



LEND

CONNECT 2020

SMALL DOLLAR, SHORT-TERM, CONSUMER LENDING CONFERENCE



EPIC API DEEP DIVE

Ken Maier
VP Technology
EPIC Loan Systems

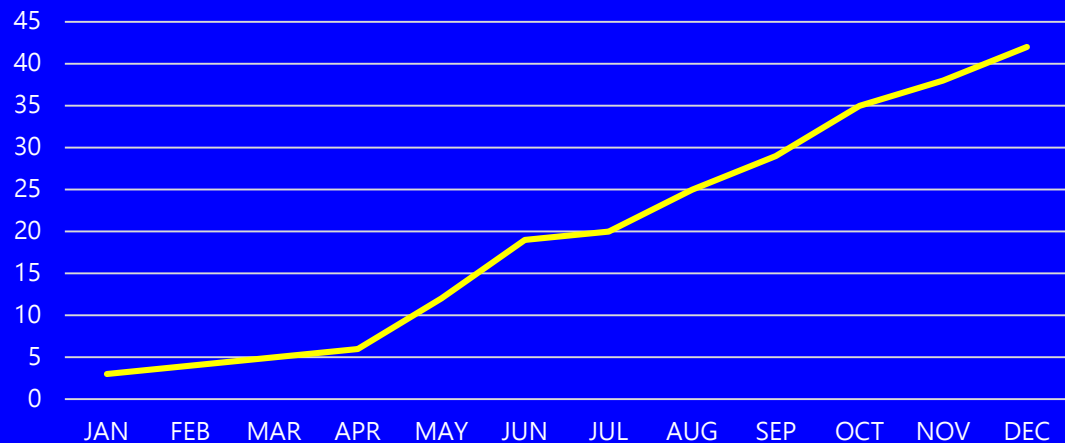


A large, dark blue, irregularly shaped graphic resembling a watercolor splash or ink blot, centered on a white background. The edges of the splash are textured and feathered, with some smaller droplets or splatters extending outwards. The year "2019" is written in a large, white, sans-serif font, centered within the dark blue area.

2019

Goal : Improve agility and reduce time for customers to receive updates

Production Updates



(LMS=23, API=19, ~3.5/m)

- 42 Releases in 2019
- Over 1200 updates

Notable Items in 2019

- Enforced Principal
- Revoke ACH (for CFPB compliance)
- Support for new Clarity products
- Email support for Decision Logic auto-start
- New ORR report
- SablePay ACH integration with same day credits
- Idology Email Product
- Pending Application queue integration with EPIC checklists



API DEEP DIVE

API Goals



PROVIDE ACCESS TO
READ AND WRITE
DATA WITHIN THE
PORTFOLIO OUTSIDE
OF THE EPIC LMS WEB
INTERFACE



PROVIDE THE ABILITY
FOR LENDERS TO
CREATE EXTERNAL
SOLUTIONS THAT
COMPLEMENT EPIC
LMS FUNCTIONALITY



PROVIDE THE ABILITY
FOR EPIC TO
INTERNALLY USE THE
API FOR LMS USER
INTERFACE
DEVELOPMENT

API Features and Functionality












EPIC Loan Systems API

Use the EPIC Loan System's API to interact with your portfolio data and to develop third-party apps that incorporate and build upon EPIC's core product features. This documentation is for the EPIC v1 API. The following API calls represent the functionality currently available to EPIC customers. You must be a verified user to make API requests.

