BITCOIN-SECURED LENDING API DESIGN

SOLUTIONS ARCHITECT PORTFOLIO

dustinwsweet@gmail.com

https://www.linkedin.com/in/dustin-sweet/

1 Overview

The API design documented herein enables the firm to offer a frequently requested new capability to its partners: the ability for their end-users to obtain a USD loan against their Bitcoin position.

We start by providing insight into the business opportunity, followed by a high-level overview of the proposed solution. Finally, a detailed presentation of all API methods and resources.

2 Design Motivations & Rationale

2.1 Input from Sales & Marketing

Results from our highly anticipated inaugural Net Promoter Score (NPS) survey have arrived. Engagement is excellent. The firm has received high marks from our integrating partners in several areas tied directly to our Key Performance Indicators (KPIs) and high-level business goals. Here is a standout verbatim:

- *"The quality of the partner implementation team is second-to-none. I am humbled to collaborate with such kind people. We hit a few tough problems during the build, but through the chaos, they remained laser-focused, resulting in the finished product flourishing with our customers!"*

The survey has also identified vital opportunities. Of all partner respondents, variations of the following two issues were the most commonly cited:

- *"Our banking customers are clamoring for a lending solution in which they can use their cryptocurrency as collateral. They want the same simplicity and trust provided by the Buy-Hold-Sell BTC program they've come to love."*
- *"Our system integrators need to move faster. Your implementation team is amazing, but product feature gaps and difficulty understanding how best to implement caused delays. We request that you dedicate a technical expert to the project. If so, we would be able to slash our time-to-market and more rapidly bring mutual revenue through customer attachment."*

2.2 Product Research Recommendations

With competitors having already launched their product offerings, it is imperative that the firm quickly bring a lending solution to market for its banking and FI partners. Of note, several of our competitors have erred strategically by forging ahead without due process concerning regulatory clarity. This oversight has resulted in critical legal setbacks, a consequence upon which we can capitalize. The opportunity is ripe for the firm to assert its dominance further and cement its position as the industry leader in regulatory compliance related to Bitcoin and cryptocurrency.

Our product team has researched the following two lending program options:

1) **Secured Personal Loan**. This product disperses funds in a lump sum at origination and makes monthly payments at a fixed interest rate and repayment timeline. Bitcoin collateral held throughout the life of the loan is returned once the lienee has paid the loan in full.
2) **Secure Line of Credit**. A revolving credit line wherein the borrower can access funds at will, with variable interest rates to be paid on the portion of the credit line used. This product automatically adds and removes Bitcoin collateral as the borrower disperses and settles funds.

In collaboration with legal & regulatory compliance, the team recommends proceeding with option #1 (Secured Personal Loan) for our initial product offering, citing the following:

- Regarding the LoC, regulatory guidance continues to evolve rapidly in key jurisdictions, and several unknowns may impede our legal path to market (among them emerging legislation) and the implications of multiple disbursements, taxes, and variable interest rates). Conversely, the regulatory pathway is clear at this time to release a secured personal loan to partners serving customers in most US states.
- From a product development standpoint, option #1 is simpler to implement and is quicker to bring to market with partner integrations. Notably, this option requires fewer transactions to hit the daily settlement & reconciliation process, a key consideration when weighing options.

2.3 Decisions

The goal remains a conservative approach - lead by establishing brand recognition and winning the customers' loyalty *first*. Then, follow on with additional lending program options (like LoC) once the regulatory waters are clear. We feel confident that the peace of mind this more straightforward offering will bring to our partners and customers will outweigh the drawbacks of forgoing the more flexible LoC option.

In addition, we have determined that we can move forward on a secured loan program for our partners' *business* customers. This product, as compared to the secured personal loan, will offer higher loan limits and longer terms at a more attractive fixed interest rate. We recommend backlogging this work for immediate implementation after the personal loan project is in general availability (GA).

2.3.1 Monetization

Upon origination of each loan, a fee (percent TBD) applied to the total loan amount will be charged to the partner and paid within the next business day's standard settlement and reconciliation process. The partner shall not pass any origination fees, annual fees, or prepayment penalties to the end customers.

If liquidation is required (e.g., the partner sells collateral to recover loss due to non-payment), the firm will automatically deduct and retain 5% of the proceeds from the sale (at market price). The remainder of the funds will be issued to the partner within 24 hours via the standard settlement and reconciliation process.

2.3.2 Future-Proofing

This generic API design facilitates the rapid introduction of future lending use cases. All references to the BTC cryptocurrency symbol are abstracted to allow for extending loans backed by different collateral types (e.g., ETH, XLM, ADA). Once the firm can offer these alternative currencies, the design will be ready to accommodate as-is.

3 API Design

⚠️ Assumptions & Caveats

1) Authentication: no new requirements. This API will be secured using the same mutual TLS (mTLS) mechanism used today by the primary partner API suite.

2) Identification of Partners (and Sub-Partners). The partner's identity is directly discoverable to the Web API service through the context of the API call via existing means, either from the caller's authorized IP address, some aspect of the caller's TLS certificate, or the inclusion of special HTTP headers. Accordingly, all mention of partner IDs is out of scope and excluded from this design. Moreover, the tenancy of each partner is a walled garden. Therefore the identifiers for all business objects of the Lending API need to be unique only per partner (as opposed to globally, across all partners).

3) Artifacts. The implementing partners will need to facilitate the archival and access by the end customer of various legal documents required to acquire a loan (e.g., Terms & Conditions, Agreements & Disclosures, Privacy Policy). The existing partner API/product can manage all requirements needed to support the workflows introduced by the new Lending API detailed herein.

4) Notifications & Alerts. The implementing partners will send their end customers various notifications and messages (SMS & email), including confirmations, receipts, payment reminders, and monthly statements. The existing API provides our partners with a means to support these scenarios. Therefore, all notification aspects are out of scope and omitted from this design.

5) Customer/Account. The firm will decide whether each partner's customer is eligible to participate in our various lending programs based on multiple regulatory factors, including (but not limited to) state of residence, tax arrears status, and other factors. **This design intentionally abstracts & conflates the customer/account entity to simplify the presentation of the lending API aspects, which is tangential.** Critically, given only an account ID, the firm has sufficient context to make an eligibility determination using the existing features of our partner API/product suite.

6) Settlement & Reconciliation. The partner will pay a fee to the firm within 24 hours of the origination of each loan. In addition, whenever the partner must liquidate a customer's collateral (e.g., due to non-payment, default, excessive LTV), the firm must facilitate the sale (at market value) and provide the proceeds back to the partner. The existing out-of-band and secure daily settlement & reconciliation process will accommodate these use cases without modification.

