lenders can perform such actions.

Endpoint: /api/lender/credit_policies/{credit_policy_id}/

Method: PATCH

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "name": "string", // Credit policy name
  "regions": ["string"], // List of regions where credit policy is applicable
  "is_prepayment": boolean, // true if there is a prepayment, false otherwise
  "loan_amount_limit_from": number, // loan lower bound
  "loan_amount_limit_to": number, // loan upper bound
  "loan_term_limit_from": number, // loan duration lower bound(days)
  "loan_term_limit_to": number, // loan duration upper bound(days)
  "interest": number, // interest value (from 0 to 1)
  "interest_delay_value": integer, // Interest delay value
  "interest_delay_unit": integer, // Interest delay unit; 1 - days, 2 - weeks,
3 - months, 4 - years
  "overdue_interest": number, // Overdue interest
  "overdue_interest_delay_value": integer, // Overdue interest delay value
  "overdue_interest_delay_unit": integer, // Overdue interest delay unit 1 -
days, 2 - weeks, 3 - months, 4 - years
  "overdue_interest_counting_type": ["string"], // Overdue interest counting
type; two possible options - "From First Overdue Date", "From Oldest Late
Payment"
  "overdue_interest_delay_value_min": integer, // Overdue interest delay value
min
  "overdue_interest_delay_value_max": integer, // Overdue interest delay value
max
  "overdue_tolerance_amount_percent": number, // Overdue tolerance amount
percent
  "overdue_tolerance_amount_fixed": integer, // The amount of debt in which
the loan isn't considered overdue
  "interest_method": ["string"], // interest calculation method; 3 possible
options - 1 for "Declining Balance", 2 for "Declining Balance (Equal
Installments)", 3 for "Fixed Flat"
  "repayment_frequency_n": integer, // Repayment frequency; 1 for "Payments
every N days in equal installments (loan body/quantity)", 2 for "Custom dates
and amounts"
  "credit_type": "string", // one of the following options: "Money on card" or
"Consumer finance"
  "manual_verification": boolean, // true if there is manual verification,
false otherwise
  "visible_on_calculator": boolean, // True if visible, false otherwise
  "is_guarantor_required": boolean, // true if guarantor is required, false
otherwise
  "prepayment_min": number, // Prepayment min
  "prepayment_max": number, // Prepayment max
  "merchant_fee": "string", // Merchant fee
  "interest_calculation_period": "string", // Interest calculation period; two
possible options - "After period", "On a Specified Date"
  "only_interest_till_final_disbursement": boolean, // true if only interest
are considered till final disbursement, false otherwise
}
```

```

    "is_tranches_enabled": boolean, // true if tranches are enabled, false
otherwise
    "min_tranches": integer, // minimum amount of tranches
    "max_tranches": integer, // maximum amount of tranches
    "disbursement_amount_more_approved": boolean, // True if only interest is
applied till tranche issuance date
    "full_prepayment": boolean, // true if full prepayment is enabled, false
otherwise
    "prepayment_strategy": "string", // prepayment strategy; two possible
options - "No change in Installments", "Installment changes" (In the first case,
prepayment does not impact the total loan cost, while in the second case, it
does.)
    "parent_credit_policy": integer, // Parent credit policy
    "parent_credit_policy_id": integer, // Parent credit policy id
    "client_type": integer, // Client type 1 for new client, 2 for existed
client
    "percent_accrual_schema": integer, // 1 for "The number of days in a month
and a year corresponds to the calendar", 2 for "The number of days in a month
corresponds to a calendar year 360 days", 3 for "The number of days in a month
30 days corresponds to 360 days in a year"
    "repayment_frequency_value": integer, // Repayment frequency; 1 for "Equal
Periods", 2 for "One-time payment"
    "repayment_frequency_unit": integer, // 1 for "Days", 2 for "Weeks", 3 for
"Month", 4 for "Years"
    "scoring_model": integer, // 1 for new client, 2 for existed client
    "verifier_type": integer // 1 for "Verifier", 2 for "Credit committee"
}

```

Response example:

```

{
  "id": integer, // credit policy id
  "collaterals": [
    {
      "id": integer, // Collateral id
      "collateral_type": {
        "id": integer, // id related to list in lower cell
        "name": "string" // Collateral type (current options - "Deposit",
"Goods purchased at the expense of a loan", "Real estate", "Automobile",
"Guarantee", "Property surety", "Financial Guarantee")
      },
      "monitoring_frequency": integer, // monitoring frequency in days
      "required": boolean // true if collateral is required, false otherwise
    }
  ],
  "verifier_type": {
    "id": integer, // verifier id
    "name": "string", // One the possible options: "Credit committee",
"Verifier".
    "name_en": "string", // Name [en]
    "name_uk": "string", // Name [uk]
    "name_ru": "string", // Name [ru]
    "name_pl": "string" // Name [pl]
  },
  "name": "string", // Credit policy name
  "regions": ["string"], // List of regions where credit policy is applicable
  "is_prepayment": boolean, // true if there is a prepayment, false otherwise
  "loan_amount_limit_from": number, // loan lower bound
}

```

```
"loan_amount_limit_to": number, // loan upper bound
"loan_term_limit_from": number, // loan duration lower bound(days)
"loan_term_limit_to": number, // loan duration upper bound(days)
"interest": number, // Interest
"interest_delay_value": integer, // Interest delay value
"interest_delay_unit": integer, // Interest delay unit; 1 - days, 2 - weeks,
3 - months, 4 - years
"overdue_interest": number, // Overdue interest
"overdue_interest_delay_value": integer, // Overdue interest delay value
"overdue_interest_delay_unit": integer, // Overdue interest delay unit; 1 -
days, 2 - weeks, 3 - months, 4 - years
"overdue_interest_counting_type": ["string"], // Overdue interest counting
type
"overdue_interest_delay_value_min": integer, // Overdue interest delay value
min
"overdue_interest_delay_value_max": integer, // Overdue interest delay value
max
"overdue_tolerance_amount_percent": number, // Overdue tolerance amount
percent
"overdue_tolerance_amount_fixed": integer, // Overdue tolerance amount fixed
"interest_method": ["string"], // Interest method
"repayment_frequency_n": integer, // Repayment frequency N days
"credit_type": ["string"], // Credit type
"manual_verification": boolean, // Manual verification
"visible_on_calculator": boolean, // Visible on calculator
"is_guarantor_required": boolean, // Guarantor required
"discount_market_value": number, // Discount amount to market value
"created_at": "string(date-time)", //created at (like 2024-05-
31T13:27:44.946561+03:00)
"updated_at": "string(date-time)", //updated at (like 2024-05-
31T13:27:44.946561+03:00)
"prepayment_min": number, // Prepayment min
"prepayment_max": number, // Prepayment max
"merchant_fee": "string", // Merchant fee
"interest_calculation_period": ["string"], // Interest calculation period
"only_interest_till_final_disbursement": boolean, // true if only interest
are considered till final disbursement, false otherwise
"is_tranches_enabled": boolean, // Is tranches enabled
"min_tranches": integer, // Min tranches
"max_tranches": integer, // Max tranches
"disbursement_amount_more_approved": boolean, // Disbursement amount more
approved
"full_prepayment": boolean, // Full prepayment
"prepayment_strategy": ["string"], // prepayment strategy; two possible
options - "No change in Installments", "Installment changes" (In the first case,
prepayment does not impact the total loan cost, while in the second case, it
does.)
"parent_credit_policy": integer, // Parent credit policy
"client_type": integer, // Client type 1 for new client, 2 for existed
client
"percent_accrual_schema": integer, // 1 for "The number of days in a month
and a year corresponds to the calendar", 2 for "The number of days in a month
corresponds to a calendar year 360 days", 3 for "The number of days in a month
30 days corresponds to 360 days in a year"
"repayment_frequency_value": integer, // Repayment frequency; 1 for "Equal
Periods", 2 for "One-time payment"
"repayment_frequency_unit": integer, // Repayment frequency unit
```

```
"scoring_model": integer // Selected scoring model
}
```

Delete credit policy

This endpoint is used to delete existing credit policies by ID. Only lenders can perform such actions.

Endpoint: `/api/lender/credit_policies/{credit_policy_id}/`

Method: **DELETE**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (response **status code** must be **204** if credit policy was successfully deleted)

User

Get current user

This endpoint is used to get information about the current user. Mostly it's called to get and save the user id for further steps.

Endpoint: `/api/users/current/`

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "id": integer, //User id
  "email": "string", //User email (like
[addressname]@[domain].[top_domain])      "last_login": "string(date-time)", //
Date of last login (like 2024-05-31T13:27:44.946561+03:00)
  "joined_at": "string(date-time)", // Date of user creation (like 2024-05-
31T13:27:44.946561+03:00)
  "last_name": "Last_name", //User last name
}
```

```

"first_name": "Name", //User first name
"language": "en", //one of the following options: "en","uk", "ru", "pl"
"groups": [
  {
    "name": "Client" //One of the following options: "Admin", "Client","
Commerce"," Lead"," Lead:Middle"," Lead:Soft", "Lead: Verifier", "Lender",
"Manager", "Verifier"
  }
],
"permissions": [
  {
    "id": integer, // Permission id
    "name": "string", // One of the following options: "Collection", "Scoring
model", "Scoring card", "Decision rules", "Application form customization",
"Credit products", "Vendor management" , "Assignment officer", "Loan migration",
"Import payments", "Blacklist", "Archive", "System administration", "Reports [
Decision report ]", "Reports [ Verifier report ]", "Reports [ All reports ]",
"Branch Management", "User KPI", "Export", "Transactions", "Loan servicing",
"Customer management", "Collateral management", "Underwriting", "Loan
origination"
    "name_en": "string", // permission name in english
    "name_uk": "string", // permission name in ukrainian
    "name_ru": "string", // permission name in russian
    "name_pl": "string" // permission name in polish
  }
],
"role": [
  {
    "id": integer, // Id related to the list in lower cell
    "name": "string", // one of the following options: "Collector",
"Administrator", "Credit Administrator", "Risk Manager", "Verifier", "Analyst"
    "name_en": "string", // role name in english
    "name_uk": "string", // role name in ukrainian
    "name_ru": "string", // role name in russian
    "name_pl": "string", // role name in polish
    "permissions": [integer] // List of permissions id
  }
],
"branch": [
  {
    "id": integer, // Branch id
    "name": "string", // Branch name
    "city": "string", // Branch city
    "description": "string", // Description
    "number": "string", // Branch number
    "permissions": [integer] // List of permissions id
  }
],

"permissions_scope": ["string"],
"is_social": boolean, // true if user has authorized via social network
"user_flow": "string" // current status of application
}

```

Client application

Create client application

This endpoint is used to create a client application (a shop request must be created before calling this endpoint).

Endpoint: `/api/client/applications/`

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "current_dpr": number, // Current dividend payout ratio in percent
  "checks": {
    "proxy_check": boolean // checks for Tor, Proxy
  },
  "risk_score": number, // Risk probability
  "reject_reason": "string", //Reject reason
  "loan_date_from": "string(date-time)", // Loan start date (like 2024-05-31T13:27:44.946561+03:00)
  "loan_date_to": "string(date-time)", // Loan end date (like 2024-05-31T13:27:44.946561+03:00)
  "prepayment": number, // Prepayment value
  "loan_amount": number, // Loan amount value
  "amount_to_pay": number, // Amount to pay including accrual
  "amount_to_pay_discount": number, // Amount to pay with discount
  "overdue_days": integer, // number of overdue days
  "max_overdue_days": integer, // maximum number of overdue days
  "overdue_payments_count": integer, // number of overdue payments
  "approve": boolean, // true if approved by verifier or committee, false otherwise
  "prepayment_status": "string", // one of following options: "Without prepayment", "Not paid", "Paid"
  "date_interest_start": "string(date)", // date of interest accrual beginning
  "closing_description": "string", // Closing description
  "shop_request": "string(uuid)", // Shop request in uuid format
  "user": integer, // User id
  "assigned_to": integer, // The user ID to which the application is assigned
  "status": integer, // 1 for "Created", 2 for "Underwriting", 3 for "Admin wait", 4 for "Contract signing", 5 for "Loan: Accepted", 6 for "Loan: Issued", 7 for "Loan: Closed", 8 for "Loan: Overdue", 9 for "Loan: Rejected", 10 for "Not Taken Up", 11 for "Waiting for guarantor", 12 for "Waiting for card", 13 for "Waiting for collateral", 14 for "Loan: Partially Disbursed", 15 for "Loan: Terminated" 16 for "Loan: Write Off"
  "risk_segment": integer, // Risk segment id
  "card": integer, // The card to which money will be transferred
}
```

```
"discounts": [integer] // List of discounts id
}
```

Response example:

```
{
  "id": integer, // Application id
  "current_dpr": number, // Current dividend payout ratio in percent
  "checks": {
    "proxy_check": boolean // checks for Tor, Proxy
  },
  "credit_type": " string ", // one of the following options: "Money on the
card" or "Consumer finance"
  "invoice": "string",
  "repayment_changes": [
    {
      "id": integer, // repayment change request id
      "approved_change": [
        {
          "id": integer, // approval id
          "created_at": "string(date-time)", // approval date
          (like 2024-05-31T13:27:44.946561+03:00)
          "repayment_change": integer, // repayment change request id
          "document": integer // additional contract id
        }
      ],
      "requested_by": [
        {
          "id": integer, // user id that made repayment change request
          "groups": [] // group to which the user belongs
        }
      ],
      "created_at": "string(date-time)", // date of repayment change
      creation (like 2024-05-31T13:27:44.946561+03:00)
      "updated_at": "string(date-time)", // date of repayment change
      update (like 2024-05-31T13:27:44.946561+03:00)
      "status": "string", // repayment change status
      "reason": "string", // reason of repayment change request
      "type": "string", // repayment change type; one of the following
      options: "CREATED", "CONTRACT_SIGNING", "APPROVED", REJECTED
      "params": JSON // check Payment schedule
      "additional_interest": number, // additional interest for the
      repayment change
      "application": integer // application id
    }
  ],
  "questionary": [
    {
      "id": integer, // questionary id
      "answered_on_questions": {} // JSON that contains question as a
      field name and field value as an answer to that question (check Application form
      )
      "created_at": "2024-06-04T12:35:52.442536+03:00", // date of
      creation (like 2024-05-31T13:27:44.946561+03:00)
      "selected_template": integer, // template id
      "application": integer, // application id
    }
  ]
}
```

```

        "user": integer // user id
    }
],
"risk_score": number, // Risk probability
"reject_reason": "string", //Reject reason
"loan_date_from": "string(date-time)", // Loan start date (like 2024-05-
31T13:27:44.946561+03:00)
"loan_date_to": "string(date-time)", // Loan end date (like 2024-05-
31T13:27:44.946561+03:00)
"prepayment": number, // Prepayment value
"loan_amount": number, // Loan amount value
"amount_to_pay": number, // Amount to pay including accrual
"amount_to_pay_discount": number, // Amount to pay with discount
"overdue_days": integer, // number of overdue days
"max_overdue_days": integer, // maximum number of overdue days
"overdue_payments_count": integer, // number of overdue payments
"discount": number, // Discount value in percentage
"approve": boolean, // true if approved by verifier or committee, false
otherwise
"created_at": "string(date-time)", // date of creation (like 2024-05-
31T13:27:44.946561+03:00)
"updated_at": "string(date-time)", // date of update (like 2024-05-
31T13:27:44.946561+03:00)
"prepayment_status": "string", // one of following options: "Without
prepayment", "Not paid", "Paid"
"date_interest_start": "string(date)", // date of interest accrual beginning
"closing_description": "string", // Closing description
"shop_request": "string(uuid)", // Shop request in uuid format
"user": id, // User id
"assigned_to": integer, // The user ID to which the application is assigned
"status": integer, // 1 for "Created", 2 for" Underwriting", 3 for "Admin
wait", 4 for "Contract signing", 5 for "Loan: Accepted", 6 for "Loan: Issued", 7
for "Loan: Closed", 8 for "Loan: Overdue", 9 for "Loan: Rejected", 10 for "Not
Taken Up", 11 for "Waiting for guarantor", 12 for "Waiting for card", 13 for
"Waiting for collateral", 14 for "Loan: Partially Disbursed", 15 for "Loan:
Terminated" 16 for "Loan: Write Off"
"risk_segment": integer, // Risk segment id
"card": integer, // The card to which money will be transferred
"discounts": [integer] // List of discounts id
}

```

Get/change/delete the application

This endpoint is used to get/change/delete information about the application depending on the method.

Endpoint: `/api/client/applications/{application_id}`

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "id": integer, // Application id
  "assigned_to": {
    "id": integer, // The user ID to whom the application is assigned
    "email": "string", // The user email to whom the application is assigned
    (like [addressname]@[domain].[top_domain])
  },
  "credit_type": "string", // one of the following options: "Money on the card"
  or "Consumer finance"
  "invoice": "string", // Invoice
  "repayment_changes": [
    {
      "id": integer, // repayment change request id
      "approved_change": [
        {
          "id": integer, // approval id
          "created_at": "string(date-time)", // approval date (like
          2024-05-31T13:27:44.946561+03:00)
          "repayment_change": integer, // repayment change request id
          "document": integer // additional contract id
        }
      ],
      "requested_by": [
        {
          "id": integer, // user id that made repayment change request
          "groups": [] // group to which the user belongs
        }
      ],
      "created_at": "string(date-time)", // date of repayment change
      creation (like 2024-05-31T13:27:44.946561+03:00)
      "updated_at": "string(date-time)", // date of repayment change update
      (like 2024-05-31T13:27:44.946561+03:00)
      "status": "string", // repayment change status
      "reason": "string", // reason of repayment change request
      "type": "string", // repayment change type; one of the following
      options: "CREATED", "CONTRACT_SIGNING", "APPROVED", REJECTED
      "params": {} // check Payment schedule
      "additional_interest": number, // additional interest in percentage
      "application": integer // application id
    }
  ],
  "questionary": [
    {
      "id": integer, // questionary id
      "answered_on_questions": JSON, JSON that contains question as a
      field name and field value as an answer to that question (check Application form
      )
      "created_at": "string(date-time)", // date of creation (like 2024-
      05-31T13:27:44.946561+03:00)
      "selected_template": integer, // template id
      "application": integer, // application id
    }
  ]
}
```

```

        "user": integer // user id
    }
],
"tranches": [
    {
        "amount": "string(decimal)", // tranche amount
        "date": "string(date)" // date of tranche issuance
    }
],
"risk_score": number, // Risk probability
"current_dpr": number, // Current dividend payout ratio in percent
"reject_reason": "string", // Reject reason
"loan_date_from": "string(date-time)", // Loan start date (like 2024-05-31T13:27:44.946561+03:00)
"loan_date_to": "string(date-time)", // Loan end date (like 2024-05-31T13:27:44.946561+03:00)
"prepayment": number, // Prepayment amount
"loan_amout": number, // Loan amount
"amount_to_pay": number, // Amount to pay including accrual
"amount_to_pay_discount": number, // Amount to pay with discount
"overdue_days": integer, // Number of overdue days
"max_overdue_days": integer, // Maximum number of overdue days
"overdue_payments_count": integer, // Number of overdue payments
"checks": {}, // Checks for Tor, Proxy
"discount": number, // Discount value in percentage
"approve": boolean, // True if approved, false otherwise
"created_at": "string(date-time)", // Creation date (like 2024-05-31T13:27:44.946561+03:00)
"updated_at": "string(date-time)", // Update date (like 2024-05-31T13:27:44.946561+03:00)
"prepayment_status": "string", // one of the following options: "Without prepayment", "Not paid", "Paid"
"date_interest_start": "string(date)", // Date of interest accrual beginning (like 2024-05-31T13:27:44.946561+03:00)
"shop_request": {
    "uuid": "string", // Shop request id in uuid format
    "region": "string", // Region name
    "product_name": "string", // Product name
    "product_id": "string", // Product id
    "product_price": number, // Product price
    "client_name": "string", // Client name
    "client_phone": "string", // Client phone
    "loan_term": integer, // Loan term in days
    "promo_code": "string", // Promo code
    "created_at": "string(date-time)", // Creation date (like 2024-05-31T13:27:44.946561+03:00)
    "updated_at": "string(date-time)", // Update date (like 2024-05-31T13:27:44.946561+03:00)
    "prepayment_amount": integer, // Prepayment amount
    "shop": integer, // Shop id
    "credit_policy": integer, // Credit policy id
    "category": integer // Category id
},
"status": {
    "id": integer, // Status id
    "name": "string", // Status name in English
    "name_en": "string", // Status name in English

```

```

    "name_uk": "string", // Status name in Ukrainian
    "name_ru": "string", // Status name in Russian
    "name_pl": "string" // Status name in Polish
  },
  "risk_segment": {
    "id": integer, // Risk segment id
    "name": "string", // Risk segment name
    "is_active": boolean, // True if active, false otherwise
    "score_from": number, // Score lower bound
    "score_to": number, // Score upper bound
    "default_probability_from": number, // Probability lower bound
    "default_probability_to": number, // Probability upper bound
    "expected_distribution": number, // Expected number of applications per risk
group
    "recomendation": "string", // Recommendation message
    "recomendation_en": "string", // Recommendation message in english
    "recomendation_uk": "string", // Recommendation message in ukrainian
    "recomendation_ru": "string", // Recommendation message in russian
    "recomendation_pl": "string", // Recommendation message in polish
    "action": "string", // one of the following options: "Approve",
"Reject", "Underwriting"
    "action_en": "string", // Action name in english
    "action_uk": "string", // Action name in ukrainian
    "action_ru": "string", // Action name in russian
    "action_pl": "string", // Action name in polish
    "created_at": "string(date-time)", // Creation date (like 2024-05-
31T13:27:44.946561+03:00)
    "updated_time": "string(date-time)", // Update date (like 2024-05-
31T13:27:44.946561+03:00)
    "scoring_model": integer // 1 for new client, 2 for existing client
  },
  "card": {
    "id": integer, // Card id
    "card_number": "string", // Card number
    "cardholder_name": "string", // Cardholder name
    "created_at": "string", // Creation date
    "user": integer // User id
  },
  "discounts": [
    {
      "id": integer, // Discount id
      "type": "string", // one of the following options: "Promo code",
"Loyalty program", "Early discount"
      "is_active": boolean, // True if discount is active, false otherwise
      "discount": number, // Discount value
      "start_date": "string(date-time)", // Discount start date (like
2024-05-31T13:27:44.946561+03:00)
      "end_date": "string(date-time)", // Discount end date (like 2024-05-
31T13:27:44.946561+03:00)
      "description": "string", // Discount description
      "credit_policies": [integer], // List of credit policies id to which
discount is applied
      "categories": [integer] // List of categories id to which discount
is applied
    },
  ],
}

```

Method: PATCH

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "current_dpr": number, // Current dividend payout ratio in percent
  "checks": {
    "proxy_check": boolean // checks for Tor, Proxy
  },
  "risk_score": number, // Risk probability
  "reject_reason": "string", //Reject reason
  "loan_date_from": "string(date-time)", // Loan start date (like 2024-05-31T13:27:44.946561+03:00)
  "loan_date_to": "string(date-time)", // Loan end date (like 2024-05-31T13:27:44.946561+03:00)
  "prepayment": number, // Prepayment value
  "loan_amount": number, // Loan amount value
  "amount_to_pay": number, // Amount to pay including accrual
  "amount_to_pay_discount": number, // Amount to pay with discount
  "overdue_days": integer, // number of overdue days
  "max_overdue_days": integer, // maximum number of overdue days
  "overdue_payments_count": integer, // number of overdue payments
  "approve": boolean, // true if approved by verifier or committee, false otherwise
  "prepayment_status": "string", // one of the following options: "Without prepayment", "Not paid", "Paid"
  "date_interest_start": "string(date)", // date of interest accrual beginning
  "closing_description": "string", // Closing description
  "shop_request": "string(uuid)", // Shop request in uuid format
  "user": integer, // User id
  "assigned_to": integer, // The user ID to which the application is assigned
  "status": integer, // 1 for "Created", 2 for "Underwriting", 3 for "Admin wait", 4 for "Contract signing", 5 for "Loan: Accepted", 6 for "Loan: Issued", 7 for "Loan: Closed", 8 for "Loan: Overdue", 9 for "Loan: Rejected", 10 for "Not Taken Up", 11 for "Waiting for guarantor", 12 for "Waiting for card", 13 for "Waiting for collateral", 14 for "Loan: Partially Disbursed", 15 for "Loan: Terminated" 16 for "Loan: Write Off"
  "risk_segment": integer, // Risk segment id
  "card": integer, // The card to which money will be transferred
  "discounts": [integer] // List of discounts id
}
```

Response example:

```
{
  "id": integer, // Application id
  "assigned_to": {
    "id": integer, // The user ID to whom the application is assigned
    "email": "string", // The user email to whom the application is assigned (like [addressname]@[domain].[top_domain])
  },
}
```

```

"credit_type": "string", // one of the following options: "Money on the card"
or "Consumer finance"
"invoice": "string", // Invoice
"repayment_changes": [
  {
    "id": integer, // repayment change request id
    "approved_change": [
      {
        "id": integer, // approval id
        "created_at": "string(date-time)", // approval date (like
2024-05-31T13:27:44.946561+03:00)
        "repayment_change": integer, // repayment change request id
        "document": integer // additional contract id
      }
    ],
    "requested_by": [
      {
        "id": integer, // user id that made repayment change request
        "groups": [] // group to which the user belongs
      }
    ],
    "created_at": "string(date-time)", // date of repayment change
creation (like 2024-05-31T13:27:44.946561+03:00)
    "updated_at": "string(date-time)", // date of repayment change
update (like 2024-05-31T13:27:44.946561+03:00)
    "status": "string", // repayment change status
    "reason": "string", // reason of repayment change request
    "type": "string", // repayment change type; one of the following
options: "CREATED", "CONTRACT_SIGNING", "APPROVED", REJECTED
    "params": JSON
    "additional_interest": number, // additional interest in percentage
    "application": integer // application id
  }
],
"questionary": [
  {
    "id": integer, // questionary id
    "answered_on_questions": {} // JSON that contains question as a
field name and field value as an answer to that question
    "created_at": "string(date-time)", // date of creation (like 2024-
05-31T13:27:44.946561+03:00)
    "selected_template": integer, // template id
    "application": integer, // application id
    "user": integer // user id
  }
],
"tranches": [
  {
    "amount": "string(decimal)", // tranche amount
    "date": "string(date)" // date of tranche issuance
  }
],
"risk_score": number, // Risk probability
"current_dpr": number, // Current dividend payout ratio in percent
"reject_reason": "string", // Reject reason

```

```
"loan_date_from": "string(date-time)", // Loan start date (like 2024-05-31T13:27:44.946561+03:00)
"loan_date_to": "string(date-time)", // Loan end date (like 2024-05-31T13:27:44.946561+03:00)
"prepayment": number, // Prepayment amount
"loan_amout": number, // Loan amount
"amount_to_pay": number, // Amount to pay including accrual
"amount_to_pay_discount": number, // Amount to pay with discount
"overdue_days": integer, // Number of overdue days
"max_overdue_days": integer, // Maximum number of overdue days
"overdue_payments_count": integer, // Number of overdue payments
"checks": {}, // Checks for Tor, Proxy
"discount": number, // Discount value
"approve": boolean, // True if approved, false otherwise
"created_at": "string(date-time)", // Creation date (like 2024-05-31T13:27:44.946561+03:00)
"updated_at": "string(date-time)", // Update date (like 2024-05-31T13:27:44.946561+03:00)
"prepayment_status": "string", // one of the following options: "Without prepayment", "Not paid", "Paid"
"date_interest_start": "string(date)", // Date of interest accrual beginning
"shop_request": {
  "uuid": "string", // Shop request id in uuid format
  "region": "string", // Region name
  "product_name": "string", // Product name
  "product_id": "string", // Product id
  "product_price": number, // Product price
  "client_name": "string", // Client name
  "client_phone": "string", // Client phone
  "loan_term": integer, // Loan term in days
  "promo_code": "string", // Promo code
  "created_at": "string(date-time)", // Creation date (like 2024-05-31T13:27:44.946561+03:00)
  "updated_at": "string(date-time)", // Update date (like 2024-05-31T13:27:44.946561+03:00)
  "prepayment_amount": integer, // Prepayment amount
  "shop": integer, // Shop id
  "credit_policy": integer, // Credit policy id
  "category": integer // Category id
},
"status": {
  "id": integer, // Status id
  "name": "string", // Status name in English
  "name_en": "string", // Status name in English
  "name_uk": "string", // Status name in Ukrainian
  "name_ru": "string", // Status name in Russian
  "name_pl": "string" // Status name in Polish
},
"risk_segment": {
  "id": integer, // Risk segment id
  "name": "string", // Risk segment name
  "is_active": boolean, // True if active, false otherwise
  "score_from": number, // Score lower bound
  "score_to": number, // Score upper bound
  "default_probability_from": number, // Probability lower bound
  "default_probability_to": number, // Probability upper bound
```

```

    "expected_distribution": number, // Expected number of applications per risk
group
    "recomendation": "string", // Recommendation message
    "recomendation_en": "string", // Recommendation message in english
    "recomendation_uk": "string", // Recommendation message in ukrainian
    "recomendation_ru": "string", // Recommendation message in russian
    "recomendation_pl": "string", // Recommendation message in polish
    "action": "string", // one of the following options: "Approve",
"Reject", "Underwriting"
    "action_en": "string", // Action name in english
    "action_uk": "string", // Action name in ukrainian
    "action_ru": "string", // Action name in russian
    "action_pl": "string", // Action name in polish
    "created_at": "string(date-time)", // Creation date (like 2024-05-
31T13:27:44.946561+03:00)
    "updated_time": "string(date-time)", // Update date (like 2024-05-
31T13:27:44.946561+03:00)
    "scoring_model": integer // 1 for new client, 2 for existing client
},
"card": {
    "id": integer, // Card id
    "card_number": "string", // Card number
    "cardholder_name": "string", // Cardholder name
    "created_at": "string", // Creation date
    "user": integer // User id
},
"discounts": [
    {
        "id": integer, // Discount id
        "type": "string", // one of the following options: "Promo code",
"Loyalty program", "Early discount"
        "is_active": boolean, // True if discount is active, false otherwise
        "discount": number, // Discount value
        "start_date": "string(date-time)", // Discount start date (like
2024-05-31T13:27:44.946561+03:00)
        "end_date": "string(date-time)", // Discount end date (like 2024-05-
31T13:27:44.946561+03:00)
        "description": "string", // Discount description
        "credit_policies": [integer], // List of credit policies id to which
discount is applied
        "categories": [integer] // List of categories id to which discount
is applied
    },
}
}

```

Method: **DELETE**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (response **status code** must be **204** if the application was successfully deleted)

Approve client application

The verifier uses this endpoint to approve client applications.

Endpoint: */api/client/applications/{application_id}/approve/*

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

same as for **GET method** (if the application was successfully approved field approve should return the value true)

Get application dashboard

The lender uses this endpoint to get the application dashboard where the dashboard means whole information about the application including user info, shop request info, discounts, loan terms, amount, status, risk segment, etc.

Endpoint: */api/lender/dashboard/applications/{application_id}/*

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "id": integer, // Application id
  "assigned_to": {
    "id": integer, // user id
    "email": "string", //User email (like [addressname]@[domain].[top_domain])
    "first_name": "string", // user first name
    "last_name": "string", // user last name
    "rating": number // user rating
  },
}
```

```

"user": {
  "id": integer, // user id
  "email": "string", //User email (like [addressname]@[domain].[top_domain])
  "first_name": "string", // user first name
  "last_name": "string", // user last name
  "rating": number // user rating
},
"questionary": [
  {
    "id": integer, // questionary id
    "answered_on_questions": JSON, JSON that contains question as a
field name and field value as an answer to that question (check Application
form)
    "created_at": "string(date-time)", // date of creation (like 2024-
05-31T13:27:44.946561+03:00)
    "selected_template": integer, // template id
    "application": integer, // application id
    "user": integer // user id
  }
],
"voting": "string", // credit committee vote
"score_card_scores": "string", // customer score (risk rating)
"repayment_changes": [
  {
    "id": integer, // repayment change request id
    "approved_change": [
      {
        "id": integer, // approval id
        "created_at": "string(date-time)", // approval date (like
2024-05-31T13:27:44.946561+03:00)
        "repayment_change": integer, // repayment change request id
        "document": integer // additional contract id
      }
    ],
    "requested_by": [
      {
        "id": integer, // user id that made repayment change request
        "groups": [] // group to which the user belongs
      }
    ],
    "created_at": "string(date-time)", // date of repayment change
creation (like 2024-05-31T13:27:44.946561+03:00)
    "updated_at": "string(date-time)", // date of repayment change
update (like 2024-05-31T13:27:44.946561+03:00)
    "status": "string", // repayment change status
    "reason": "string", // reason of repayment change request
    "type": "string", // repayment change type; one of the following
options: "CREATED", "CONTRACT_SIGNING", "APPROVED", REJECTED
    "params": JSON
    "additional_interest": number, // additiona interest
    "application": integer // application id
  }
],
"risk_score": number, // Risk probability
"current_dpr": number, // Current dividend payout ratio in percent

```

```

    "reject_reason": "string", // Reject reason
    "loan_date_from": "string(date-time)", // Loan start date (like 2024-05-31T13:27:44.946561+03:00)
    "loan_date_to": "string(date-time)", // Loan end date (like 2024-05-31T13:27:44.946561+03:00)
    "prepayment": number, // Prepayment amount
    "loan_amout": number, // Loan amount
    "amount_to_pay": number, // Amount to pay including accrual
    "amount_to_pay_discount": number, // Amount to pay with discount
    "overdue_days": integer, // Number of overdue days
    "max_overdue_days": integer, // Maximum number of overdue days
    "overdue_payments_count": integer, // Number of overdue payments
    "checks": {}, // Checks for Tor, Proxy
    "discount": number, // Discount value
    "approve": boolean, // True if approved, false otherwise
    "created_at": "string(date-time)", // Creation date (like 2024-05-31T13:27:44.946561+03:00)
    "updated_at": "string(date-time)", // Update date (like 2024-05-31T13:27:44.946561+03:00)
    "prepayment_status": "string", // one of following options: "Without prepayment", "Not paid", "Paid"
    "date_interest_start": "string(date)", // Date of interest accrual beginning
    "shop_request": {
        "uuid": "string", // Shop request id in uuid format
        "region": "string", // Region name
        "product_name": "string", // Product name
        "product_id": "string", // Product id
        "product_price": number, // Product price
        "client_name": "string", // Client name
        "client_phone": "string", // Client phone
        "loan_term": integer, // Loan term in days
        "promo_code": "string", // Promo code
        "created_at": "string(date-time)", // Creation date (like 2024-05-31T13:27:44.946561+03:00)
        "updated_at": "string(date-time)", // Update date (like 2024-05-31T13:27:44.946561+03:00)
        "prepayment_amount": integer, // Prepayment amount
        "shop": integer, // Shop id
        "credit_policy": integer, // Credit policy id
        "category": integer // Category id
    },
    "status": {
        "id": integer, // 1 for "Created", 2 for "Underwriting", 3 for "Admin wait", 4 for "Contract signing", 5 for "Loan: Accepted", 6 for "Loan: Issued", 7 for "Loan: Closed", 8 for "Loan: Overdue", 9 for "Loan: Rejected", 10 for "Not Taken Up", 11 for "Waiting for guarantor", 12 for "Waiting for card", 13 for "Waiting for collateral"
        "name": "string", // Status name in English
        "name_en": "string", // Status name in English
        "name_uk": "string", // Status name in Ukrainian
        "name_ru": "string", // Status name in Russian
        "name_pl": "string" // Status name in Polish
    },
    "risk_segment": {
        "id": integer, // Risk segment id
        "name": "string", // Risk segment name
        "is_active": boolean, // True if active, false otherwise

```

```

"score_from": number, // Score lower bound
"score_to": number, // Score upper bound
"default_probability_from": number, // Probability lower bound
"default_probability_to": number, // Probability upper bound
"expected_distribution": number, // Expected number of applications per risk
group
"recommendation": "string", // Recommendation message
"recommendation_en": "string", // Recommendation message in english
"recommendation_uk": "string", // Recommendation message in ukrainian
"recommendation_ru": "string", // Recommendation message in russian
"recommendation_pl": "string", // Recommendation message in polish
"action": "string", // one of the following options: "Approve",
"Reject", "Underwriting"
"action_en": "string", // Action name in english
"action_uk": "string", // Action name in ukrainian
"action_ru": "string", // Action name in russian
"action_pl": "string", // Action name in polish
"created_at": "string(date-time)", // Creation date (like 2024-05-
31T13:27:44.946561+03:00)
"updated_time": "string(date-time)", // Update date (like 2024-05-
31T13:27:44.946561+03:00)
"scoring_model": integer // 1 for new client, 2 for existing client
},
"card": {
  "id": integer, // Card id
  "card_number": "string", // Card number
  "cardholder_name": "string", // Cardholder name
  "created_at": "string", // Creation date
  "user": integer // User id
},
"discounts": [
  {
    "id": integer, // Discount id
    "type": "string", // one of the following options: "Promo code",
"Loyalty program", "Early discount"
    "is_active": boolean, // True if discount is active, false otherwise
    "params": {
      "code": "string", // code name; if type == "Promo code"
      "loans_count_from": integer, // closed loan number lower bound;
If type== "Loyalty program"
      "loans_count_to": integer, // closed loan number lower bound;
If type== "Loyalty program"
      "min_loan_amount": float, // loan amount lower bound; If
type== "On early payment"
      "max_loan_amount": float, // loan amount upper bound; If
type== "On early payment"
      "days": integer // number of days; If type== "On early payment"
    }
    "discount": number, // Discount value
    "start_date": "string(date-time)", // Discount start date (like
2024-05-31T13:27:44.946561+03:00)
    "end_date": "string(date-time)", // Discount end date (like 2024-05-
31T13:27:44.946561+03:00)
    "description": "string", // Discount description

```

```
        "credit_policies": [integer], // List of credit policies id to which
discount is applied
        "categories": [integer] // List of categories id to which discount
is applied
    ],
}
```

Application form

This endpoint is used to get or change information about a client like gender, position, education, etc. There are two types of questionnaires - individual and entity (hence, different mandatory questions). There also is an opportunity to add custom questions, which are not included in this response description.

Endpoint: `/api/questionnaires/client/questionnaire/answer/{questionnaire_id}/`

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "id": integer, // questionnaire id
  "answered_on_questions": {
    "email": "string", //User email (like
[addressname]@[domain].[top_domain])
    "gender": "string", // User gender; one of the following options:
"Male", "Female"; if questionnaire type is "individual"
    "position": "string", User position; one of the following options:
"None", "Worker", "Military", "Specialist", "Middle manager", "Senior manager",
"Owner or co-owner", "Other"; if the questionnaire type is "individual"
    "education": "string", User education; one of the following options:
"Incomplete secondary", "Secondary", "Secondary-special", "Incomplete higher",
"Higher", "Second higher", "MBA"; if the questionnaire type is "individual"
    "last_name": "string", // User last name; used both for "individual" and
"entity" questionnaire types
    "birth_date": "string(date)", // User birth date; if the questionnaire
type is "individual"
    "first_name": "string", // User first name; used both for "individual"
and "entity" questionnaire types
    "phone_number": "string", // User phone number; used both for
"individual" and "entity" questionnaire types
    "marital_status": "string", // User marital status; one of the following
options: "Single", "Married", "Divorced", "Civil marriage", "Widow(er)"; if the
questionnaire type is "individual"
```

```

    "employment_type": "string", // User employment type; one of the
following options: "Fulltime", "Partial", "Freelance", "Temporary",
"Unemployed", "Retired", "Student", "Maternity leave", "Self-employed",
"Unemployed"; if the questionnaire type is "individual"
    "employment_duration": integer, // User employment duration in months;
if the questionnaire type is "individual"
    "children_under_18_year": integer, Number of children under 18 years
old; if questionnaire type is "individual"
    "identify_document_code": "string(decimal)", // User identify document
code; if questionnaire type is "individual"
    "seniority_years": "string", // Number of years of seniority of the user
    "address ": "string", // company address; if questionnaire type is
"entity"
    "country": "string", // company location country; if questionnaire type
is "entity"
    "Office_phone_number": "string", // office phone number; if
questionnaire type is "entity"
    "industry": "string", // industry of the company; if questionnaire type
is "entity"
    "Enterprise_name": "string", company name; if questionnaire type is
"entity"
    "employees_count": "string", // number of employees; if questionnaire
type is "entity"
    "Registration_number": "string", // company registration number; if
questionnaire type is "entity"
    }
}

```

Method: **PATCH**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```

{
    "email": "string", //User email (like
[addressname]@[domain].[top_domain])
    "gender": "string", // User gender; one of the following options:
"Male", "Female"; if questionnaire type is "individual"
    "position": "string", User position; one of the following options:
"None", "Worker", "Military", "Specialist", "Middle manager", "Senior manager",
"Owner or co-owner", "Other"; if the questionnaire type is "individual"
    "education": "string", User education; one of the following options:
"Incomplete secondary", "Secondary", "Secondary-special", "Incomplete higher",
"Higher", "Second higher", "MBA"; if the questionnaire type is "individual"
    "last_name": "string", // User last name; used both for "individual" and
"entity"questionnaire types
    "birth_date": "string(date)", // User birth date; if the questionnaire
type is "individual"
    "first_name": "string", // User first name; used both for "individual"
and "entity"questionnaire types

```

```

    "phone_number": "string", // User phone number; used both for
"individual" and "entity" questionnaire types
    "marital_status": "string", // User marital status; one of the following
options: "Single", "Married", "Divorced", "Civil marriage", "Widow(er)"; if the
questionnaire type is "individual"
    "employment_type": "string", // User employment type; one of the
following options: "Fulltime", "Partial", "Freelance", "Temporary",
"Unemployed", "Retired", "Student", "Maternity leave", "Self-employed",
"Unemployed"; if the questionnaire type is "individual"
    "employment_duration": integer, // User employment duration in months;
if the questionnaire type is "individual"
    "children_under_18_year": integer, Number of children under 18 years
old; if questionnaire type is "individual"
    "identify_document_code": "string(decimal)", // User identify document
code; if questionnaire type is "individual"
    "seniority_years": "string", // Number of years of seniority of the user
    "address": "string", // company address; if questionnaire type is
"entity"
    "country": "string", // company location country; if questionnaire type
is "entity"
    "Office_phone_number": "string", // office phone number; if
questionnaire type is "entity"
    "industry": "string", // industry of the company; if questionnaire type
is "entity"
    "Enterprise_name": "string", company name; if questionnaire type is
"entity"
    "employees_count": "string", // number of employees; if questionnaire
type is "entity"
    "Registration_number": "string", // company resgistration number; if
questionnaire type is "entity"
}

```

Response example:

same as for GET method

Bank card

This endpoint is used to specify details about the card for a given user.

Endpoint: `/api/client/{user_id}/bank_card/`

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```

{
  "application_id": integer, // Application id
  "card_number": "string", // Card number

```

```
"cardholder_name": "string", // Cardholder name
"user": integer // User id
}
```

Response example:

```
{
  "application_id": integer, // Application id
  "card_number": "string", // Card number
  "cardholder_name": "string", // Cardholder name
  "user": integer // User id
}
```

Payment

Get payment info

This endpoint is used for getting payment information so the user can make a payment using the order id.

Endpoint: /api/platon/request_payment_info/

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "order": integer, // Order id
  "email": "string", //User email (like [addressname]@[domain].[top_domain])
  "phone": "string", // User phone
  "first_name": "string", // User first name
  "last_name": "string" // User last name
}
```

Make payment

The user uses this endpoint to make a payment. At this moment our system is integrated with Ukrainian payment system Platon, but can easily be integrated with other payment system. Hence, request/response body may be different.

Endpoint: /api/platon/payment_callback/

Method: POST

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "date": "string(date-time)", // Date of payment (like 2024-05-31T13:27:44.946561+03:00)
  "id": "string", // External platon ID in XXXXX-XXXXX-XXXXX format
  "card": integer, // User card
  "amount": number, // Payment amount
  "commission": number, // Commission value
  "card_token": "string", // Plato's payment system variable
  "order": integer // Order id (received from Get payment info endpoint)
}
```

Response example:

Empty (**status code** must be **200** if payment was successful)

Get repayment table

The user uses this endpoint to get the repayment table.

Endpoint: `/api/payment-schedule/repayment_table/{application_id}/`

Method: GET

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
[
  {
    "id": integer, // payment id in the system
    "status": "string", // payment status; 1 for "Scheduled", 2 for "Paid",
    3 for "Overdued"
    "date": "string(date)", // date of payment
    "principal": "string", // principal amount
    "interest": "string", // interest amount
    "commission": "string", // commission amount
    "paid": "string", // paid amount
  }
]
```

```
    "created_at": "string(date-time)", // creation date (like 2024-05-31T13:27:44.946561+03:00)
    "updated_at": "string(date-time)", // update date (like 2024-05-31T13:27:44.946561+03:00)
    "loan": integer // loan id
  },
]
```

Underwriting

Get/send the underwriting department

We use this endpoint to receive or send information about underwriting departments.

Endpoint: /api/loan_routing/underwriter_departments/

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
[
  {
    "id": integer, // underwriting id
    "verifiers": [
      {
        "id": integer, // verifier id
        "user": {
          "id": integer, // user id
          "email": "string", //User email (like [addressname]@[domain].[top_domain])
          "last_name": "string", // user last name
          "first_name": "string" // user first name
        },
        "loan_amount_from": number, // loan lower bound
        "loan_amount_to": number, // loan upper bound
        "fact_addr_regions": ["string"], // List of regions where given verifiers can make decisions
        "processed_applications": integer, // number of processed applications
        "credit_policies": [integer], // List of credit policies id
        "shops": [integer] // List of shop id
      }
    ],
    "supervisor": {
```

```

        "id": integer, // supervisor id
        "email": "string", // supervisor email (like
[addressname]@[domain].[top_domain])
        "last_name": "string", // supervisor last name
        "first_name": "string" // supervisor first name
    },
    "supervisor_deputy": {
        "id": integer, // supervisor deputy id
        "email": "string", // supervisor deputy email (like
[addressname]@[domain].[top_domain])
        "last_name": "string", // supervisor deputy last name
        "first_name": "string" // supervisor deputy first name
    },
    "name": "string" // underwriting department name
},
]

```

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```

{
    "verifiers": [integer], // List of verifiers id
    "supervisor": integer, // supervisor id
    "supervisor_deputy": integer, // supervisor deputy id
    "name": "string" // underwriting name
}

```

Response example:

```

[
    {
        "id": integer, // underwriting id
        "verifiers": [
            {
                "id": integer, // verifier id
                "user": {
                    "id": integer, // user id
                    "email": "string", //User email (like
[addressname]@[domain].[top_domain])
                    "last_name": "string", // user last name
                    "first_name": "string" // user first name
                },
                "loan_amount_from": number, // loan lower bound
                "loan_amount_to": number, // loan upper bound
                "fact_addr_regions": ["string"], // List of regions where given
verifiers can make decisions
                "processed_applications": integer, // number of processed
applications
                "credit_policies": [integer], // List of credit policies id
                "shops": [integer] // List of shop id
            }
        ]
    }
]

```

```

    ],
    "supervisor": {
      "id": integer, // supervisor id
      "email": "string", // supervisor email (like
[addressname]@[domain].[top_domain])
      "last_name": "string", // supervisor last name
      "first_name": "string" // supervisor first name
    },
    "supervisor_deputy": {
      "id": integer, // supervisor deputy id
      "email": "string", // supervisor deputy email (like
[addressname]@[domain].[top_domain])
      "last_name": "string", // supervisor deputy last name
      "first_name": "string" // supervisor deputy first name
    },
    "name": "string" // underwriting department name
  },
]

```

Get/change/delete the underwriting department by ID

We use this endpoint to receive or change information about underwriting departments by given id.

Endpoint: `/api/loan_routing/underwriter_departments/{id}/`

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```

[
  {
    "id": integer, // underwriting id
    "verifiers": [
      {
        "id": integer, // verifier id
        "user": {
          "id": integer, // user id
          "email": "string", //User email (like
[addressname]@[domain].[top_domain])
          "last_name": "string", // user last name
          "first_name": "string" // user first name
        },
        "loan_amount_from": number, // loan lower bound
        "loan_amount_to": number, // loan upper bound
        "fact_addr_regions": ["string"], // List of regions where given
verifiers can make decisions
      }
    ]
  }
]

```

```

        "processed_applications": integer, // number of processed
applications
        "credit_policies": [integer], // List of credit policies id
        "shops": [integer] // List of shop id
    }
],
"supervisor": {
    "id": integer, // supervisor id
    "email": "string", // supervisor email (like
[addressname]@[domain].[top_domain])
    "last_name": "string", // supervisor last name
    "first_name": "string" // supervisor first name
},
"supervisor_deputy": {
    "id": integer, // supervisor deputy id
    "email": "string", // supervisor deputy email (like
[addressname]@[domain].[top_domain])
    "last_name": "string", // supervisor deputy last name
    "first_name": "string" // supervisor deputy first name
},
"name": "string" // underwriting department name
},
]

```

Method: PATCH

Headers

Authorization: Token

Content-Type: application/json

Request body:

```

{
    "verifiers": [integer], // List of verifiers id
    "supervisor": integer, // supervisor id
    "supervisor_deputy": integer, // supervisor deputy id
    "name": "string" // underwriting name
}

```

Response example:

```

[
    {
        "id": integer, // underwriting id
        "verifiers": [
            {
                "id": integer, // verifier id
                "user": {
                    "id": integer, // user id
                    "email": "string", //User email (like
[addressname]@[domain].[top_domain])
                    "last_name": "string", // user last name
                    "first_name": "string" // user first name
                },
                "loan_amount_from": number, // loan lower bound
                "loan_amount_to": number, // loan upper bound
            }
        ]
    }
]

```

```

        "fact_addr_regions": ["string"], // List of regions where given
verifiers can make decisions
        "processed_applications": integer, // number of processed
applications
        "credit_policies": [integer], // List of credit policies id
        "shops": [integer] // List of shop id
    }
},
"supervisor": {
    "id": integer, // supervisor id
    "email": "string", // supervisor email (like
[addressname]@[domain].[top_domain])
    "last_name": "string", // supervisor last name
    "first_name": "string" // supervisor first name
},
"supervisor_deputy": {
    "id": integer, // supervisor deputy id
    "email": "string", // supervisor deputy email (like
[addressname]@[domain].[top_domain])
    "last_name": "string", // supervisor deputy last name
    "first_name": "string" // supervisor deputy first name
},
"name": "string" // underwriting department name
},
]

```

Method: DELETE

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (response **status code** must be **204** if underwriting department was successfully deleted)

Guarantor

Get/Create a guarantor

This endpoint is used to create or get a current guarantor.

Endpoint: </api/client/guarantors/>

Method: GET

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "guarantor": [
    {
      "uuid": "string", // Guarantor id in uuid4 format
      "email": "string", // Guarantor email (like
[addressname]@[domain].[top_domain])
      "phone_number": "string", // Guarantor phone number
      "first_name": "string", // Guarantor first name
      "last_name": "string", // Guarantor last name
      "middle_name": "string", // Guarantor middle name
      "status": "string", // Status name; one of the following options:
"Invitation sent", "Registration", "Contract is signed"
      "passport_type": "string", // Passport type; one of the following options:
"ID card", "Series"
      "passport": "string", // Passport
      "date_of_issue": "string(date-time)", // Date of passport issuing (like
2024-05-31T13:27:44.946561+03:00)
      "issued_by": "string", // Name of department that issued the passport
      "identification_number": "string", // Identification number
      "created_at": "string(date-time)", // Creation date (like 2024-05-
31T13:27:44.946561+03:00)
      "updated_at": "string(date-time)", // Update date (like 2024-05-
31T13:27:44.946561+03:00)
      "user": integer, // User id
      "application": integer // Application id
    }
  ]
}
```

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "guarantor": [
    {
      "uuid": "string", // Guarantor id in uuid4 format
      "email": "string", // Guarantor email (like
[addressname]@[domain].[top_domain])
      "phone_number": "string", // Guarantor phone number
      "first_name": "string", // Guarantor first name
      "last_name": "string", // Guarantor last name
    }
  ]
}
```

```

    "middle_name": "string", // Guarantor middle name
    "status": "string", // Status name; one of the following options:
    "Invitation sent", "Registration", "Contract is signed"
    "passport_type": "string", // Passport type; one of the following options:
    "ID card", "Series"
    "passport": "string", // Passport
    "date_of_issue": "string(date-time)", // Date of passport issuing (like
2024-05-31T13:27:44.946561+03:00)
    "issued_by": "string", // Name of department that issued the passport
    "identification_number": "string", // Identification number
    "created_at": "string(date-time)", // Creation date (like 2024-05-
31T13:27:44.946561+03:00)
    "updated_at": "string(date-time)", // Update date (like 2024-05-
31T13:27:44.946561+03:00)
    "user": integer, // User id
    "application": integer // Application id
  }
]
}

```

Response example:

```

{
  "guarantor": [
    {
      "uuid": "string", // Guarantor id in uuid4 format
      "email": "string", // Guarantor email (like
[addressname]@[domain].[top_domain])
      "phone_number": "string", // Guarantor phone number
      "first_name": "string", // Guarantor first name
      "last_name": "string", // Guarantor last name
      "middle_name": "string", // Guarantor middle name
      "status": "string", // Status name; one of the following options:
      "Invitation sent", "Registration", "Contract is signed"
      "passport_type": "string", // Passport type; one of the following options:
      "ID card", "Series"
      "passport": "string", // Passport
      "date_of_issue": "string(date-time)", // Date of passport issuing (like
2024-05-31T13:27:44.946561+03:00)
      "issued_by": "string", // Name of department that issued the passport
      "identification_number": "string", // Identification number
      "created_at": "string(date-time)", // Creation date (like 2024-05-
31T13:27:44.946561+03:00)
      "updated_at": "string(date-time)", // Update date (like 2024-05-
31T13:27:44.946561+03:00)
      "user": integer, // User id
      "application": integer // Application id
    }
  ]
}

```

Register a guarantor

This endpoint is used to register current guarantor.

Endpoint: /api/client/guarantors/{guarantor_uuid}/register/

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "email": "string", // guarantor email (like
[addressname]@[domain].[top_domain])
  "password": "string", // guarantor password
  "phone_number": "string", // guarantor phone
  "first_name": "string", // guarantor first name
  "last_name": "string" // guarantor last name
}
```

Response example:

Empty (**status code** must be **201** if guarantor was successfully registered)

Send the guarantor the SMS

This endpoint is used to send the guarantor the SMS.

Endpoint: /api/smsclub/send_guarantor_sms/

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "guarantor_id": "string" // guarantor id in uuid4 format
}
```

Response example:

Empty (**status code** must be **201** if sms code was successfully sent)

Check the SMS

This endpoint is used to check the SMS code. If the code is valid then the contract is signed.

Endpoint: /api/smsclub/check_guarantor_sms/

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "guarantor_id": "string", // guarantor id in uuid4 format
  "sms_code": "string" // SMS code received via phone
}
```

Response example:

Empty (**status code** must be **200** if sms code is correct)

Discount

Create discount

This endpoint is used to create discounts.

Endpoint: */api/discount/discounts/*

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "type": "string", // discount type; one of the following options: "Promo Code", "Loyalty program", "Early discount"
  "is_active": boolean, // true if active, false otherwise
  "params": {
    "code": "string", // code name; if type == "Promo code"
    "loans_count_from": integer, // The lower bound of the range of closed loans in which the discount is applied; If type== "Loyalty program";
    "loans_count_to": integer, // The upper bound of the range of closed loans in which the discount is applied; If type== "Loyalty program";
    "min_loan_amount": float, // The lower bound of the loan amount in which the discount is applied; If type== "Early discount";
    "max_loan_amount": float, // The lower bound of the range of closed loans in which the discount is applied; If type== "Early discount";
    "days": integer // The number of days since receiving the loan to qualify for the discount; If type== "Early discount"
  },
  "discount": number, // discount value
}
```

```

    "start_date": "string(date-time)", // discount start date (like 2024-05-31T13:27:44.946561+03:00)
    "end_date": "string(date-time)", // discount end date (like 2024-05-31T13:27:44.946561+03:00)
    "description": "string", // discount description
    "credit_policies": [integer], // list of credit policies id to which discount is applicable
    "categories": [integer] // list of categories id to which discount is applicable
}

```

Response example:

```

{
  "id": integer, // discount id;
  "type": "string", // discount type; one of the following options: "Promo Code", "Loyalty program", "Early discount"
  "is_active": boolean, // true if active, false otherwise
  "params": {
    "code": "string", // code name; if type == "Promo code"
    "loans_count_from": integer, // The lower bound of the range of closed loans in which the discount is applied;
    If type== "Loyalty program";
    "loans_count_to": integer, // The upper bound of the range of closed loans in which the discount is applied;
    If type== "Loyalty program";
    "min_loan_amount": float, // The lower bound of the loan amount in which the discount is applied;
    If type== "Early discount";
    "max_loan_amount": float, // The lower bound of the range of closed loans in which the discount is applied;
    If type== "Early discount";
    "days": integer // The number of days since receiving the loan to qualify for the discount;
    If type== "Early discount"
  },
  "discount": number, // discount value
  "start_date": "string(date-time)", // discount start date (like 2024-05-31T13:27:44.946561+03:00)
  "end_date": "string(date-time)", // discount end date (like 2024-05-31T13:27:44.946561+03:00)
  "description": "string", // description
  "credit_policies": [integer], // list of credit policies id to which discount is applicable
  "categories": [integer] // list of categories id to which discount is applicable
}

```

Delete discount

This endpoint is used to delete discounts.

Endpoint: `/api/discount/discounts/{discount_id}/`

Method: DELETE

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (response **status code** must be **204** if the discount was successfully deleted)

Commission

Create commission

This endpoint is used to create commissions.

Endpoint: `/api/lender/credit_policies/{credit_policy_id}/commissions/`

Method: POST

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "name": "string", // Commission name
  "amount": number, // commission value
  "type": integer, // one of the following options: 1 for "One-time", 2 for
"Periodic"
  "reason": integer, // commission reason; one of the following options: 1 for
"On loan issuance", 2 for "On loan close", 3 for "On loan overdue", 4 for "On
Loan Early Repayment"; this field has to be present only if type == 1
  "periodicity": integer, // commission periodicity; one of the following
options: 1 for "Every day", 2 for "Every week", 3 for "Every month", 4 for
"Every quarter", 5 for "Every year"; this field has to be present only if type
== 2
  "subtype": integer // commission calculation type; one of the following
options: 1 for "Fixed", 2 for "As a percentage of the credit limit", 3 for "As a
percentage of the loan amount", 4 for "A percentage of the pre-paid amount"
}
```

Response example:

```
{
```

```
"id": integer, // commission id
"name": "string", // commission name
"amount": number, // commission value; depending on the commission type is
considered whether flat amount or
percentage / 100
"created_at": "string(date-time)", // date of commission creation (like
2024-05-31T13:27:44.946561+03:00)
"updated_at": "string(date-time)", // date of commission update (like 2024-
05-31T13:27:44.946561+03:00)
"credit_policy": integer, // credit policy id to which commission is applied
"type": integer, // one of the following options: 1 for "One-time", 2 for
"Periodic"
"reason": integer, // reason; one of the following options: 1 for "On loan
issuance", 2 for "On loan close", 3 for "On loan overdue", 4 for "On Loan Early
Repayment";
"subtype": integer, // commission calculation type; one of the following
options: 1 for "Fixed", 2 for "As a percentage of the credit limit", 3 for "As a
percentage of the loan amount", 4 for "A percentage of the pre-paid amount"
"periodicity": integer // commission periodicity; one of the following
options: 1 for "Every day", 2 for "Every week", 3 for "Every month", 4 for
"Every quarter", 5 for "Every year";
}
```

Delete commission

This endpoint is used to delete commissions.

Endpoint: `/api/lender/credit_policies/{credit_policy_id}/commissions/{commission_id}/`

Method: **DELETE**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (response **status code** must be **204** if the commission was successfully deleted)

Payment schedule

Precalculation

This endpoint is used to perform the precalculation of the payment schedule with given input parameters.

Endpoint: `/api/payment-schedule/calculate/`

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "credit_policy_id": integer, // credit policy id
  "amount": number, // loan amount
  "loan_term": integer, // Loan term in days
  "category_id": integer, // product category id
  "promo_code": "string" // promocode value
}
```

Response example:

```
[{
  "id": integer, // payment id in the system
  "status": "string", // payment status; 1 for "Scheduled", 2 for "Paid",
  3 for "Overdued"
  "date": "string", // date of payment
  "principal": "string", // principal amount
  "interest": "string", // interest amount
  "commission": "string", // commission amount
  "paid": "string", // paid amount
  "created_at": "string(date-time)", // creation date (like 2024-05-
  31T13:27:44.946561+03:00)
  "updated_at": "string(date-time)", // update date (like 2024-05-
  31T13:27:44.946561+03:00)
  "loan": integer // loan id
}]
```

Get repayment table

The user uses this endpoint to get the repayment table.

Endpoint: `/api/payment-schedule/repayment_table/{application_id}/`

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
[{
  "id": integer, // payment id in the system
  "status": "string", // payment status; 1 for "Scheduled", 2 for "Paid",
  3 for "Overdued"
}
```

```
"date": "string", // date of payment
"principal": "string", // principal amount
"interest": "string", // interest amount
"commission": "string", // commission amount
"paid": "string", // paid amount
"created_at": "string(date-time)", // creation date (like 2024-05-31T13:27:44.946561+03:00)
"updated_at": "string(date-time)", // update date (like 2024-05-31T13:27:44.946561+03:00)
"loan": integer // loan id
}]
```

Make repayment change request

The user uses this endpoint to make a repayment change request.

Endpoint: </api/payment-schedule/repayment-change/>

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
[{
  "type": "string", // repayment change request type; one of the following
options: "DECREASE_DATE_OF_PAYMENT", "DECREASE_AMOUNT_OF_PAYMENTS",
"EXTEND_LOAN", "FREEZE_LOAN"
  "params": {
    "date_of_payment": "string(date)", // new date of payment; this field is
present if type == "DECREASE_DATE_OF_PAYMENT"
    "number_of_payments": integer, // number of payment with reduced amount;
this field is present if type == "DECREASE_AMOUNT_OF_PAYMENTS"
    "amount": number, // new amount for payments with reduced principal;
this field is present if type == "DECREASE_AMOUNT_OF_PAYMENTS"
    "extension_periods": integer, // number of additional payments; this
field is present if type == "EXTEND_LOAN"
    "freeze_to_date": "string(date)" // date of loan freezing; this field is
present if type == "FREEZE_LOAN"
  },
  "additional_interest": number, // additional interest
  "application": integer // application id
}]
```

Response example:

Same as for precalculation but regarding repayment change request

Approve repayment change request

The lender uses this endpoint to approve a repayment change request. To receive a repayment change request ID make an application info request and extract value from field `repayment_changes[0].id` (check [Client application](#))

Endpoint: `/api/payment-schedule/{repayment_change_request_id}/approve/`

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (**status code** must be **200** if the request was successfully approved)

Contract

Create contract template

This endpoint is used to create contract templates. There are 4 types of contracts: Invoice, Contract, Additional contract, and Guarantor.

Endpoint: `/api/lender/contract_templates/`

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
{
  "credit_type": "string", // credit type; one of the following options:
  "Money on the card", "Consumer finance"
  "contract": "string", // contract text
  "invoice": "string", // invoice text
  "additional_contract": "string", // additional contract text
  "guarantor_contract": "string", // guarantor contract text
  "credit_policy": [integer] // List of credit policies id to which this
  contract template is related
}
```

Response example:

```
{
  "id": integer, // contract template id
  "credit_type": "string", // credit type; one of the following options:
  "Money on the card", "Consumer finance"
  "contract": "string", // contract text
}
```

```
"invoice": "string", // invoice text
"additional_contract": "string", // additional contract text
"guarantor_contract": "string", // guarantor contract text
"created_at": "string(date-time)", // date of contract template creation
(like 2024-05-31T13:27:44.946561+03:00)
"updated_at": "string(date-time)", // date of contract template update (like
2024-05-31T13:27:44.946561+03:00)
"credit_policy": [integer] // List of credit policies id to which this
contract template is related
}
```

Receive contract

This endpoint is used to get contracts related to the application. The same endpoint is used to receive all types of contracts, to specify contract type add in the end document_type={contract_type} (Invoice for Invoice, Contract for Contract, Additional+contract for Additional contract, Guarantor+contract for Guarantor).

Endpoint: `/api/client/applications/{application_id}/documents/?document_type={contract_type}/`

Method: GET

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
[
  {
    "id": integer, // contract id
    "application": integer, // application id
    "document_type": "string", document type; on of the following options:
    "Invoice", "Contract", "Additional contract", "Guarantor contract"
    "document": "string", // contract text
    "signed": boolean, // true if contract was signed, false otherwise
    "created_at": "string(date-time)" // date of contract creation (like
    2024-05-31T13:27:44.946561+03:00)
  }
]
```

Sign additional contract

This endpoint is used to sign additional contracts after the repayment change request is approved by the lender. To receive an approved change payment request id make an application info request and extract value from the field repayment_changes[0].approved_change.id (check [Client application](#))

Endpoint: `/api/payment-schedule/{approved_change_payment_request_id}/sign/`

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (**status code** must be **200** if the contract was successfully signed)

Delete additional contract

This endpoint is used to delete contract template.

Endpoint: */api/lender/contract_templates/{contract_template_id}/*

Method: **DELETE**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (response **status code** must be **204** if the contract template was successfully deleted)

Tranches

Receive client tranches

A user uses this endpoint to receive (check [Credit policy](#)).

Endpoint: */api/tranches/client/*

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
[
  {
    "id": integer, // tranche id
    "application": integer, // application id
    "amount": number, // tranche amount
    "date": "string" // date of tranche issuance
  }
]
```

Configure client tranches

A user uses this endpoint to configure tranches on his side, but tranches have to be enabled in the credit policy (check [Credit policy](#)).

Endpoint: */api/tranches/client/*

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```
[
  {
    "application": integer, // application id
    "amount": number, // tranche amount
    "date": "string" // date of tranche issuance
  }
]
```

Response example:

```
[
  {
    "id": integer, // tranche id
    "application": integer, // application id
    "amount": number, // tranche amount
    "date": "string" // date of tranche issuance
  }
]
```

Receive tranches regarding the application

A user uses this endpoint to get tranches regarding the application (check [Credit policy](#)).

Endpoint: */api/client/applications/{application_id}/tranches/*

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
[
  {
    "amount": number, // tranche amount
    "date": "string" // date of tranche issuance
  }
]
```

Scoring model

Get scorecard

This endpoint is used to receive information about the scorecard.

Endpoint: </api/lender/scorecard/>

Method: **GET**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

```
{
  "id": integer, // feature id
  "categories": [
    {
      "id": integer, // category id
      "name": "string", // category name
      "score": integer // category score
    }
  ],
  "name": "string", // feature name
  "categories_editable": boolean, // true if categories are editable, false otherwise
  "created_at": "string(date-time)", // date of feature creation (like 2024-05-31T13:27:44.946561+03:00)
  "updated_at": "string(date-time)", // date of feature update (like 2024-05-31T13:27:44.946561+03:00)
}
```

```
"card_type": "string" // card type; one of the following options -
"individual" or "entity"
}
```

Configure scorecard

This endpoint is used to configure certain scorecards.

Endpoint: `/api/lender/scorecard/{id}/`

Method: **PUT**

Headers

Authorization: Token

Content-Type: application/json

Request body:

In order to configure certain features you need to store information into the categories field. For example:

```
{
  "id": 5,
  "categories": [
    {
      "id": 1, // categories id
      "name": "Middle manager",
      "score": 5
    },
    {
      "id": 2,
      "name": "Military",
      "score": 1
    },
    {
      "id": 3,
      "name": "None",
      "score": 70
    },
    {
      "id": 4,
      "name": "Other",
      "score": 40
    },
    {
      "id": 5,
      "name": "Owner or co-owner",
      "score": 0
    },
    {
      "id": 6,
      "name": "Senior manager",
      "score": 0
    },
  ],
}
```

```

    {
      "id": 7,
      "name": "Specialist",
      "score": 5
    },
    {
      "id": 8,
      "name": "Worker",
      "score": 17
    }
  ],
  "name": "Position",
  "categories_editable": boolean,
  "created_at": "2022-09-06T15:58:42.724638+03:00",
  "updated_at": "2023-08-08T17:41:39.026083+03:00",
  "card_type": "individual"
}

```

Response example:

```

{
  "id": integer, // feature id
  "categories": [
    {
      "id": integer, // category id
      "name": "string", // category name
      "score": integer // category score
    }
  ],
  "name": "string", // feature name
  "categories_editable": boolean, // true if categories are editable, false otherwise
  "created_at": "string(date-time)", // date of feature creation (like 2024-05-31T13:27:44.946561+03:00)
  "updated_at": "string(date-time)", // date of feature update (like 2024-05-31T13:27:44.946561+03:00)
  "card_type": "string" // card type; one of the following options - "individual" or "entity"
}

```

Create risk segment

This endpoint is used to create a risk segment.

Endpoint: /api/lender/scorecard/risk_segments/

Method: **POST**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```

{
  "name": "middle risk segment", // risk segment name
  "is_active": boolean, // true if risk segment is active, false otherwise
  "score_from": number, // risk segment score lower bound
  "score_to": number, // risk segment score upper bound
  "default_probability_from": number, // default probability lower bound
  "default_probability_to": number, // default probability upper bound
  "expected_distribution": number, // expected distribution
  "recomendation": "Underwriting", // recomendation message related to risk
segment
  "action": "Underwriting", // action type; one of the following options -
"Approve", "Reject", "Underwriting"
  "scoring_model": integer // scoring model id
}

```

Response example:

```

{
  "id": integer // risk segment id
  "name": "middle risk segment", // risk segment name
  "is_active": boolean, // true if risk segment is active, false otherwise
  "score_from": number, // risk segment score lower bound
  "score_to": number, // risk segment score upper bound
  "default_probability_from": number, // default probability lower bound
  "default_probability_to": number, // default probability upper bound
  "expected_distribution": number, // expected distribution
  "recomendation": "Underwriting", // recomendation message related to risk
segment
  "action": "Underwriting", // action type; one of the following options -
"Approve", "Reject", "Underwriting"
  "created_at": "string(date-time)", // date of creation (like 2024-05-
31T13:27:44.946561+03:00)
  "updated_at": "string(date-time)", // date of update (like 2024-05-
31T13:27:44.946561+03:00)
  "scoring_model": integer // scoring model id
}

```

Update risk segment

This endpoint is used to create a risk segment.

Endpoint: `/api/lender/scorecard/risk_segments/{id}/`

Method: **PATCH**

Headers

Authorization: Token

Content-Type: application/json

Request body:

```

{
  "name": "middle risk segment", // risk segment name
  "is_active": boolean, // true if risk segment is active, false otherwise
  "score_from": number, // risk segment score lower bound
  "score_to": number, // risk segment score upper bound

```

```
"default_probability_from": number, // default probability lower bound
"default_probability_to": number, // default probability upper bound
"expected_distribution": number, // expected distribution
"recomendation": "Underwriting", // recomendation message related to risk
segment
  "action": "Underwriting", // action type; one of the following options -
"Approve", "Reject", "Underwriting"
  "scoring_model": integer // scoring model id
}
```

Response example:

```
{
  "id": integer // risk segment id
  "name": "middle risk segment", // risk segment name
  "is_active": boolean, // true if risk segment is active, false otherwise
  "score_from": number, // risk segment score lower bound
  "score_to": number, // risk segment score upper bound
  "default_probability_from": number, // default probability lower bound
  "default_probability_to": number, // default probability upper bound
  "expected_distribution": number, // expected distribution
  "recomendation": "Underwriting", // recomendation message related to risk
segment
  "action": "Underwriting", // action type; one of the following options -
"Approve", "Reject", "Underwriting"
  "created_at": "string(date-time)", // date of creation (like 2024-05-
31T13:27:44.946561+03:00)
  "updated_at": "string(date-time)", // date of update (like 2024-05-
31T13:27:44.946561+03:00)
  "scoring_model": integer // scoring model id
}
```

Delete risk segment

This endpoint is used to delete the risk segment.

Endpoint: `/api/lender/scorecard/risk_segments/{risk_segment_id}/`

Method: **DELETE**

Headers

Authorization: Token

Content-Type: application/json

Request body:

Empty

Response example:

Empty (response **status code** must be **204** if risk segment was successfully deleted)