LOANPRO API INTEGRATION GUIDE

The purpose of this document is to outline the steps involved in integrating with the LoanPro API. While these steps could be generic to any integration, LoanPro encompasses more functionality than most software, so extra instruction is warranted. We have included recommendations and some best-practices information along with warnings about potential pitfalls. The goal is to help you develop an integration that will be secure, simple, and maintainable.

Table of Contents

Research (Lenders/Programmers)

The research phase of API integration involves gathering the information you need to start the project. Ideally, when you start the planning phase, you should know how to find documentation that explains how to use the LoanPro API, the terminology LoanPro uses, and the features that are available through the API. It is unwise to assume that something can be done through the LoanPro API if it's not mentioned in our documentation. We offer the following resources to help in your integration process:

- Restler Documentation (Programmers) - Shows the available endpoints, payloads and request methods that are available in our API.
- Articles (Lenders/Programmers) - Gives information about the features of LoanPro including what they are and how to use them.
- Plunker (Programmers) - Gives sample code for some of the most used API requests.
- Plunker Presentations (Programmers/Lenders) - Gives some explanation of the sample code.
- SDK (Programmers) - Code that can be used as a part of your integration to make some basic functions easier.
- Elasticsearch Documentation (Programmers) - Documentation about how to create Elasticsearch requests.
- Database Schema (Programmers) - A basic map of our database. This is only useful if you have a read-only copy of the database, which is only available to enterprise customers.
- Programmer-Lender Reference (Lenders/Programmers) - A reference guide to help lenders and programmers make sure they are talking about the same thing. This guide will give basic explanation of each endpoint and each payload component.

Each item in the above list shows whether it is intended for use by lenders, programmers, or both. As a part of the research phase, you should familiarize yourself with the documents that are intended for personnel in your role.

Why is Research Important?

Usually the biggest hurdle to overcome when planning an integration is getting on the same page when it comes to terminology. LoanPro uses specific words and names when talking about loans and loan-related processes. You may use different words. Your developers may or may not understand how lending works, but they understand servers, payloads, responses, code, and architecture.

For example, when you want to write an uncollectible balance off your books, LoanPro refers to this process as a charge-off. You may refer to it as a write-off, closing a delinquent loan, or in some other way. Your programmer may not know the specifics of how this process works.

If you tell your programmer to “make sure we can see write-offs” in the system you are creating, the programmer will likely have a question about what that is. When they look for that in LoanPro, they will need to know that what you call a write-off, LoanPro refers to as a charge-off.

Familiarizing yourself with our resources and using them in your development process can help you and your developers get and stay on the same page. It should also help you know what LoanPro does and doesn’t do, so you don’t waste time and money moving your development in the wrong direction.

Planning (Lenders)

The goal of the planning phase is to decide what you would like your LoanPro integration to accomplish, how it will work, and generally how it will look. This will typically involve personnel with knowledge of lending, your workflow, company policies, and lending laws. During this phase, you should create a written plan for your developers so that they will have something to refer to that defines the “what” of the integration. The “how” should be taken care of by their knowledge of development and our documentation.

Why is Planning Important?

One of the biggest problems companies have when integrating an application with LoanPro is a lack of experience in project management. Often companies overlook the fact that there is substantial work to do in planning what the integration will do, or they mistakenly think their programmers can work with LoanPro’s staff on the project, rendering planning unnecessary. This is not an option since LoanPro doesn’t offer project management services. Your programmers will need specific direction in order to create what you want.

The most important reason for planning your project is that it will help you figure out and define what you want in a concrete way. For example, if you decide that you want your integrated application to show average return on investment for your loans, your programmers will need to know what that actually means.

Let’s say you decide that average return on investment is the total amount paid in interest, fees, and escrow, divided by the number of loans you have.

That seems simple enough, but in reality, depending on how you use the system, you will likely want the total number of loans to exclude archived loans or loans in certain statuses. You may find that your application is faster if instead of pulling the individual numbers for interest, fees, and escrow that you just use the total paid minus the amount paid in principal. You should record the decisions on what you want so they can be communicated to your developers. A major test of your planning will be how well you can communicate the plan.

Scope

Your planning should cover each feature you plan to include in your integration. This includes anything that will be visible within the integrated application, any calculations the integrated application will perform, and any other function the integrated application will do. Your planning should also include basic navigation within the application, pages or functions outside the application, and an idea for how the different pages and features will be grouped.

Don't make assumptions in documenting your plan. You may think, "Anyone would know what a financial statement should include" and simply record that the application needs to be able to create a financial statement. The results of being too general, in this case, could be that your developers spend time creating something that is not what you wanted. Ultimately this will cost you company a lot more time and money that proper planning would have. The more thorough your planning and documentation, the better the integrated application.

Decisions about programming languages, hosting technologies, database structure, color scheme, etc. are outside the scope of the planning phase.