7) Modern Frameworks. Several Lending API methods are long-running due to the time-intensive activities of our backend microservices. Examples include loan provisioning, de-provisioning, and cryptocurrency transfers between the customer's wallet and their secure collateral vault wallet. The following operations are affected:

- `CreateLoan`
- `Add Collateral`
- `Liquidate Collateral`
- `Close Loan`

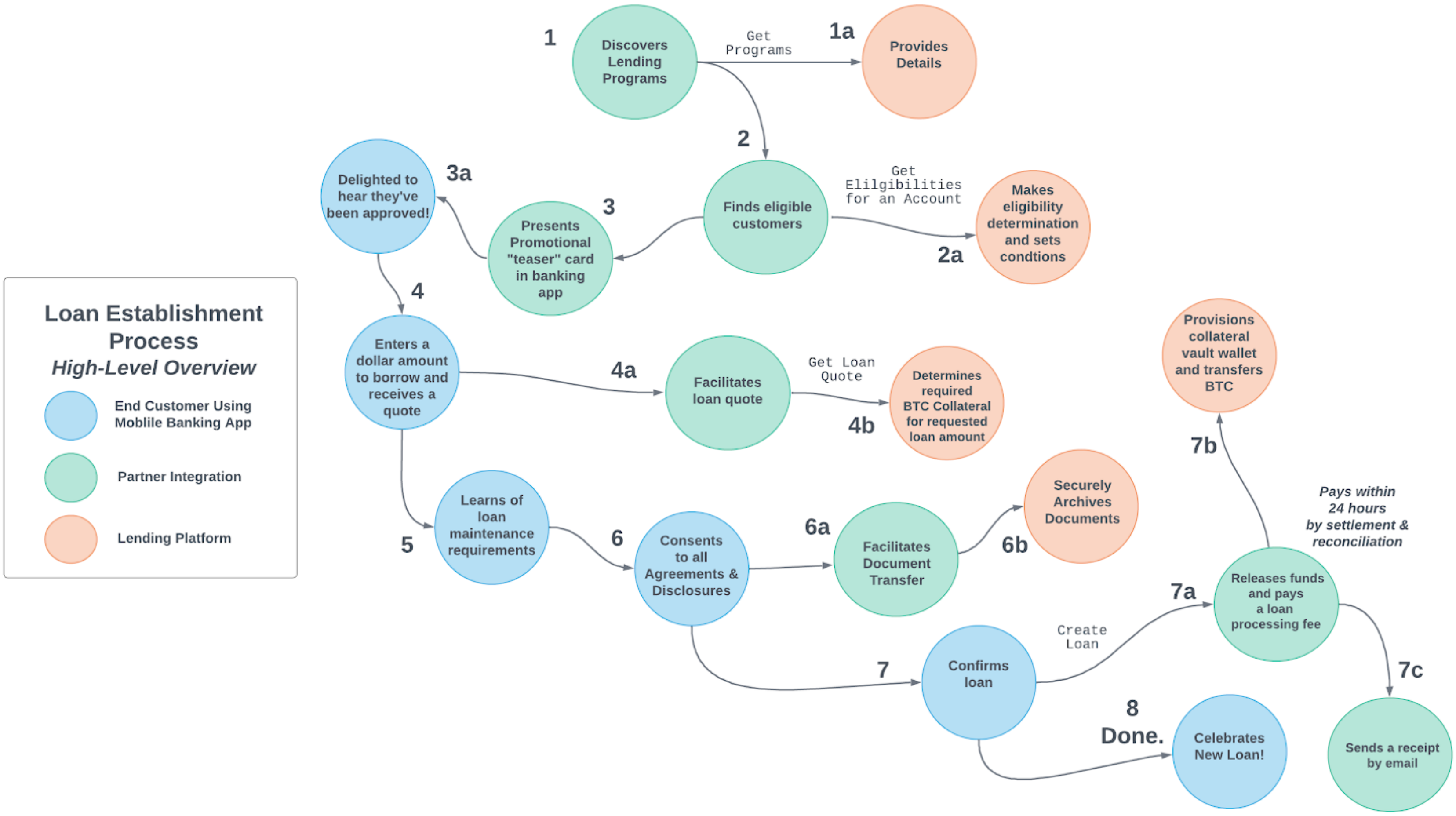
For optimal client experience and server utilization, the existing Java implementation of the firm's RESTful Web API service will use a modern framework that leverages asynchronous patterns (async/await) instead of blocking REST calls.

7) Implementation & Deployment Timelines, Security Aspects, Etc.
Out of scope. Instead, the focus of this design is the presentation of a comprehensive set of API methods & object models to solve the business problem in a viable manner, specified in-depth.

3.1 Standard Flows

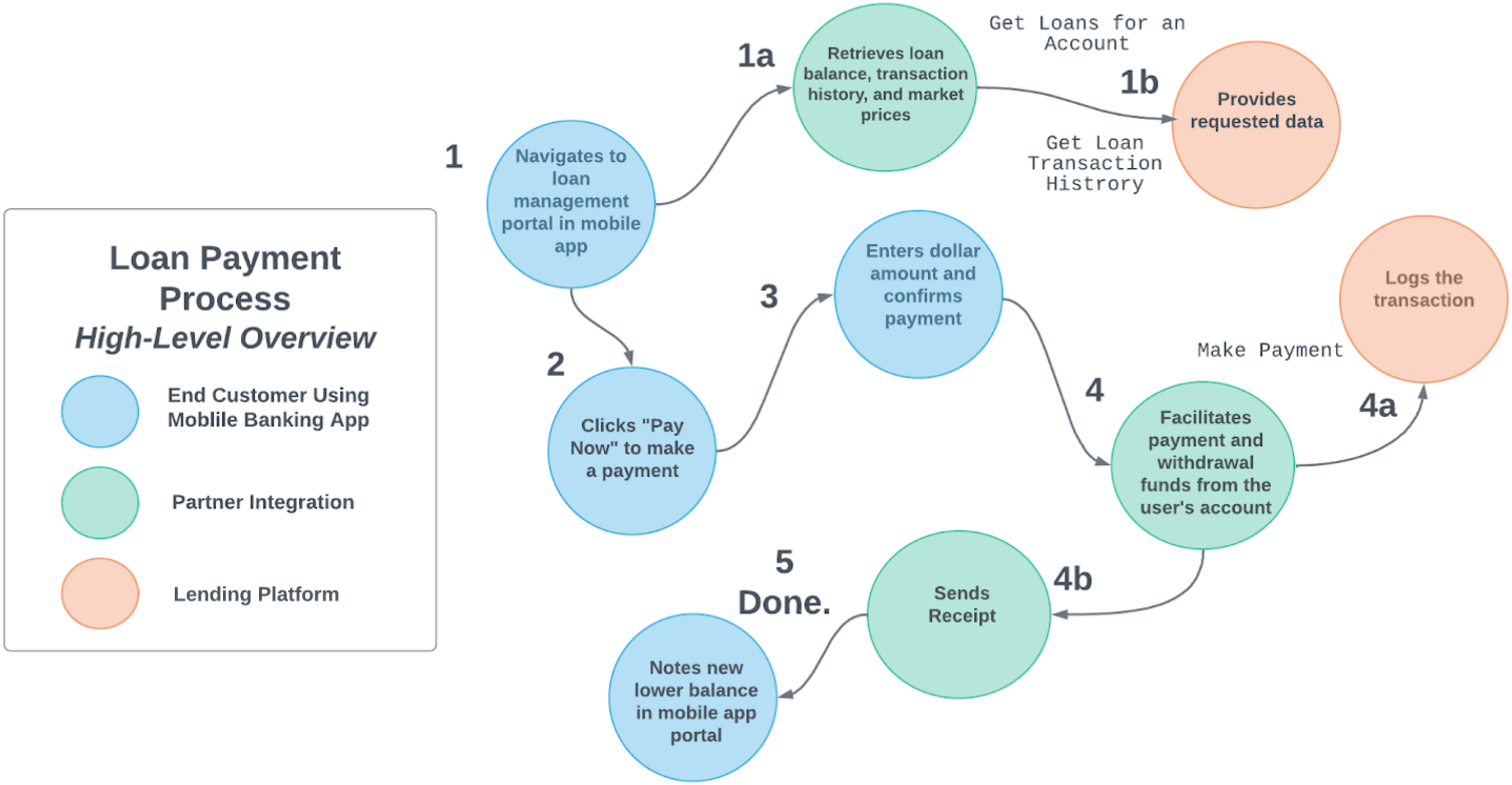
3.1.1 Loan Establishment Overview

The following depicts the high-level process, the interactions between all parties, and the API methods used.


Loan Establishment Process
High-Level Overview
End Customer Using Moblile Banking App
Partner Integration
Lending Platform
1
Discovers Lending Programs
Get Programs
1a
Provides Details
2
Finds eligible customers
Get Elilgibilities for an Account
2a
Makes eligibility determination and sets condtions
3
Presents Promotional "teaser" card in banking app
3a
Delighted to hear they've been approved!
4
Enters a dollar amount to borrow and receives a quote
4a
Facilitates loan quote
Get Loan Quote
4b
Determines required BTC Collateral for requested loan amount
5
Learns of loan maintenance requirements
6
Consents to all Agreements & Disclosures
6a
Facilitates Document Transfer
6b
Securely Archives Documents
7
Confirms loan
Create Loan
7a
Releases funds and pays a loan processing fee
7b
Provisions collateral vault wallet and transfers BTC
Pays within 24 hours by settlement & reconciliation
7c
Sends a receipt by email
8
Done.
Celebrates New Loan!

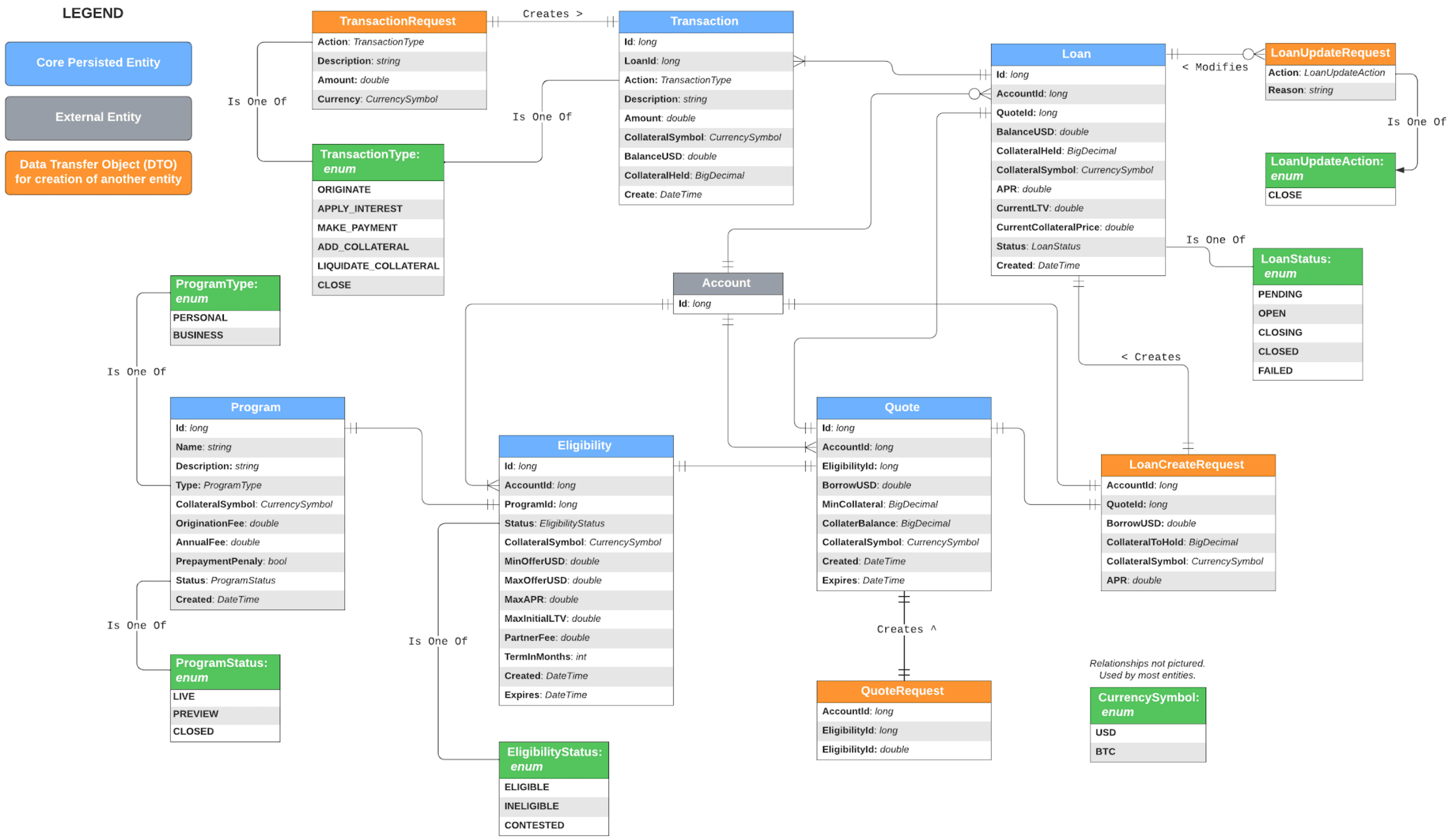
3.1.2 Loan Payment

The following depicts the high-level process, the interactions between all parties, and the API methods used. The “Add Collateral” workflow (not pictured) is essentially the same.


Loan Payment Process
High-Level Overview
End Customer Using Moblile Banking App
Partner Integration
Lending Platform
1
Navigates to loan management portal in mobile app
1a
Retrieves loan balance, transaction history, and market prices
Get Loans for an Account
1b
Provides requested data
Get Loan Transaction Histrory
2
Clicks "Pay Now" to make a payment
3
Enters dollar amount and confirms payment
4
Facilitates payment and withdrawal funds from the user's account
Make Payment
4a
Logs the transaction
4b
Sends Receipt
5
Done.
Notes new lower balance in mobile app portal

3.2 Business Object Models & Relationships

Following is a layout of all resources used in the requests & responses of this API and the relationships between them.



LEGEND

- Core Persisted Entity
- External Entity
- Data Transfer Object (DTO) for creation of another entity

TransactionRequest
Action: *TransactionType*
Description: *string*
Amount: *double*
Currency: *CurrencySymbol*

Transaction
Id: *long*
LoanId: *long*
Action: *TransactionType*
Description: *string*
Amount: *double*
CollateralSymbol: *CurrencySymbol*
BalanceUSD: *double*
CollateralHeld: *BigDecimal*
Create: *DateTime*

TransactionType: *enum*
ORIGINATE
APPLY_INTEREST
MAKE_PAYMENT
ADD_COLLATERAL
LIQUIDATE_COLLATERAL
CLOSE

Loan
Id: *long*
AccountId: *long*
QuoteId: *long*
BalanceUSD: *double*
CollateralHeld: *BigDecimal*
CollateralSymbol: *CurrencySymbol*
APR: *double*
CurrentLTV: *double*
CurrentCollateralPrice: *double*
Status: *LoanStatus*
Created: *DateTime*

LoanUpdateRequest
Action: *LoanUpdateAction*
Reason: *string*

LoanUpdateAction: *enum*
CLOSE

LoanStatus: *enum*
PENDING
OPEN
CLOSING
CLOSED
FAILED

Account
Id: *long*

ProgramType: *enum*
PERSONAL
BUSINESS

Program
Id: *long*
Name: *string*
Description: *string*
Type: *ProgramType*
CollateralSymbol: *CurrencySymbol*
OriginationFee: *double*
AnnualFee: *double*
PrepaymentPenaly: *bool*
Status: *ProgramStatus*
Created: *DateTime*

ProgramStatus: *enum*
LIVE
PREVIEW
CLOSED

Eligibility
Id: *long*
AccountId: *long*
ProgramId: *long*
Status: *EligibilityStatus*
CollateralSymbol: *CurrencySymbol*
MinOfferUSD: *double*
MaxOfferUSD: *double*
MaxAPR: *double*
MaxInitialLTV: *double*
PartnerFee: *double*
TermInMonths: *int*
Created: *DateTime*
Expires: *DateTime*

EligibilityStatus: *enum*
ELIGIBLE
INELIGIBLE
CONTESTED

Quote
Id: *long*
AccountId: *long*
EligibilityId: *long*
BorrowUSD: *double*
MinCollateral: *BigDecimal*
CollaterBalance: *BigDecimal*
CollateralSymbol: *CurrencySymbol*
Created: *DateTime*
Expires: *DateTime*

LoanCreateRequest
AccountId: *long*
QuoteId: *long*
BorrowUSD: *double*
CollateralToHold: *BigDecimal*
CollateralSymbol: *CurrencySymbol*
APR: *double*

QuoteRequest
AccountId: *long*
EligibilityId: *long*
EligibilityId: *double*

Relationships not pictured. Used by most entities.