Config		Show/Hide	List Operations	Expand Operations
Customers		Show/Hide	List Operations	Expand Operations
GET	/api/v1/customers	Returns a collection of CustomerID 's for every customer in the portfolio matching search criteria provided		
GET	/api/v1/customers/{customerId}	Returns a customer object for the provided CustomerID		
PUT	/api/v1/customers/{customerId}	Update existing customer		
GET	/api/v1/customers/{customerId}/contactStatus	Returns phone, SMS, and email status for the provided CustomerID		
GET	/api/v1/customers/{customerId}/documents			
POST	/api/v1/customers/{customerId}/documents	Adds a new document (multipart/form-data; boundary)		
GET	/api/v1/customers/{customerId}/documents/{documentId}	Returns a document object for a specific DocumentID		
GET	/api/v1/customers/{customerId}/documents/{documentId}/url	Returns a time limited signed URL for a document		
GET	/api/v1/customers/{customerId}/loans	Returns a collection of LoanID 's associated with a customer matching the search criteria provided		
GET	/api/v1/customers/{customerId}/notes	Returns a collection of Notes associated with a customer matching the search criteria provided		
POST	/api/v1/customers/{customerId}/notes	Adds a new note		
Loans		Show/Hide	List Operations	Expand Operations
GET	/api/v1/loans	Returns a collection of LoanID 's for every loan in the portfolio matching search criteria provided		

- Production availability **May 2016**
- **Read and write API**
- **REST design** is programming language agnostic
- Token based **authentication**
- **Self-managed** credentialing system
- **Sandbox environment** for testing
- **Version** control for backwards compatibility
- Online **documentation**
- Designed with **security, performance, and scalability** from inception

EPIC Data Access Comparison

	DATA EXTRACTS	DART	API
Bulk Access			
Data Richness			
Data Recency			
Updatable			
Recommended Usage	Reporting and analysis where delayed updates are okay	Reporting and analysis when real-time data is required	Best for interactive fast-response applications and/or where updating data is required