Direction

When planning your integration, the goal should be to define what you want in such a way that you will be able to answer any question about the integration. There will be some things that will change slightly based on input from your developers because it will be more cost/time effective to develop in a certain way, but you should define the integration well enough that you will know what your preferences are so you can make these decisions intelligently. You should know what features are important enough that it will be worth it to pay more to develop them.

Features List

Make a list of features you want to include in your integration. This list should be as comprehensive as possible, but don't worry about the specifics of how each feature will work yet.

Categorize Features

In order to organize your software application, categorize your features into main categories and subcategories. The purpose of this is to help properly set up the navigation flow of your application. There are three main areas of the LoanPro API: Loans, Customers, and Reports. Use these as your main categories and include as many subcategories as you think are necessary.

Example:

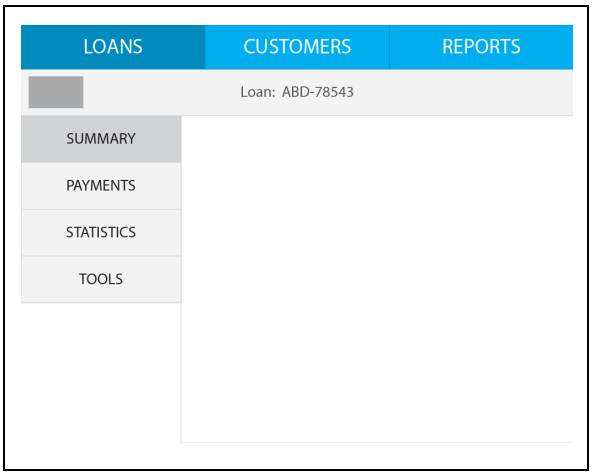
In LoanPro we made very definite decisions that customers should not be dependent on loans and loans should not be dependent on customers. A question arose about payment profile information. Should credit card information be stored with a loan or with a customer? It made the most sense to associate payment profile data with customers so the same payment profiles would be present for each loan a customer is linked to.

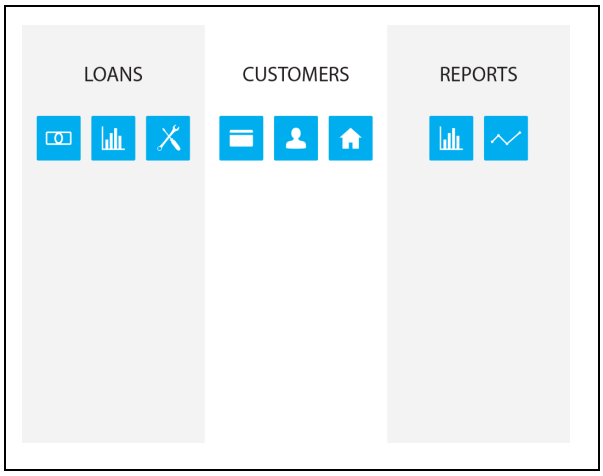
But, while payment profiles are associated with customers (actually stored in PCI Wallet), payments are logged directly on loans. While it may not be intuitive to separate a credit card payment from the credit card itself, it was an important decision in the development of LoanPro. Similar decisions will have to be made in planning your application.

Basic Navigation

Now that you have a list of features and you have categorized them, it's time to plan your basic navigation. This doesn't have to be elaborate, but you should decide whether the main navigation will be shown across the top or down the side. You should also have an idea of how submenus will be organized. It isn't necessary to draw a picture, but since your workflow will be aided by the site navigation, you should put some thought into the navigation.

Here are two examples for how navigation could be set up.





Features Definition

Now you can complete the work of outlining how each feature will work. The Feature Outline document (APPENDIX A) should be completed for each feature. It is likely that you will work directly with programmers throughout the process, so the Feature Outline document is designed to help you get a more concrete idea of each feature.

When you are done with your planning, you should be ready to get your programmers started on wireframing the application.

Wireframing (Programmers)

The wireframing step will help to keep developers and lenders on the same page. If the planning phase has been completed, the lender should have a good idea what the integrated application will do and how it will be structured. It is recommended that a rough wireframe for each page/feature/workflow should be completed and submitted for approval before significant work is done on the feature.