CurrencySymbol: *enum*
USD
BTC

3.3 API Endpoints

3.3.1 Lending Programs

These endpoints return lists of the firm's lending programs. Our integration partners will use them to retrieve the basic loan structure of each program, including the target customer type, collateral currency & various fees. With this information, they will target customers with high-level discovery content within their products to entice customers to learn more.

Note: The 'PREVIEW' status incentivizes our most devoted partners to become early adopters. From the firm's perspective, all such programs are production ready.

3.3.1.1 Get All Programs

Returns *all* lending programs, including those that are well-established, newly released, and no longer available.

GET `{{baseUrl}}/programs`

Response Properties
This API method returns a list of program objects

- id: Unique identifier of the program
- name: Friendly descriptive identifier of the program
- description: Short informative summary to provide context
- type: Either ‘PERSONAL’ or ‘BUSINESS’. Additional types may be added
- collateralSymbol: The short name (symbol) of the cryptocurrency
- originationFee: The startup cost passed on to the partners’ end customer
- annualFee: The fee passed on to the partners’ end customer, charged

Example Request

GET https://api.example.com/lending/v1/programs

Example Response

```
[
  {
    "id": 1,
    "name": "BTC Secured Personal Loan",
    "description": "A Bitcoin-backed loan",
    "type": "PERSONAL",
    "collateralSymbol": "BTC",
    "originationFee": 0,
```

yearly

- **status**: The use readiness state, one of ‘LIVE’, ‘PREVIEW, or ‘CLOSED’
- **created**: The UTC timestamp of the creation of the loan

HTTP Response Codes

200 OK

```
        "annualFee": 0,
        "prepaymentPenalty": false,
        "status": "LIVE",
        "created": "2022-03-01T12:05:37"
    }
]
```

3.3.1.2 Get a Specific Program

Returns a single lending program, given its unique id.

GET `{{baseUrl}}/programs/<ID>`

URL Parameters

- **ID**: Unique identifier of the lending program to retrieve

Response Properties

- **id**: Unique identifier of the program
- **name**: Friendly descriptive identifier of the program
- **description**: Short informative summary to provide context
- **type**: Either ‘PERSONAL’ or ‘BUSINESS’. Additional types to be added
- **collateralSymbol**: The short name (symbol) of the cryptocurrency
- **originationFee**: The startup cost passed on to the partners’ end customer
- **annualFee**: The fee passed on to the partners’ end customer, charged

Example Request

```
GET https://api.example.com/lending/v1/programs/2
```

Example Response

```
{
  "id": 2,
  "name": "BTC Secured Business Loan",
  "description": "A loan to help your business grow!",
  "type": "BUSINESS",
  "collateralSymbol": "BTC",
  "originationFee": 0,
  "annualFee": 100,
  "prepaymentPenalty": false,
```

yearly

- **status**: The use readiness state, one of ‘LIVE’, ‘PREVIEW, or ‘CLOSED’
- **created**: The UTC timestamp of the creation of the loan

```
    "status": "PREVIEW",
    "created": "2022-03-01T12:07:45"
}
```

HTTP Response Codes

200 OK

404 NOT FOUND

3.3.2 Customer Eligibilities

Following is a comparison of three customer accounts and their eligibility determinations and conditions. For demonstration only - actual business rules will vary. For this example, the customer's state of residence determines overall eligibility, and the BTC balance (and other unlisted factors) determines whether each will qualify for a business loan (offered at more attractive interest rates, higher amounts, and longer terms).

Customer Account	1	2	3
State of Residence	South Carolina	Arizona	Nebraska
BTC Balance	0.037511243	1.43778521	0.20039265
Eligible?	Personal ✗ Business ✗	Personal ✓ Business ✓	Personal ✓ Business ✗
Min Offer USD	-	$100 / $5,000	$100 / -
Max Offer USD	-	$10,000 / $250,000	$7,500 / -

Max APR	-	12% / 8.5%	15% / -
Max Initial LTV	-	50% / 45%	40%
Partner Fee	-	4% / 3%	4%
Term in Months	-	12 / 36	12

⚠ **Implementation Note:** Work closely with the product development team to ensure the API controller will work seamlessly with the new *Lending Eligibility Rules Engine Microservice* (LEREM) currently in development.

3.3.2.1 Get Eligibilities for an Account

Retrieves a list of eligibility determinations for all lending programs for a given user.

GET `{{baseUrl}}/eligibilities?accountId=<ACCOUNT_ID>`

Query Parameters

- **accountId:** Unique id of the customer's account for which to get eligibility determination

Response Properties

This API method returns a list of eligibility objects

- **id:** Unique identifier of the eligibility determination
- **accountId:** Unique id of the customer's account associated with the eligibilities
- **programId:** Unique id of the lending program to which the eligibility applies
- **status:** The determination, one of 'ELIGIBLE', 'INELIGIBLE', or 'CONTESTED'
- **collateralSymbol:** The short name (symbol) of the cryptocurrency for use as collateral

Example Request

```
GET
https://api.example.com/lending/v1/eligibilities?accountId=1
```

Example Response

```
[
  {
    "id": 1,
    "accountId": 1,
    "programId": 1,
    "status": "ELIGIBLE",
    "collateralSymbol": "BTC",
```

- minOfferUSD: The smallest possible loan allowed, denominated in US dollars
- maxOfferUSD: The largest possible loan allowed, denominated in US dollars
- maxAPR: The maximum APR the partner is allowed to charge the customer
- maxInitialLTV: The maximum Loan-to-value at origination. Value of collateral must be greater than or equal to the borrowed amount divided by this number (expressed as a percentage)
- partnerFee: The US dollar amount expressed as a percentage of the borrowed amount, which must be paid to the firm within 24 hours of loan origination
- created: The UTC timestamp of the creation of the loan
- expires: The UTC timestamp at which the eligibility determination is no longer valid

```
        "minOfferUSD": 100,
        "maxOfferUSD": 10000,
        "maxAPR": 15,
        "maxInitialLTV": 40,
        "partnerFee": 2,
        "termInMonths": 12,
        "created": "2022-04-03T17:14:52",
        "expires": "2022-04-04T17:14:52"
    }
]
```

HTTP Response Codes

200 OK

404 NOT FOUND

3.3.3 Loan Quotes

Based on regulatory and legal guidance, the firm will determine whether each partner’s customer is eligible to participate in our lending programs (as well as the specific conditions).

3.3.3.1 Create a Loan Quote

Returns a quote for a loan, given the amount requested and the cryptocurrency's market price. This operation will use the properties of the referenced eligibility determination to verify that the requested loan amount is within limits and to calculate the required collateral amount.