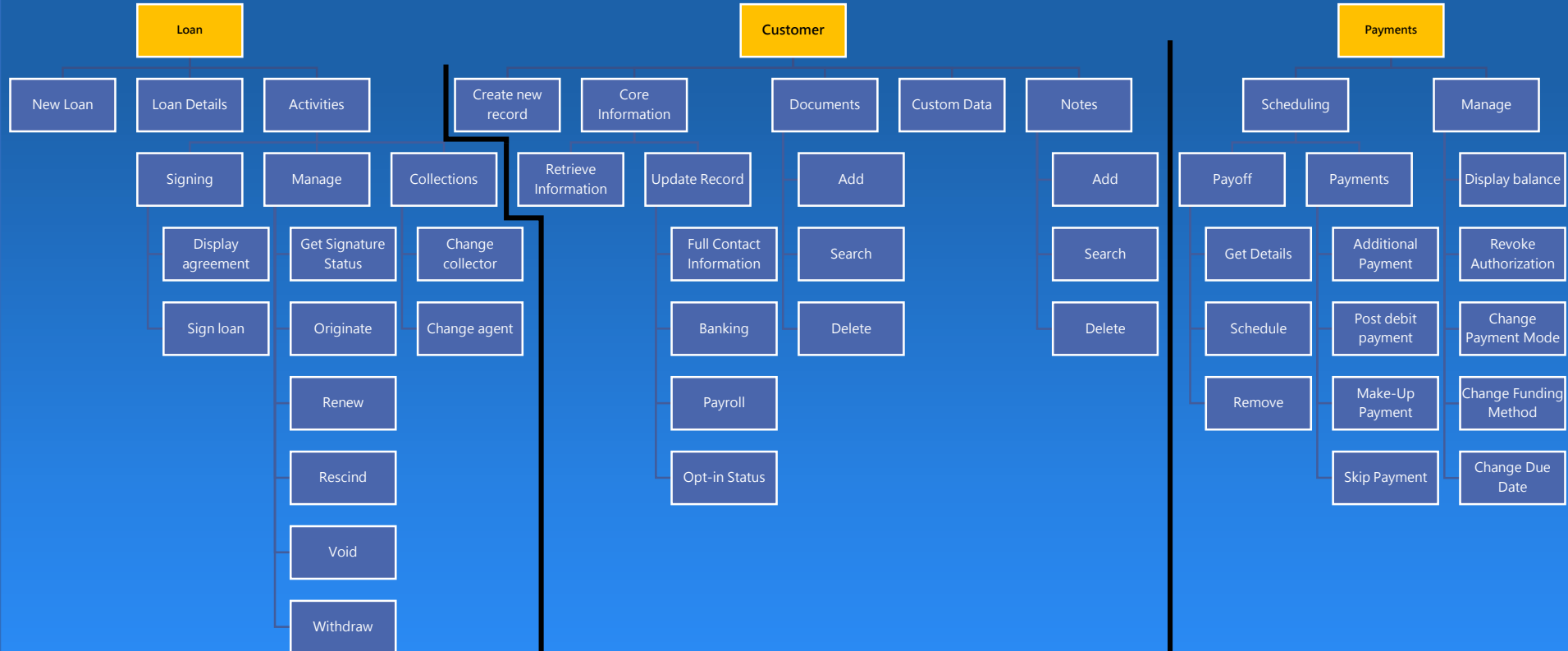
 Good
 Better
 Best



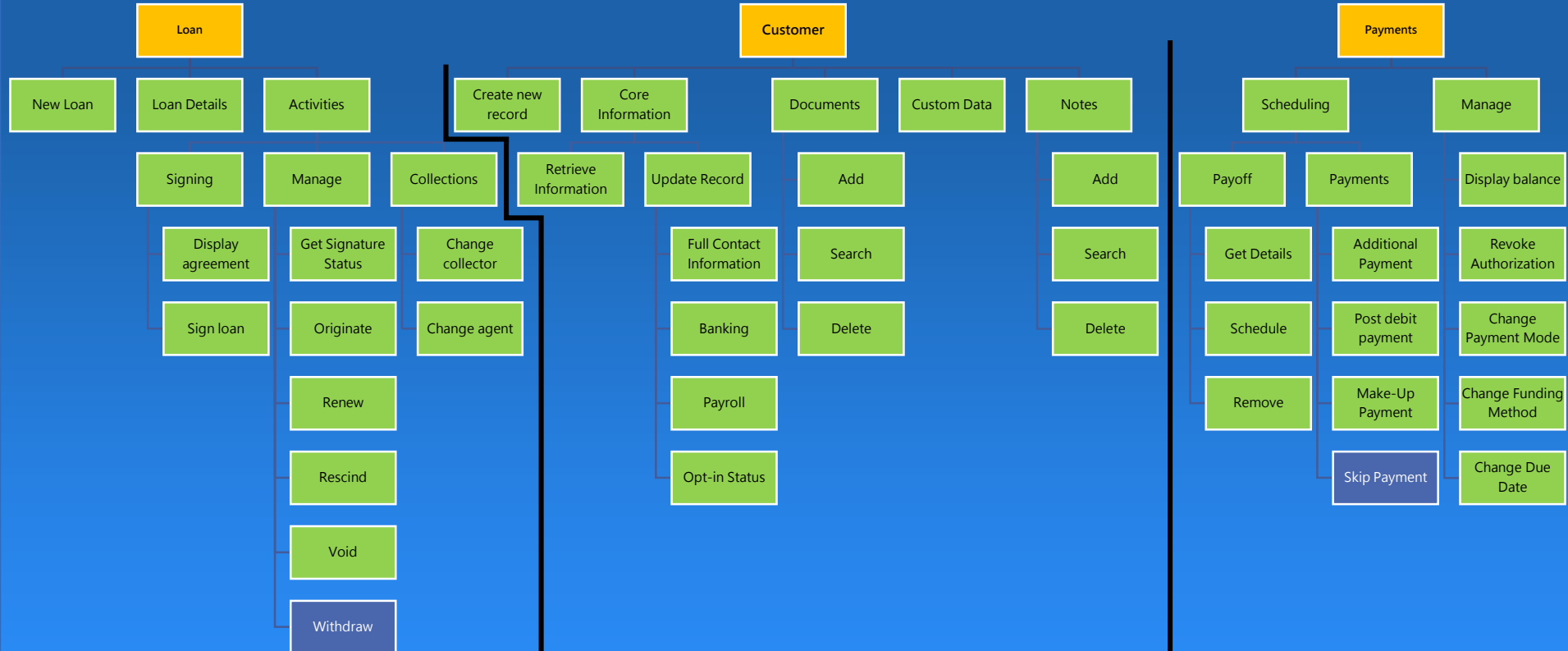
API functionality has been focused on...