Building the Integration (Programmers)

Typical Integrations

Customer-Facing Website

Our users often want to create an integration that provides a way for their customers to log in, make payments, view account data, etc. This type of integration is usually concerned with:

- Authentication - LoanPro offers a customer-facing website as one of our features. We offer a simple authentication endpoint ***tenants({id})/customers/authenticate*** that will return a 200 response if successful, or a 401 if not successful. This means you can use LoanPro to manage passwords and you won't have to build out authentication in your integrated application. The payload is simply: ***{"username":"<username>","password":"<password>"}***.
- Customer Information - Customer information can be pulled and updated through the ***/odata.svc/Customers()*** endpoint. It is important to note that payment profile (credit/debit card, bank account) information is also related to a customer, but should be added, updated, or deleted using the PCI Wallet API. Here are some resources that give more information about working with customers: Customer Overview Article, Create a Customer API Article, Create a Customer Plunker Sample, Create a Customer Plunker Presentation.
- Payments - *See the guidelines below in Dynamic and Transactional Data for payments.* Although payments are typically pulled and logged through the loans endpoint, it is a significant part of the integration. There are many things associated with payments including logging payments, processing credit/debit card or bank transactions, automatic payments, and payment profiles. Here are some resources that give more information about the different aspects of payments:
    - Logging Payments: Log Payment Article, Log Payment API Article
    - Processing Payments: Process Payment API Article

    - Automatic Payments (Autopays): Autopay Article, Autopay API Article, Create an Autopay Plunker Sample, Create an Autopay Plunker Presentation
    - Payment Profiles: Create Payment Profile API Article
- Loans - Any loan data you want to display to your customer should be available through the ***/odata.svc/Loans()*** endpoint. All loan-related entities can be pulled along with the base loan entity. This includes: loan setup, loan settings, collateral, insurance, historical archive, payments, charges, transactions, and more. For more information on the loans endpoint, look at these resources: Create a Loan Plunker Sample, Restler Documentation, Create a Loan API Article, Create a Loan Plunker Presentation.

Loan Origination System Integration

It is common for lenders to have a homegrown origination system. For lenders who have made this investment, it is common to integration with a loan servicing platform. This type of integration usually involves creating static data in LoanPro from the origination system. *See the Static Data section below for best practices information.* The two main areas of concern are customer creation and loan creation.

- Customer Creation - There are lots of entities associated with customers that you can create through the customers endpoint (***/odata.svc/Customers***). These include things like address, phone number, references, and employer. Here are some resources that will help you learn more about creating customers through the API: Creating a Customer API Article, Create a Customer Plunker Sample, Create a Customer Plunker Presentation, Restler Documentation.
- Loan Creation - It is important to note that loans are separate from customers and aren't dependent on them in any way. The loan creation process is typically fastest when customers are created first and then linked to loans during loan creation. Here are some resources that give more information on loan creation: Create a Loan Plunker Sample, Restler Documentation, Create a Loan API Article, Create a Loan Plunker Presentation.

NACHA File Processing

Some companies want to use LoanPro to process payments by creating NACHA files and sending these files to their payment processor (possibly a bank). NACHA file creation is actually done by PCI Wallet. Here are some resources that will give you more information about NACHA files: Pulling NACHA Files Plunker Sample, Pulling NACHA Files Plunker Presentation, NACHA Return Codes API Article.

Some companies want to build a Balanced NACHA File, which PCI Wallet does not support. If this is the case, the process to follow is to pull a Payment Breakdown Report from LoanPro, configure the NACHA File using your own in-house resources, then submit the file to your bank for processing.

Architecture

We will not provide any specifics about how your integrated application should be architected, but here are a few best practices to keep in mind:

Security

- Keep the API tokens on the server at all times. It should probably go without saying that you should not send API tokens to the client.
- Store payment profile information only in PCI Wallet. If you choose to store it anywhere else, you will be responsible to meet PCI-DSS standards.
    - Limit the handling of payment card information whenever possible. Don't let payment profile data touch your server.
    - Never send card information to your server, even to set cookies!
- Always use HTTPS
- Never store SSN/SIN, bank account information, or credit card information on your servers or databases in plain text.
    - Always use the latest security policies.
    - If you can't follow security policies, then don't store sensitive data outside of LoanPro.
- Never store passwords in plaintext or unencrypted form.
- Always use the latest security procedure (such as salting) and the latest secure algorithms.
- If you can't follow security policies, never store passwords outside of LoanPro or other single-sign-on sites.

Static Data

There are a few common tasks that most integrations with LoanPro perform that enter static data into the system. Below are some recommended practices when performing these tasks.

Customer Creation

- Before you create a customer, check that the customer doesn't already exist.
- When creating a customer, check for transactional warnings to see if the SSN/SIN, Address, or Phone already matches that of another customer. *See the Transactional Warning section below for more information.*
- Create customers before creating loans where possible. This allows you to link customers to loans during the loan-create process, which will speed things up.
- Combine API calls where possible to limit latency in the customer creation process.