For example:

The requested loan amount is `$800`, offered by the partner at `8.5%` interest:

Eligibility Property	Value	Within Limits?
Min Offer USD	100	✓
Max Offer USD	10,000	✓
Max APR	15%	✓
MaxInitialLTV	40%	-

Required collateral: `(Loan Amount / MaxInitialLTV) / BTC Market Price`

`(800 / 0.4) / 46195 =` **`0.04329473 BTC`**

⚠ **Implementation Note:** Because Bitcoin prices can be volatile, the quote is valid only for a brief time (60 seconds). Once expired, the quote can no longer be used to create a loan (via the Create a Loan endpoint).

POST `{{baseUrl}}/quotes`

Example Request

POST

https://api.example.com/lending/v1/quotes

```
{
  "accountId": 1,
  "eligibilityId": 1,
  "borrowUSD": 800
```

Request Properties

- **accountId:** Id of the customer's account for which to generate a quote
- **eligibilityId:** Id of the eligibility determination (must be status 'ELIGIBLE')
- **borrowUSD:** The requested loan amount, in US dollars

Response Properties

- **id**: Unique identifier of the loan quote
- **accountId**: Unique id of the customer's account associated with the loan quote
- **eligibilityId**: Unique id of the eligibility determination used to generate the quote
- **borrowUSD**: The requested loan amount, in US dollars
- **minCollateral**: The minimum amount of cryptocurrency that must be locked as collateral to secure the loan. *(Note: this value is derived from the eligibility determination's `MaxInitialLTV` property, as well as the price of the currency at the time of quote request)*.
- **collateralBalance**: The amount of cryptocurrency currently held in the account's wallet.
- **created**: The UTC timestamp of the creation of the loan quote
- **expires**: The UTC timestamp at which the loan quote is no longer valid

HTTP Response Codes

201 CREATED

400 BAD REQUEST

409 CONFLICT

"The account already has an active loan."

"The eligibility is not associated with the account."

422 UNPROCESSABLE ENTITY

"The accountId is invalid."

"The eligibilityId is invalid."

"The eligibility is not associated with the account."

"The account is not eligible for this lending program."

"The eligibility determination for this account is expired."

"The requested loan amount is less than the minimum allowed."

"The requested loan amount is greater than the maximum allowed."

"The requested loan amount requires collateral that exceeds the balance of this account."

```
}
```

Example Response

```
{
  "id": 1,
  "accountId": 1,
  "eligibilityId": 1,
  "borrowUSD": 800,
  "minCollateral": 0.04313599,
  "collateralBalance": 0.075,
  "collateralSymbol": "BTC",
  "created": "2022-04-03T13:57:45",
  "expires": "2022-04-03T13:58:45"
}
```

3.3.3.2 Get a Specific Loan Quote

Returns a previously generated loan quote, given its unique id.

GET `{{baseUrl}}/quotes/<ID>`

URL Parameters

- **ID:** Unique identifier of the lending program to retrieve

Response Properties

- **id:** Unique identifier of the loan quote
- **accountId:** Unique id of the customer's account associated with the loan quote
- **eligibilityId:** Unique id of the eligibility determination used to generate the quote
- **borrowUSD:** The requested loan amount, in US dollars
- **minCollateral:** The minimum amount of cryptocurrency that must be locked as collateral to secure the loan. *(Note: this value is derived from the eligibility determination's `MaxInitialLTV` property, as well as the price of the currency at time of quote request).*
- **collateralBalance:** The amount of cryptocurrency currently held in the account's wallet.
- **created:** The UTC timestamp of the creation of the loan quote
- **expires:** The UTC timestamp at which the loan quote is no longer valid

HTTP Response Codes

200 OK

404 NOT FOUND

Example Request

GET

https://api.example.com/lending/v1/quotes/1

Example Response

```
{
  "id": 1,
  "accountId": 1,
  "eligibilityId": 1,
  "borrowUSD": 800,
  "minCollateral": 0.04297471,
  "collateralBalance": 0.075,
  "collateralSymbol": "BTC",
  "created": "2022-04-03T17:18:38",
  "expires": "2022-04-03T17:19:380"
}
```

3.3.3.3 Get Loan Quotes for an Account

GET `{{baseUrl}}/quotes?accountId=<ACCOUNT_ID>`

Query Parameters

- **accountId:** Unique id of the customer's account for which to retrieve loan quote

Response Properties

This API method returns a list of loan quote objects

- **id:** Unique identifier of the loan quote
- **accountId:** Unique id of the customer's account associated with the loan quote
- **eligibilityId:** Unique id of the eligibility determination used to generate the quote
- **borrowUSD:** The requested loan amount, in US dollars
- **minCollateral:** The minimum amount of cryptocurrency than must be locked as collateral to secure the loan. *(Note: this value is derived from the eligibility determination's `MaxInitialLTV` property, as well as the currency's price at the time of quote request).*
- **collateralBalance:** The amount of cryptocurrency currently held in the account's wallet.
- **created:** The UTC timestamp of the creation of the loan quote
- **expires:** The UTC timestamp at which the loan quote is no longer valid

HTTP Response Codes

200 OK

404 NOT FOUND

Example Request

```
GET
https://api.example.com/lending/v1/quotes?
accountId=1
```

Example Response

```
[
  {
    "id": 1,
    "accountId": 1,
    "eligibilityId": 1,
    "borrowUSD": 800,
    "minCollateral": 0.04297471,
    "collateralBalance": 0.075,
    "collateralSymbol": "BTC",
    "created": "2022-04-03T17:18:38",
    "expires": "2022-04-03T17:19:38"
  }
]
```

3.3.4 Loans

3.3.4.1 Create a Loan

Creates a loan from the referenced quote, for a given account

⚠ **Implementation Notes:**

- Before proceeding with loan creation, this method must ensure that all required documents have been agreed upon by the end customer and uploaded to the archival endpoint. In the absence of the above, this method must return HTTP 403 Forbidden.

- Make sure the reference quote is unexpired. If it is expired, you need to generate a new one and call this method again before creating a loan.

Upon successful loan creation:

- The partner must transfer funds into the settlement account to honor the processing fee to the firm in the amount specified within the eligibility determination of the customer account (partnerFee property).

- The partner, as lienholder, is legally authorized to liquidate collateral. Given qualifying conditions (e.g., non-payment, excessive LTV), the firm must transfer the collateral from the customer's wallet into a new secure vault wallet (both within the firm's custody). The customer will maintain ownership of the funds in the vault wallet but will not be able to access them until they have met the loan repayment terms.

POST `{{baseUrl}}/loans`

Request Properties

- **accountId:** Id of the customer's account for which to create a loan
- **quoteId:** Id of the loan quote *(Note: quote must be unexpired)*
- **borrowUSD:** The requested loan amount, in US dollars
- **collateralToHold:** The amount of cryptocurrency to be held as collateral *(Note: this*

Example Request

POST

https://api.example.com/lending/v1/loans

```
{
  "accountId": 1,
  "quoteId": 1,
```

amount must be equal to or greater than the amount indicated in the referenced loan quote.)

- collateral: The symbol of the cryptocurrency to be held as collateral
- APR: The annual percentage rate the partner will charge the customer (Note: must be equal to or less than the maxAPR value specified in the original eligibility determination)

Response Properties

- id: Unique identifier of the loan
- accountId: Unique id of the customer's account associated with the loan
- quoteId: Unique id of the quote which determined the collateral for the loan
- balanceUSD: The outstanding (unpaid) balance of the loan
- collateralHeld: The amount of cryptocurrency being held in the collateral vault wallet
- collateralSymbol: The short name (symbol) of the cryptocurrency
- apr: The annual percentage rate charged by the partner to the customer
- currentLTV: The loan-to-value at of the loan at current time
- currentCollateralPrice: The market price of the collateral currency at the current time
- status: The state of the loan, one of 'PENDING', 'OPEN', 'CLOSING', 'CLOSED', or 'FAILED'
- created: The UTC timestamp at which the loan was created

HTTP Response Codes

201 CREATED

400 BAD REQUEST

403 FORBIDDEN

"One or more agreement documents are missing"

409 CONFLICT

"The quote is not associated with the account."