- black-box integrations
- self-service portals
- streamlining operations

Portal Activities



Portal Activities















Getting started with the EPIC API

API Credential Management

- Self-Service Credential Management
- Ability to manage security of access
- Create up to 10 different credentials
- Production and Sandbox















API Credentials					
Name	Access Key	Type	Date Created	Last Used	
External Development Team	AGR5qPScNVBUQ4bS0DpKDZZlzz...	Sandbox	01-08-2020 09:16:25		 Edit  Delete
John	AFwrrTL3m8ndqGnbwlNdy2X/NafV...	Production	12-05-2019 13:10:41		 Edit  Delete
Cyn	AKHotQXalTT+A6yKJEjpJf18Peray...	Production	08-21-2019 08:32:46		 Edit  Delete
Ken Test	AJF4MSjn9PGYGEwwFKDS+nwVo...	Production	08-02-2019 10:31:07		 Edit  Delete
API Team	ANStuqVwm4cyC4bCgd6mL/wU/6d...	Production	05-24-2018 11:06:14	06-13-2019 08:00:34	 Edit  Delete
paulo	AA340XFPj2ShppqxVyr1q1O25E4td...	Production	11-14-2017 08:59:09	01-17-2020 10:09:48	 Edit  Delete

API Credential Management

Create up to 10 credentials with easy to remember names

Both production and sandbox credentials can be created

Know when credentials are created and last used

API Credentials					
Name	Access Key	Type	Date Created	Last Used	
External Development Team	AGR5qPScNVBUQ4bS0DpKDZZlzz...	Sandbox	01-08-2020 09:16:25		 Edit  Delete
John	AFwrrTL3m8ndqGnbwINdy2X/NafV...	Production	12-05-2019 13:10:41		 Edit  Delete
Cyn	AKHotQXaITT+A6yKJEjpJf18Peray...	Production	08-21-2019 08:32:46		 Edit  Delete
Ken Test	AJF4MSjn9PGYGEwwFKDS+nwVo...	Production	08-02-2019 10:31:07		 Edit  Delete
API Team	ANStuqVwm4cyC4bCgd6mL/wU/6d...	Production	05-24-2018 11:06:14	06-13-2019 08:00:34	 Edit  Delete
paulo	AA340XFPj2ShppqxVyr1q1O25E4td...	Production	11-14-2017 08:59:09	01-17-2020 10:09:48	 Edit  Delete

Immediate credential termination



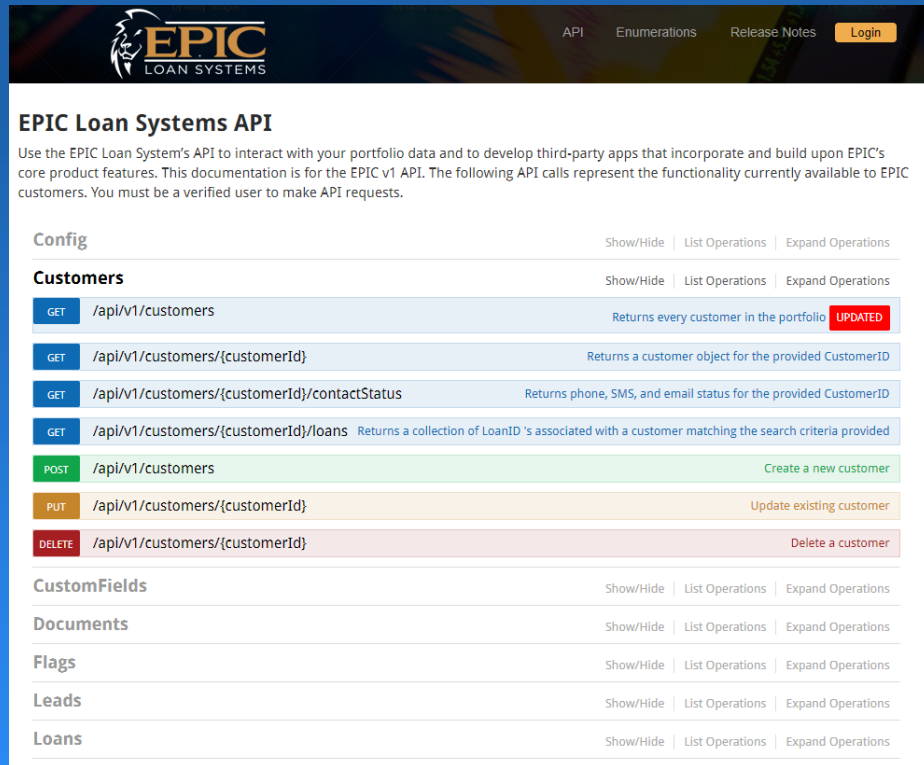
PERHAPS YOU SHOULD

Have read the manual first

Interactive Documentation

- Designed for **self-service**
- **Online documentation** for all available API methods with JSON structures
- Indicators of what's changed
- **Interactive** API queries
- Online list of **enumerations**
- **Release notes**

<https://api.epicloansystems.com/help>
<https://api.sandbox.epicloansystems.com/help>



The screenshot displays the EPIC Loan Systems API documentation page. At the top, there is a navigation bar with the EPIC LOAN SYSTEMS logo, links for API, Enumerations, Release Notes, and a Login button. The main heading is "EPIC Loan Systems API", followed by a descriptive paragraph about the API's purpose. Below this, a table lists various API endpoints categorized by function: Config, Customers, CustomFields, Documents, Flags, Leads, and Loans. Each category has a "Show/Hide", "List Operations", and "Expand Operations" link. The "Customers" section is expanded, showing a list of endpoints with their methods (GET, POST, PUT, DELETE), URLs, and descriptions. Each endpoint entry has a colored button indicating its status: "UPDATED" (red), "Create a new customer" (green), "Update existing customer" (orange), and "Delete a customer" (red).

Method	Endpoint	Description	Status
GET	/api/v1/customers	Returns every customer in the portfolio	UPDATED
GET	/api/v1/customers/{customerId}	Returns a customer object for the provided CustomerID	
GET	/api/v1/customers/{customerId}/contactStatus	Returns phone, SMS, and email status for the provided CustomerID	
GET	/api/v1/customers/{customerId}/loans	Returns a collection of LoanID 's associated with a customer matching the search criteria provided	
POST	/api/v1/customers	Create a new customer	
PUT	/api/v1/customers/{customerId}	Update existing customer	
DELETE	/api/v1/customers/{customerId}	Delete a customer	

Interactive Documentation

Login with API credentials

GET /api/v1/customers/{customerId} Returns a customer object for the provided CustomerID

Response Class (Status 200)

Model | Model Schema

```
{
  "Language": 1,
  "LastName": "Johnson",
  "NumberCurrentResidenceMonths": 19,
  "NumberCurrentResidenceYears": 4,
  "PhoneFaxDisconnected": false,
  "PhoneFaxDollotCall": false,
  "PhoneHome": "(653)555-9580",
  "PhoneMobile": "(261)555-4827",
  "SSN": "121111843",
  "State": "GA",
}
```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
customerId	8671	Unique Customer Id	path	Integer

Try it out! | Hide Response

Curl

```
curl -X GET --header 'Accept: application/json' "https://api.epicloansystems.com:443/api/v1/customers/8671?Auth=31f873e47583404a85ee031b2cc73b10|QUYwQnp0TF1xQzYrQ1BwJS9rR2NzZ2"
```

Request URL

```
https://api.epicloansystems.com:443/api/v1/customers/8671?Auth=31f873e47583404a85ee031b2cc73b10|QUYwQnp0TF1xQzYrQ1BwJS9rR2NzZ2
```

Response Body

```
{
  "customerId": 8671,
  "address": "Acacia St.",
  "address2": "4863",
  "bankAccountAge": 161,
  "bankAccountNumber": "18800660400122477",
  "bankAccountType": 1,
  "bankCountryId": null,
  "bankName": "Bank of America"
}
```

Try it out!

Enter all input parameters

Curl and Request URL samples

Response body or error message

... when I post leads I cannot find the loans. #frustrated

Lenders routinely use front-end systems for initial lead vetting before posting into EPIC.

This created two challenges:

1. How to store reference to accepted leads in the front-end systems?
2. How to 'link' the front-end system with EPIC's data?

- The lead token feature extends the lead post process to also include an encrypted key
- Using this key with the API will return specific information about the lead that was accepted
- Lenders can optionally enable lead tokens.

```
<?xml version="1.0" encoding="utf-8"?>
  <Response xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Result>A</Result>
  <RedirectUrl>https://alpo.epicloansystems.com/redirect.aspx?riid=82ggh6d0-aac5-48b2-89cb-
a54c64bbf2a4</RedirectUrl>
  <LeadTokenId>sdf4555gsfskjdfhskdjshdf72kjsdfskjdhfs</LeadTokenId>
</Response>
```


- Returns details about an accepted lead
- Must have leadToken feature enabled in portfolio
- Useful to know the specific record information that is created in EPIC
- Result of this call returns the customerId and loanId
- Example:

```
{  
  "customerId" : "12345",  
  "loanId" : "42313"  
}
```


... but I want to push data into the system. #confused

Custom Fields

Custom fields permit passing in data during the lead post process and storing it with the customer record.

- 10 fields are available today
 - Can be set during lead post or API
- Available as smart tags
- Not available on user interface
- API methods available for retrieval and update

Custom Fields (Lead Posting)

Custom fields are passed in a new section in the lead post. See lead post documentation for more details.

```
<CUSTOMFIELDS>
  <CUSTOMFIELD>
    <ID>1</ID>
    <DESCRIPTION>Example text describing the field</DESCRIPTION>
    <VALUE>Z3R4TUB12343</VALUE>
  </CUSTOMFIELD>
  <CUSTOMFIELD>
    <ID>3</ID>
    <DESCRIPTION>Example text describing another field</DESCRIPTION>
    <VALUE>DFDFDFDF</VALUE>
  </CUSTOMFIELD>
</CUSTOMFIELDS>
```


Custom Fields (Lead Posting)

Custom fields are available as smart tags under the "Client" folder.

```
@@client.CustomFields.Field01@@  
@@client.CustomFields.Field02@@  
@@client.CustomFields.Field03@@  
@@client.CustomFields.Field04@@  
@@client.CustomFields.Field05@@  
@@client.CustomFields.Field06@@  
@@client.CustomFields.Field07@@  
@@client.CustomFields.Field08@@  
@@client.CustomFields.Field09@@  
@@client.CustomFields.Field10@@
```


Custom Fields (API Methods)

CustomFields

[Show/Hide](#)[List Operations](#)[Expand Operations](#)

GET	/api/v1/customers/{customerId}/customFields	List custom fields for a customer
GET	/api/v1/customers/{customerId}/customFields/{customFieldId}	Returns specific custom field details for a customer
PUT	/api/v1/customers/{customerId}/customFields/{customFieldId}	Update custom field

```
{
  "CustomFieldId": 1,
  "Description": "Reference BlackBox Id",
  "Value": "ABC123",
  "DateUpdated": "2020-01-19T00:00:00-06:00"
}
```

Use to keep track of foreign key references, underwriting ID's for loan reporting, other external information

~~... but I want to push data into the system. #confused~~

... can I upload photos or agreements generated
outside EPIC? #stillconfused

Documents

Robust document management API's permit storage and retrieval of any document type in a customer account.