Payment Profile Creation

- Payment profile information should only be stored inside of PCI Wallet, including credit/debit cards and bank accounts (checking & savings).
    - Do not allow payment profile information to touch your platform or it will become subject to PCI-Compliance. This will require that you maintain the PCI standard for data storage and handling. There is no reason to store payment profile data in your application, as PCI-Wallet is already handles this.
    - Do not store plain-text payment information inside of LoanPro in the notes or anywhere else. This will compromise PCI compliance.
- If you need to collect payment profile information within your application or from a web page, use an iframe with the provided form from PCI wallet so the data is entered directly into PCI Wallet and doesn't touch any other application.
    - This form has a separate token which is the only token that will ever make it to a web browser
- When a payment profile is created, immediately store the returned token and profile name in LoanPro. Compliance dicates that PCI Wallet can't keep tokens or give out raw payment profile data, so the token must be stored externally in order to use payment profiles.

Loan Creation

- The loan must exist before you can link other things to it (nested entities) or use tools on it. This means you will need to create the loan before you can do things like update loan collateral, generate custom forms, etc.
- Use as few requests as possible.
- This is the recommended order for the loan creating process:
    - Create the loan.
    - Link customers to the loan (unless the loan is a quick quote). It is best to create customers before the loan is created.
    - Activate the loan.
    - Once the loan is activated:
        - Create autopays.
        - Log payments.
        - Log charges.
        - Create advancements, credits, stop interest dates, past due resets, etc.
- Use a systemized identifier as the displayed account ID. LoanPro will generate a database ID for each loan, but it won't be meaningful or descriptive.
- Don't artificially adjust the loan numbers which you are required to disclose to borrowers by Regulation Z and the Truth In Lending Act. Instead adjust the loan settings to match contracts.
- Generate Truth-In-Lending required numbers using the LoanPro Quick Quote tool, not from third-party software.

Dynamic and Transactional Data

Reports

- Searches are prefered over using reports endpoints because they are faster and employ better error handling.
- Use reports when searches are inadequate for the task you are trying to accomplish.
- Use a robust SDK that detects errors before submitting your requests to the LoanPro servers.
- When using reports, employ a background process that will check the completion status of the report approximately every 30 minutes.
    - Don’t do it too often as it will bog down servers.
    - Don’t do it too infrequently because generated reports are automatically removed around midnight.
- Keep a record of reports that are in progress in your own database to make it faster and easier to check their status. Once the report is completed, remove its database record.

Payments

- Save all payment profile information (credit/debit cards, bank accounts) in PCI-Wallet.
- Process all payments through LoanPro. Processing payments through LoanPro will ensure that payments are correctly posted to loans. You will also have access to payment return or void features if you process through loanpro.
- Reverse all payments through LoanPro.

3rd Party Connections

- Use the LoanPro API instead of the Connections API when possible. This will pull the information you want and will also update appropriate LoanPro fields.
- Don’t create integrations with third parties if they already exist. The third-party services available through Connections include:
    - OFAC Testing - This tool checks customer data against the OFAC list to ensure companies don’t do business with individuals against whom there are official sanctions.
    - Credit Pull - This tool pulls individual credit.
    - ZIP Code Decode - This tool pulls the most prevalent city and state associated with a ZIP code (U.S. only).
    - Address Validation and Standardization - This tool will validate that an address exists and change the format to the one used by the U.S. Postal Service (U.S. only).

    - Phone Carrier Lookup - This tool will look up the carrier with which a phone number is associated.
    - SMS Text Messaging - This tool lets you send SMS messages through the software.

Transactional Warnings

Sometimes when requests are sent to the LoanPro API, the requests won't be immediately processed because the request will impact other things in the software that need to be considered. In this case, the response will contain a transaction warning with a specific message. You can then make a choice to either make a change, or resubmit the request, but ignore the warnings.

It's important that when you send requests to the API, you don't ignore the responses. Often responses will contain error handling, but it is especially important that you don't ignore transactional warnings, because they mean that the requested action was not completed.

Updating data

- Always pull the latest data before an update.
- Make sure to lock the data if your integration has asynchronous activity
    - Use atomic operations
    - Make sure locks are performed across all processes
    - Unlock when the request is successful
- Send back the minimal information needed
    - Send back just the dirty data and the proper "__id" and "__update" flags
    - Send back "__id" and "__update" flags for all entities involved

Testing the Integration (Lenders/Programmers)

It is important that you test your integration before you put it into service and during its use. Much of the testing will be done by hand. If you choose to create unit tests that ping the API, make sure these tests don't run too frequently, and are lightweight so they don't bog down servers and slow down overall performance. A simple test to ensure LoanPro function is to perform a loan search which includes parameters intended to limit the search results to only a few loans.

APPENDIX A - Feature Outline

Feature Name: _____

Feature Main Category:

☐ Loan ☐ Customer ☐ Report

Feature Subcategory(ies): _____

What is the main purpose of the feature? _____

Detailed feature description: _____

What actions will this feature perform?

What data will this feature require?

What data will this feature need to store?

What calculations will this feature require?

Write out any calculations needed using mathematical notation.

Calculation Name:
Calculation Name:
Calculation Name:
Calculation Name: