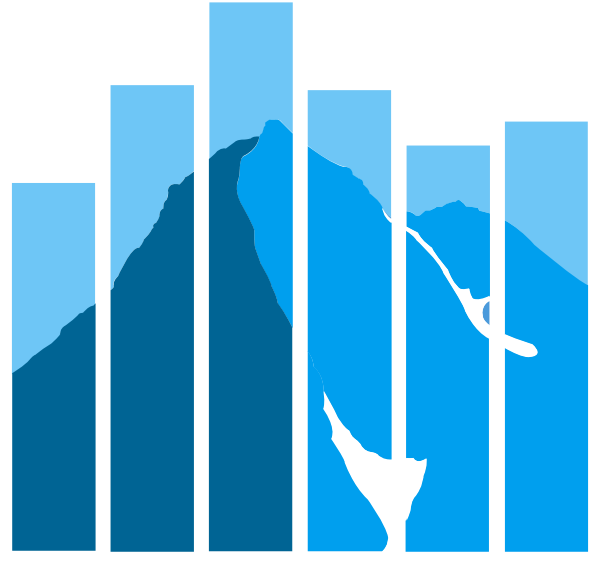


THE LOANPRO API

LOAN CREATION

loanprosoftware.simnang.com

API SAMPLE

Fork Loan Create — https://plnkr.co/edit/vychF4

⚠ CAUTIONARY NOTE ⚠

The provided API samples store API credentials in the browser

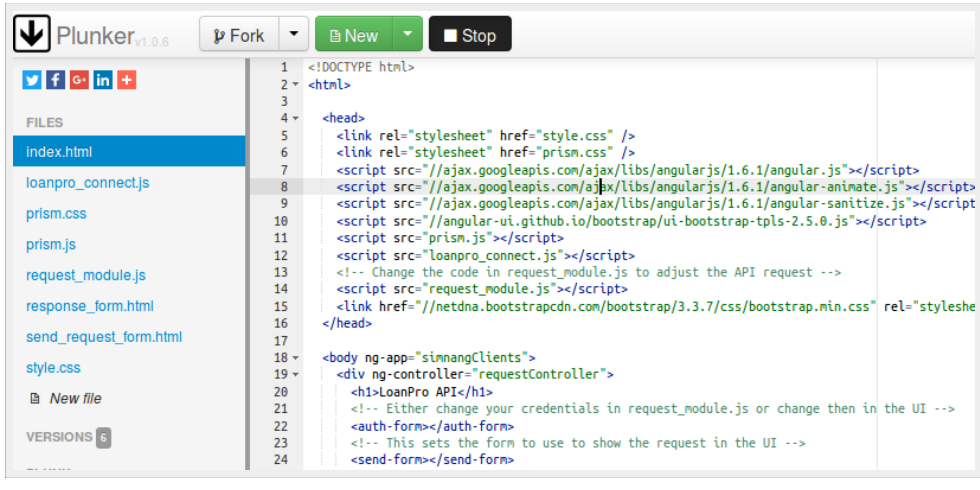
Use the API samples to explore the API and not as your integration

WHY IS THERE A LOANSETUP ENTITY?

When a loan is activated, the numbers in LoanSetup are locked

However, other information in the loan is not locked (eg. display id, settings)

LOANS SAMPLE



Creating Loans →

File structure ↓

INDEX.HTML

- The entry point of the web app
- Includes all requisite libraries (ie. Angular, Prizm, Bootstrap, etc)

LOANPRO_CONNECT.JS

- Backend for communicating to the API via Angular
- Defines directives used to simplify code (ie. "auth-form", "send-form", "response-form")
- Good to use in your app if you are determined to publish your API token and have it stolen 🙃

PRIZM.CSS & PRIZM.JS

- Used for highlighting JSON response from the server
- Not needed in 99.999% of integrations

REQUEST_MODULE.JS

THIS IS THE MOST IMPORTANT FILE!

- Defines the request sent to the server
  - uri - which endpoint to call
  - method - what HTTP method to use
  - data - the JSON data to send for POST, PUT, DELETE
  - query - JSON object representing the URL query
  - title - irrelevant, used to make samples look pretty

REQUEST_MODULE.JS (CONTINUED)

- Also where the sample API key and token are stored

DO NOT STORE API CREDENTIALS HERE AND PUBLISH!

Ok to store on local, but **never** put on the web - including Plunker

REQUEST_MODULE.JS (CONTINUED)

```
1
2  app.controller('requestController', ['$scope','$http', '$sce', '$controller', function($scope,$http, $sce, $controller) {
3    $controller('loanproClient', {$scope: $scope, $http: $http, $sce: $sce});
4
5    // Change the following to match your credentials
6    $scope.api_auth.tenant_id = 5200243;
7    $scope.api_auth.api_token = "cc0199330033f963007b07e75f1b7fb6b7025887";
8
9    $scope.queueAPICall({
10     uri:"odata.svc/Customers",
11     method: "PUT",
12     data: {
13       name: "Bob",
14       employer:{
15         name: "Simnang",
16         job: "Tech Support"
17       }
18     },
19     query: {$expand:"Employer,MailAddress,Phones"},
20     title: "Customers"
21   });
22
23 }]);
```

Sample Tokens

API Endpoint

HTTP Method

JSON data being sent

URL Query

Used to make sample pretty

RESPONSE_FORM.HTML

Displays the results from the server

- Raw tab - shows unformatted JSON response
- Formatted tab - shows formatted JSON response
- Info tab - gives error information

SEND_REQUEST_FORM.HTML

Displays the raw HTTP request that is sent

Has the "Send" button

STYLE.CSS

Holds CSS code to make the samples look good

API ENDPOINT

Loans are created by sending a POST request to:

```
https://loanpro.simnang.com/api/public/api/1/odata.svc/Loans
```

REQUIRED LOAN INFORMATION

- title
  - Human readable name for the loan
- modTotal — set to 0
- active — set to 0
- archived — set to 0
- deleted — set to 0

OPTIONAL LOAN INFORMATION

- displayId
  - Loan ID displayed on the UI
- loanAlert
  - Loan alert to display on the UI

LOAN SETUP

Get ready for a long list (or just read the article API - Loan Setup)

LOAN SETUP PART 1

- loanAmount
  - Amount borrowed
- discount
  - Discount when loan was purchased
- underwriting
  - Underwriting/Finance fee
- loanRate
  - Loan interest rate
- loanRateType
  - The type of interest rate (ie. annual, bi-weekly, etc)
- loanTerm
  - The number of payment periods on the loan

LOAN RATE TYPE

- loan.rateType.annually
- loan.rateType.biweekly
  - 1% bi-weekly = 26% annual
- loan.rateType.monthly
  - 1% monthly = 12% annual
- loan.rateType.semiannually
  - 1% semi-annual = 2% annual
- loan.rateType.semimonthly
  - 1% semi-monthly = 24% annual
- loan.rateType.weekly
  - 1% weekly = 52% annual

LOAN SETUP PART 2

- contractDate
  - Date the contract was signed
- firstPaymentDate
  - Date the first payment comes due
- loanClass
  - Code for the collateral type
- loanType
  - Code for the loan type

LOAN CLASS

- loan.class.carLoan
  - This is for car loans
- loan.class.consumer
  - This is for consumer loans
- loan.class.mortgage
  - This is for real estate loans
- loan.class.other
  - This is for other loans

LOAN TYPE

- loan.type.creditLimit
  - This is for credit limit loans
- loan.type.flooring
  - This is for flooring loans
- loan.type.installment
  - This is for installment loans
- loan.type.lease
  - This is for leases

FUN FACTOID

- When was the World Bank's first loan given?
  - May 9th, 1947

LOAN SETUP PART 3

- discountSplit
  - If discount income should be recognized over the life of the loan
- discountCalc
  - Calculation of how much discount income is recognized with each payment
- paymentFrequency
  - Frequency with which payments come due
- calcType (loan.calcType.simpleInterest)
  - Calculation type for the loan
- daysInYear
  - Number of days in the year to use in calculations
- begEnd
  - Payments come due at the beginning or end of the payment period

DISCOUNT CALC

For information on how each is calculated, see the article Discount Calculation

- loan.discountCalc.full
- loan.discountCalc.percentage
- loan.discountCalc.percentFixed
- loan.discountCalc.rebalancing
- loan.discountCalc.straightLine

PAYMENT FREQUENCY

- loan.frequency.annually
- loan.frequency.biWeekly
- loan.frequency.custom
  - Payments are due at a custom interval
- loan.frequency.monthly
- loan.frequency.quarterly
- loan.frequency.semiannually
- loan.frequency.semiMonthly
- loan.frequency.single
- loan.frequency.weekly

CALCULATION TYPE

- loan.calcType.interestOnly
  - Interest-only loans.
- loan.calcType.rule78
  - Rule-78 calculation (these loans are illegal in many areas)
- loan.calcType.simpleInterest
  - Simple interest loans (most common)
- loan.calcType.simpleIntLocked
  - This is for simple interest locked loans. Simple interest locked means no more and no less than the originally calculated interest amount will come due on the loan.

DAYS IN YEAR

- loan.daysInYear.actual
  - Use the actual number of days in a year
- loan.daysInYear.frequency
  - Use a number of days in a year that is calculated based on the payment frequency
  - number of days in a payment period x number of payment periods in a year

BEGINNING / END

- loan.begend.beg
  - Payment comes due in the beginning of the payment period
- loan.begend.end
  - Payment comes due at the end of the payment period

LOAN SETUP PART 4

- firstPeriodDays
  - How to calculate number of days in first period
- firstDayInterest
  - If interest is charged on the first day
- diyAlt
  - If should use alternate days in year calculation
- daysInPeriod
  - Number of days in a period for custom periods
- roundDecimals
  - Number of decimal places to round to for calculations
  - between 2 and 7

FIRST PERIOD DAYS

- loan.firstPeriodDays.actual
  - Calculations are based on the actual number of days in the first period
- loan.firstPeriodDays.forceRegular
  - Calculations are based on a regular number of days in the first payment period regardless of the number of days in that period
- loan.fristPeriodDays.frequency
  - Calculations are based off a regular length first payment period if the length of the period is within one day of a regular period

FIRST DAY INTEREST

- loan.firstdayinterest.no
  - Do not charge interest on the first day of a loan
- loan.firstdayinterest.yess
  - Do charge interest on the first day of the loan

DAYS IN YEAR ALTERNATE

- loan.diyAlt.no
- loan.diyAlt.yes

DAYS IN PERIOD

See Loan Setup Collections for the rest of period lengths

- loan.daysinperiod.1
  - Payment due every day
- loan.daysinperiod.1B
  - Payment due every business day (Monday through Friday)
- loan.daysinperiod.2
  - Payment due every other day

FUN FACTOID

- When did the first web browser prototype get released?
  - Tim Lee's prototype browser came out on the NeXT computer in 1990

LOAN SETUP PART 5

- lastAsFinal
  - The last payment amount will grow or shrink to make the last scheduled payment the actual last payment
- nddCalc (loan.nddCalc.standard)
  - Next due date calculation
- endInterest
  - Whether interest will accrue after the originally scheduled final payment
- feesPaidBy
  - Whether payments apply to fees assessed in the same payment period or before, or assessed chronologically before the payment applies
- paymentDateApp
  - Payment date application

LAST AS FINAL

- loan.lastasfinal.no
  - Add payments to the schedule to adjust for missed payments or extra interest accrual
- loan.lastasfinal.yes
  - Make the last loan payment larger to adjust for missed payments or extra interest accrual

NEXT DUE DATE CALCULATION

Determines how to calculate the next due date. See Next Due Date (NDD) Calculation

- loan.nddCalc.interestOnly
- loan.nddCalc.standard

END INTEREST

- loan.endInterest.loanExp
- loan.endInterest.no

FEES PAID BY

- loan.feesPaidBy.date
- loan.feesPaidBy.period

PAYMENT DATE APPLICATION

- loan.pmtdateapp.acutal
  - Actual/Next
- loan.pmtdateapp.last
  - Last/Next

LATE FEES

- lateFeeType
  - Type of late fee
- lateFeeCalc
  - Late fee calculation
- lateFeeAmount
  - Fixed amount for late fee
- lateFeePercent
  - Percentage for late fee calculation
- lateFeePercentBase
  - Base for late fee percentage calculation
- graceDays
  - Days after a payment comes due before a late fee is assessed

NUMBERS THAT DON'T AFFECT CALCULATIONS

- amountDown
  - Down Payment
- reserve
  - Amount held back from dealer/store
- salesPrice
  - Original sales price of collateral
- gap
  - Amount of the GAP Insurance
- warranty
  - Amount covered by warranty
- dealerProfit
  - Amount of profit made by dealer/store
- taxes
  - Amount paid in taxes (for leases, use the Escrow bucket)

LOAN SETUP SPECIAL NUMBERS

- creditLimit
  - Credit limit of the loan
  - For credit limit loans only
- curtailPercentBase (loan.curtailpercentbase.loanAmount)

SAMPLE LOAN CREATION

```
POST https://loanpro.simnang.com/api/public/api/1/odata.svc/Loans
```


```
{
  "displayId": "L001001",
  "title": "My First Loan",
  "modTotal": 0,
  "active": 0,
  "archived": 0,
  "loanAlert": "",
  "deleted": 0,
  "LoanSetup": {
    "loanAmount": "12000.00",
    "discount": "500.00",
    "underwriting": "0.00",
    "loanRate": "12.0212",
    "loanRateType": "loan.rateType.annually",
    "loanTerm": "36",
    "contractDate": "2015-05-07",
    "firstPaymentDate": "2015-05-08",
    "amountDown": "0.00",
    "reserve": "5.00",
    "salesPrice": "12000",
    "gap": "1120.",
    "warranty": "2500"
```

WHY THE LONG LIST?

- LoanPro supports almost every loan
  - No standard "cookie-cutter" loan exists in the industry
  - LoanPro supports over 300,000 loan calculation methods

WHAT DO I DO WITH THE LONG LIST?

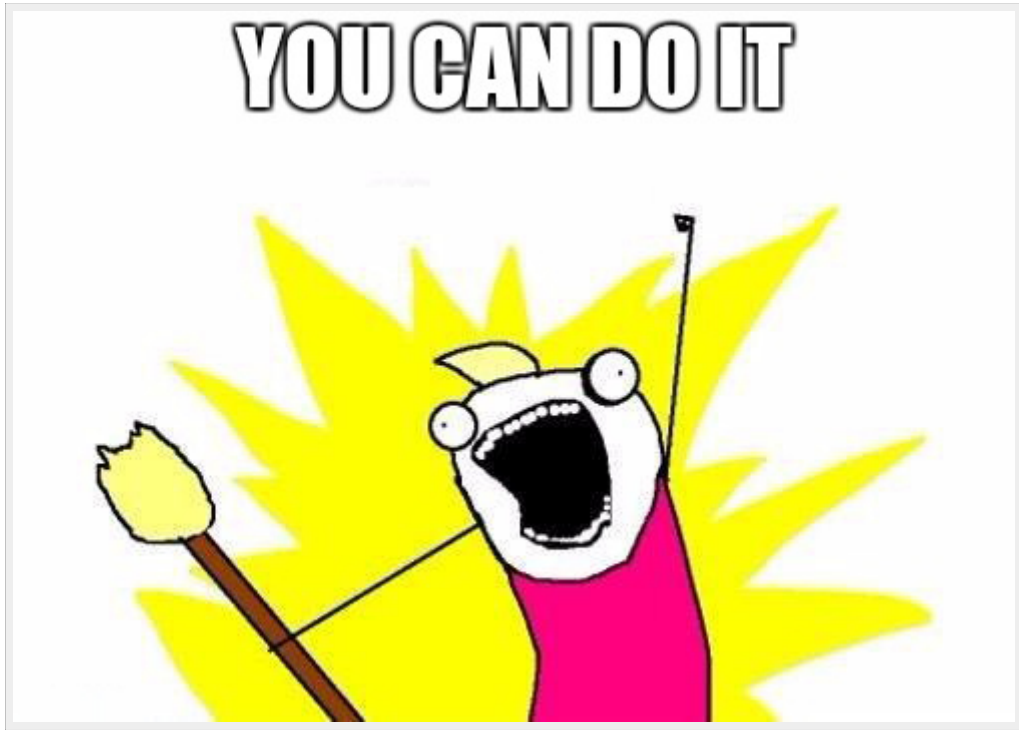
- Work with your finance people and write down the settings your loans use
- Create cookie-cutter loan templates in you code using those settings
- Create a layer of abstraction on-top of your templates to make loan creation easy

ARTICLES

- API - Create a Loan
- API - Loan Setup
- Loan Setup Collections
- Discount Calculation
- First Period Days
- Loan Type
- Calculation Type
- Setup Terms
- Search for "Loan Setup"

FINAL THOUGHTS

- Loan creation isn't scary once you know what you want
- Use the UI to understand the process, then transition to the API
- Remeber:



QUESTIONS?

Feel free to ask in our Forums

Or, email us at support@simnang.com