422 UNPROCESSABLE ENTITY

```
  "borrowUSD": 800,
  "collateralToHold": 0.04297471,
  "collateral": "BTC",
  "APR": 7.5
}
```

Example Response

```
{
  "id": 1,
  "accountId": 1,
  "quoteId": 1,
  "balanceUSD": 800,
  "collateralHeld": 0.04297471,
  "collateralSymbol": "BTC",
  "apr": 7.5,
  "currentLTV": 40,
  "currentCollateralPrice": 46538.99,
  "status": "OPEN",
  "created": "2022-04-03T17:20:18"
}
```

"The accountId is invalid."

"The quoteId is invalid."

"The quote is expired."

"The loan amount must match the amount specified in the loan quote"

"The collateral amount must be equal to or greater than amount specified in the loan quote"

"The eligibility determination associated with the quote is invalid."

"The requested collateral amount exceeds the balance of this account."

"The annual percentage rate (APR) exceeds the eligibility limits of the lending program"

3.3.4.2 Get a Specific Loan

Returns a loan, given its unique id.

GET `{{baseUrl}}/loans/<ID>`

URL Parameters

- **ID:** Unique identifier of the loan to retrieve

Response Properties

- **id:** Unique identifier of the loan
- **accountId:** Unique id of the customer's account associated with the loan
- **quoteId:** Unique id of the quote which determined the collateral for the loan
- **balanceUSD:** The outstanding (unpaid) balance of the loan
- **collateralHeld:** The amount of cryptocurrency being held in the collateral vault wallet
- **collateralSymbol:** The short name (symbol) of the cryptocurrency
- **apr:** The annual percentage rate charged by the partner to the customer
- **currentLTV:** The loan-to-value at the current time
- **currentCollateralPrice:** The market price of the collateral currency at the current time
- **status:** The state of the loan, one of 'PENDING', 'OPEN', 'CLOSING', 'CLOSED',

Example Request

```
GET
https://api.example.com/lending/v1/loans/1
```

Example Response

```
{
  "id": 1,
  "accountId": 1,
  "quoteId": 1,
  "balanceUSD": 800,
  "collateralHeld": 0.04297471,
  "collateralSymbol": "BTC",
  "apr": 7.5,
  "currentLTV": 40.3,
  "currentCollateralPrice": 46201,
  "status": "OPEN",
```

or 'FAILED'

- **created:** The UTC timestamp at which the loan was created

```
  "created": "2022-04-03T17:20:18"
}
```

HTTP Response Codes

200 OK

404 NOT FOUND

3.3.4.3 Get Loans for an Account

Returns all loans, past and present, for a given account.

⚠ **Implementation Note:** Only one loan may be open at any given time for each account. This limitation will be lifted in the next release version once the team has addressed issues with the backend service.

GET `{{baseUrl}}/loans?accountId=<ACCOUNT_ID>`

Query Parameters

- **accountId:** Unique id of the customer's account for which to retrieve the loan

Response Properties

This API method returns a list of loan objects

- **id:** Unique identifier of the loan
- **accountId:** Unique id of the customer's account associated with the loan
- **quoteId:** Unique id of the quote which determined the collateral for the loan

Example Request

```
GET
https://api.example.com/lending/v1/loans?accountId=1
```

Example Response

```
[
  {
    "id": 1,
    "accountId": 1,
    "quoteId": 2,
```

- **balanceUSD:** The outstanding (unpaid) balance of the loan
- **collateralHeld:** The amount of cryptocurrency being held in the collateral vault wallet
- **collateralSymbol:** The short name (symbol) of the cryptocurrency
- **apr:** The annual percentage rate charged by the partner to the customer
- **currentLTV:** The loan-to-value at current time
- **currentCollateralPrice:** The market price of the collateral currency at current time
- **status:** The state of the loan, one of ‘PENDING’, ‘OPEN’, ‘CLOSING’, ‘CLOSED’, or ‘FAILED’
- **created:** The UTC timestamp at which the loan was created

HTTP Response Codes

200 OK

404 NOT FOUND

```
        "balanceUSD": 800,
        "collateralHeld": 0.04297471,
        "collateralSymbol": "BTC",
        "apr": 7.5,
        "currentLTV": 40.3,
        "currentCollateralPrice": 46167,
        "status": "OPEN",
        "created": "2022-04-03T17:20:18"
    }
]
```

3.3.4.4 Get Loan Transaction History

Returns a list of all transactions associated with the loan, within a given timeframe, in descending order by date.

⚠️**Implementation Note:** For auditing purposes & legal compliance, the partner must log all transactions associated with the loan, including origination and closure. In the case of a payment, for example, even though the firm is entirely insulated from the movement of USD funds, the partner must nevertheless log the transaction (using the *Make Payment* endpoint) so that the loan balance can be adjusted accordingly.

GET `{{baseUrl}}/loans/<ID>/transactions?from=<START>&to=<END>`

URL Parameters

- **ID:** Unique identifier of the loan for which to retrieve the transaction history

Example Request

GET

https://api.example.com/lending/v1/loans/1/transactions?from=1646168541&to=1649

Query Parameters

- **from**: The start timeframe (in Unix seconds)
- **to**: **(optional)** The end timeframe (in Unix seconds). Defaults to now.

Response Properties

This API method returns a list of loan transaction objects, sorted by date in descending order

- **id:** Unique identifier of the loan transaction
- **loanId:** Unique identifier of the loan to which the transaction pertains
- **action:** The type of action conducted by the transaction, one of: ‘ORIGINATE’, ‘APPLY_INTEREST’, ‘MAKE_PAYMENT’, ‘ADD_COLLATERAL’, ‘LIQUIDATE_COLLATERAL’, or ‘CLOSE’
- **description:** A short informative message giving reason for the transaction
- **amount:** The amount transacted (Note: this field is unused with ‘ORIGINATE’ & ‘CLOSE’)
- **currencySymbol:** The short name of the currency to which amount applies (e.g., ‘BTC’ or ‘USD’)
- **balanceUSD:** The outstanding balance of the loan after the transaction, in US dollars
- **collateralHeld:** The amount of cryptocurrency being held in the collateral vault wallet
- **created:** The UTC timestamp at which the transaction was created

HTTP Response Codes

200 OK

400 BAD REQUEST

404 NOT FOUND

030382

Example Response

```
[
  {
    "id": 3,
    "loanId": 1,
    "action": “CLOSE”,
    "description": “Closed account”,
    "amount": 0,
    "currencySymbol": "BTC",
    "balanceUSD": 0,
    "collateralHeld": 0,
    "created": "2022-04-03T17:44:41"
  },
  {
    "id": 2,
    "loanId": 1,
    "action": “MAKE_PAYMENT”,
    "description": "Paid in full",
    "amount": 259.99,
    "currencySymbol": "USD",
    "balanceUSD": 0,
    "collateralHeld": 0.04032722,
    "created": "2022-04-03T17:44:23"
  },
 {
    "id": 1,
    "loanId": 1,
    "action": “ORIGINATE”,
    "description": “Open new loan”,
    "amount": 800,
    "currencySymbol": "BTC",
    "balanceUSD": 800,
    "collateralHeld": 0.045,
    "created": "2022-04-03T17:40:15"
  }
]
```

3.3.4.5 Close Loan

Updates the status of the loan to ‘CLOSED’.

The outstanding loan balance must be zero, or this operation must return 403 Forbidden.

PATCH `{{baseUrl}}/loans/<ID>`

URL Parameters

- **ID:** Unique identifier of the loan to close

Request Properties

- **action:** The operation to be conducted. Must be ‘CLOSE’

Example Request

```
PATCH
https://api.example.com/lending/v1/loans/1
{
  "action": "CLOSE"
}
```

Response Properties

- **id**: Unique identifier of the loan
- **accountId**: Unique id of the customer's account associated with the loan
- **quoteId**: Unique id of the quote which determined the collateral for the loan
- **balanceUSD**: The outstanding (unpaid) balance of the loan
- **collateralHeld**: The amount of cryptocurrency being held in the collateral vault wallet
- **collateralSymbol**: The short name (symbol) of the cryptocurrency
- **apr**: The annual percentage rate charged by the partner to the customer
- **currentLTV**: The loan-to-value at current time
- **currentCollateralPrice**: The market price of the collateral currency at current time
- **status**: The state of the loan, one of 'PENDING', 'OPEN', 'CLOSING', 'CLOSED', or 'FAILED'
- **created**: The UTC timestamp at which the loan was created

HTTP Response Codes

204 NO CONTENT

400 BAD REQUEST

403 FORBIDDEN

"The loan is not in a state to permit closure"

404 NOT FOUND

Example Response

No Content is returned.

3.3.5 Loan Transactions

3.3.5.1 Apply Interest

This endpoint is for adding the interest amount to the outstanding balance of the loan, monthly.

POST `{{baseUrl}}/loans/<ID>/transactions`

URL Parameters

- **ID:** Unique identifier of the loan for which to add interest amount to the outstanding balance

Request Properties

- **action:** The transaction operation to be conducted. Must be 'APPLY_INTEREST''
- **description:** A brief summary message to provide the reason for the transaction
- **amount:** The amount to be added to the outstanding balance of the loan
- **currency:** The currency in which the amount is denominated. Must be 'USD'

Response Properties

- **id:** Unique identifier of the newly created *apply interest* transaction
- **loanId:** Unique identifier of the loan to which the transaction applies
- **action:** The action conducted: 'APPLY_INTEREST'
- **description:** A short informative message providing a reason for the transaction
- **amount:** The amount added to the outstanding balance
- **currencySymbol:** The currency in which the amount is denominated. Must be 'USD'.
- **balanceUSD:** The outstanding balance of the loan after the transaction, in US dollars
- **collateralHeld:** The amount of cryptocurrency being held in the collateral vault wallet
- **created:** The UTC timestamp at which the transaction was created

HTTP Response Codes

201 CREATED

400 BAD REQUEST

409 CONFLICT

"The loan is not in an active state"

422 UNPROCESSABLE ENTITY

"Expected 'USD' currency for applying interest."

Example Request

POST

https://api.example.com/lending/v1/loans/1/transactions

```
{
  "action": "APPLY_INTEREST",
  "description": "Adding interest for May",
  "amount": 5.33,
  "currency": "USD"
}
```

Example Response

```
{
  "id": 2,
  "loanId": 1,
  "action":  "APPLY_INTEREST",
  "description": "Adding interest for May",
  "amount": 5.33,
  "currencySymbol": "USD",
  "balanceUSD": 805.33,
  "collateralHeld": 0.04297471,
  "created": "2022-04-03T17:32:17"
}
```

3.3.5.2 Make Payment

This endpoint is for making payments to the customer’s outstanding balance.

POST `{{baseUrl}}/loans/<ID>/transactions`

URL Parameters

- **ID:** Unique identifier of the loan for which to make a payment

Request Properties

- **action:** The transaction operation to be conducted. Must be ‘MAKE_PAYMENT’’
- **description:** A brief summary message to provide the reason for the transaction
- **amount:** The amount to pay down the loan
- **currency:** The currency in which the amount is denominated. Must be ‘USD’

Response Properties

- **id:** Unique identifier of the newly created *make payment* transaction
- **loanId:** Unique identifier of the loan to which the transaction applies
- **action:** The action conducted:: ‘MAKE PAYMENT’
- **description:** A short informative message providing a reason for the transaction
- **amount:** The amount paid on the outstanding balance
- **currencySymbol:** The currency in which the amount is denominated. Must be ‘USD’.
- **balanceUSD:** The outstanding balance of the loan after the transaction, in US dollars
- **collateralHeld:** The amount of cryptocurrency being held in the collateral vault wallet
- **created:** The UTC timestamp at which the transaction was created

HTTP Response Codes

201 CREATED

400 BAD REQUEST

409 CONFLICT

“The loan is not in an active state"

422 UNPROCESSABLE ENTITY

“Expected 'USD' currency for a payment transaction."

“Payment amount exceeds loan balance."

Example Request

```
POST
https://api.example.com/lending/v1/loans/1/
transactions

{
  "action": "MAKE_PAYMENT",
  "description": "Payment for May",
  "amount": 100,
  "currency": "USD"
}
```

Example Response

```
{
  "id": 3,
  "loanId": 1,
  "action": "MAKE_PAYMENT",
  "description": "Payment for May",
  "amount": 100,
  "currencySymbol": "USD",
  "balanceUSD": 705.33,
  "collateralHeld": 0.045,
  "created": "2022-04-03T17:34"
}
```

3.3.5.3 Add Collateral

This endpoint is for adding collateral to secure the customer's balance further.

The customer must have sufficient funds in their wallet, as this operation results in a cryptocurrency transfer from the customer's wallet into the secure vault wallet.

⚠**Implementation Note:** TODO: measure the performance of this long-running operation. We may need to move to a fully asynchronous model and securely enqueue the transfer operation. **Be sure to lock funds** to prevent race conditions in which a concurrent liquidation request may intervene.

POST `{{baseUrl}}/loans/<ID>/transactions`

URL Parameters

- **ID:** Unique identifier of the loan for which to add collateral

Request Properties

- **action:** The transaction operation to be conducted. Must be 'ADD_COLLATERAL''
- **description:** A brief summary message to provide the reason for the transaction
- **amount:** The amount to pay down the loan
- **currency:** The currency in which the amount is denominated. Must be 'BTC'. Additional currency types may be added later.

Response Properties

- **id:** Unique identifier of the newly created *add collateral* transaction
- **loanId:** Unique identifier of the loan to which the transaction applies
- **action:** The action conducted: 'ADD_COLLATERAL''
- **description:** A short informative message providing a reason for the transaction
- **amount:** The amount added to collateral
- **currencySymbol:** The currency in which the amount is denominated. Must be 'USD'.
- **balanceUSD:** The outstanding balance of the loan after the transaction, in US dollars
- **collateralHeld:** The amount of cryptocurrency being held in the collateral vault wallet
- **created:** The UTC timestamp at which the transaction was created

HTTP Response Codes

201 CREATED

400 BAD REQUEST

409 CONFLICT

"The loan is not in an active state"

422 UNPROCESSABLE ENTITY

"Expected 'BTC' currency for an add collateral transaction."

Example Request

POST

https://api.example.com/lending/v1/loans/1/transactions

```
{
  "action": "ADD_COLLATERAL",
  "description": "Topping off collateral",
  "amount": 0.005,
  "currency": "BTC"
}
```

Example Response

```
{
  "id": 4,
  "loanId": 1,
  "action":  "ADD_COLLATERAL",
  "description": "Topping off collateral",
  "amount": 0.005,
  "currencySymbol": "BTC",
  "balanceUSD": 605.33,
  "collateralHeld": 0.05,
  "created": "2022-04-03T17:36:05"
}
```

3.3.5.4 Liquidate Collateral

This endpoint is for liquidating the customer’s collateral.

The partner is legally authorized to liquidate the customer’s collateral under certain conditions (such as non-payment or excessive LTV) as a lienholder. Specific to the reason, the partner must make multiple attempts to notify the customer and resolve the issue. For example, in the event of non-payment, we recommend that the partner first inform the customer via email and SMS at least three times over 72 hours before proceeding with liquidation. For this example, the liquidation amount should be no more than is required for the partner to recover the lost payment, plus fees to the firm.

⚠ **Implementation Note:** Before proceeding with liquidation, ensure the partner has uploaded the required legal documents to our archive endpoint. If not, fail with 403 Forbidden.

```
POST {{baseUrl}}/loans/<ID>/transactions
```

URL Parameters

- **ID:** Unique identifier of the loan for which to liquidate collateral

Request Properties

- **action:** The transaction operation to be conducted. Must be ‘LIQUIDATE_COLLATERAL’’
- **description:** A brief summary message to provide the reason for the transaction
- **amount:** The amount to liquidate
- **currency:** The currency in which the amount is denominated. Can be either ‘USD’ or ‘BTC’ Additional currency types may be added later.

Response Properties

- **id:** Unique identifier of the newly created *liquidate collateral* transaction

Example Request

POST

https://api.example.com/lending/v1/loans/1/transactions

Example Request (USD)

```
{
  "action": "LIQUIDATE_COLLATERAL",
  "description": "Missed Payment",
  "amount": 100,
  "currency": "USD"
}
```

Example Response - USD

- **loanId:** Unique identifier of the loan to which the transaction applies
- **action:** The action conducted: ‘LIQUIDATE_COLLATERAL’’
- **description:** A short informative message providing reason for the transaction
- **amount:** The amount to liquidate
- **currencySymbol:** The currency in which the amount is denominated. Can be either ‘USD’ or ‘BTC’
- **balanceUSD:** The outstanding balance of the loan after the transaction, in US dollars
- **collateralHeld:** The amount of cryptocurrency being held in the collateral vault wallet.
- **created:** The UTC timestamp at which the transaction was created

HTTP Response Codes

201 CREATED

400 BAD REQUEST

403 FORBIDDEN

409 CONFLICT

“The loan is not in an active state"

422 UNPROCESSABLE ENTITY

“Expected 'BTC' currency for an add collateral transaction."

```
{
  "id": 5,
  "loanId": 1,
  "action": 4,
  "description": "Missed Payment",
  "amount": 100,
  "currencySymbol": "USD",
  "balanceUSD": 605.33,
  "collateralHeld": 0.04782722,
  "created": "2022-04-03T17:40:34"
}
```

Example Request - BTC

```
{
  "Action": "LIQUIDATE_COLLATERAL",
  "Description": "BTC Price Drop",
  "Amount": 0.0075,
  "Currency": "BTC"
}
```

Example Response - BTC

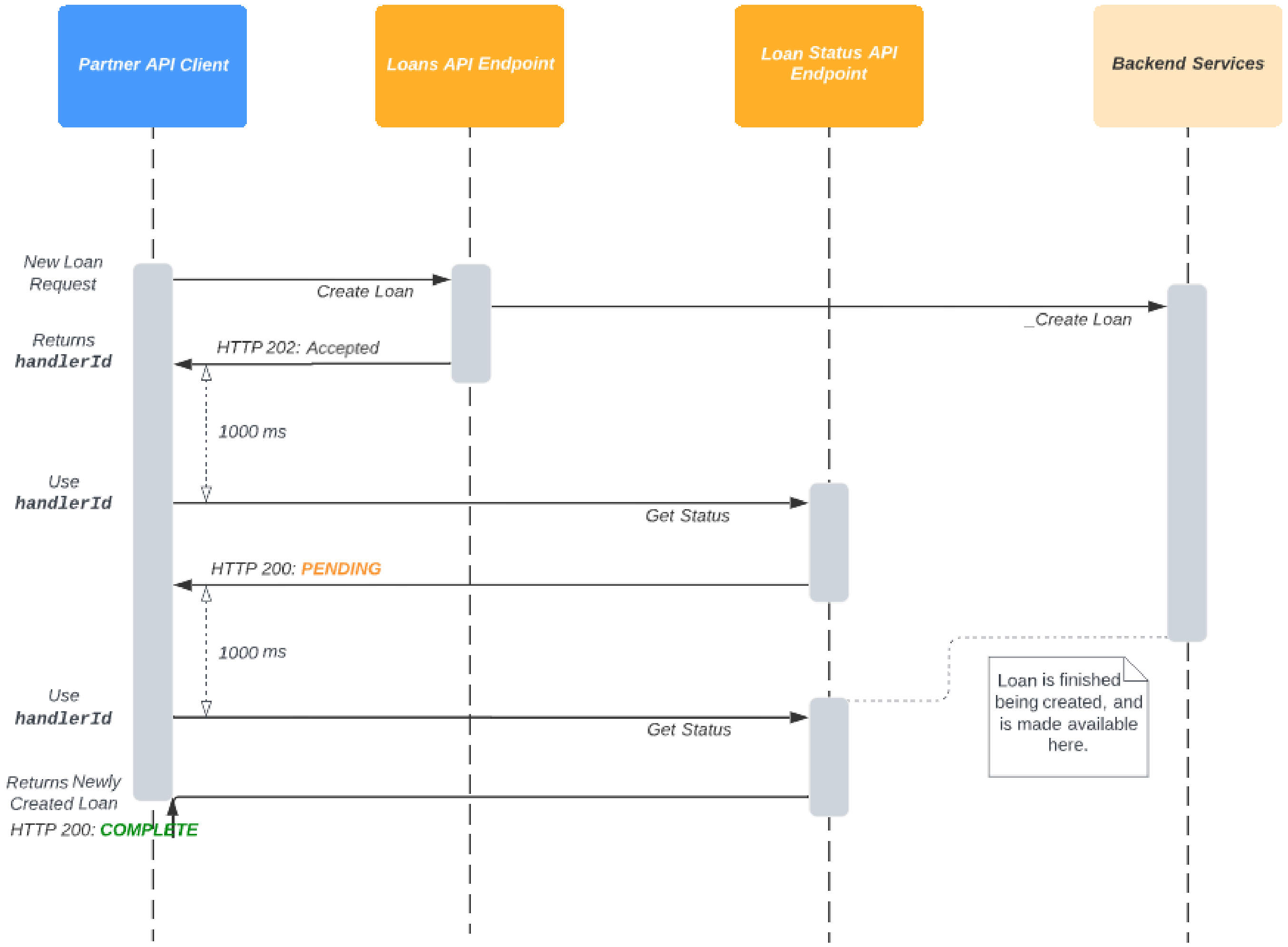
```
{
  "id": 6,
  "loanId": 1,
  "action": 4,
  "description": "BTC Price Drop",
  "amount": 0.0075,
  "currencySymbol": "BTC",
  "balanceUSD": 259.99,
  "collateralHeld": 0.04032722,
  "created": "2022-04-03T17:42:28"
}
```

4 Milestones & Delivery Schedule

To be delivered within sprint 22.

5 Appendix

For the MVP, polling will be used for long-running operations. This will be deprecated in favor of webhooks, once those become available.


Partner API Client
Loans API Endpoint
Loan Status API Endpoint
Backend Services
New Loan Request
Create Loan
_Create Loan
Returns handlerId
HTTP 202: Accepted
1000 ms
Use handlerId
Get Status
HTTP 200: PENDING
1000 ms
Use handlerId
Get Status
Loan is finished being created, and is made available here.
Returns Newly Created Loan
HTTP 200: COMPLETE