- System generated loan agreements available as PDF
- Upload your own loan agreement, photos, documents, or any other KYC information

Documents

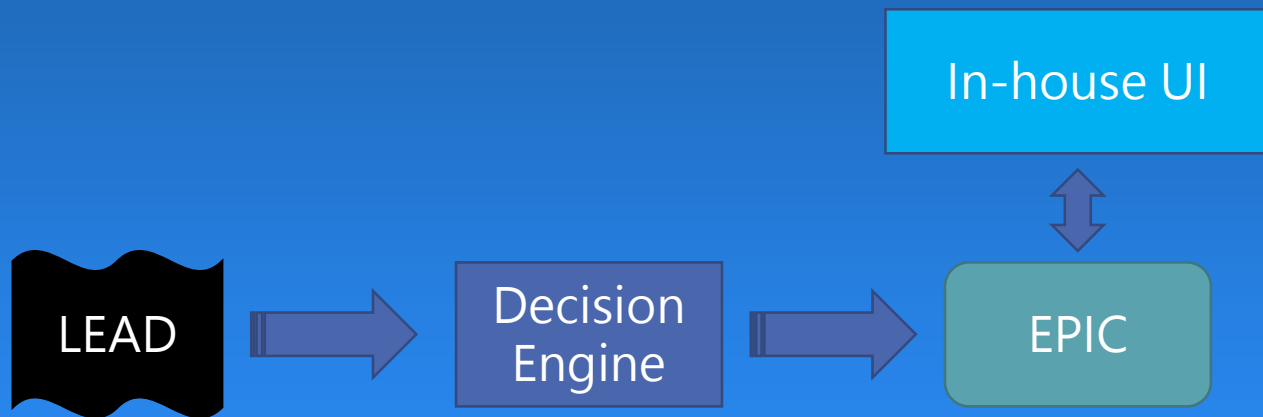
[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/v1/customers/{customerId}/documents	Returns a collection of document objects associated with the provided CustomerID
GET	/api/v1/customers/{customerId}/documents/{documentId}	Returns a document object for a specific DocumentID
GET	/api/v1/customers/{customerId}/documents/{documentId}/url	Returns a time limited signed URL for a document
POST	/api/v1/customers/{customerId}/documents	Adds a new document (multipart/form-data; boundary)
PUT	/api/v1/customers/{customerId}/documents/{documentId}	Update document metadata for a specific DocumentID
DELETE	/api/v1/customers/{customerId}/documents/{documentId}	Delete a document by DocumentID

... I've created an application for my agents, but I
need to be able to automate some activities
#poweruser

Integrating with an in-house application

Streaming lining an in-house application to work with incoming applications



Flags

Manage portfolio flags and set flags for customers.

- Integrate with your local workflow to keep track of events that have been completed for a customer

Flags

Show/Hide | List Operations | Expand Operations

GET

/api/v1/customers/{customerId}/flags

List enabled flags for a customer

GET

/api/v1/flags

List flags for a portfolio

POST

/api/v1/flags

Add a new flag

PUT

/api/v1/customers/{customerId}/flags/{flagId}

Update flag status for a customer

PUT

/api/v1/flags/{flagId}

Update a flag name

DELETE

/api/v1/flags/{flagId}

Delete a flag

Pending Application Management

Challenges when working with pending applications:

- How do you change values on a pending application?
- Can the assigned loan manager be changed?
- Can the effective date be changed?
- Can a loan be moved into a different store?

Pending Application Management

Challenges when working with pending applications:

- How do you change values on a pending application?

Yes!

- Can the assigned loan manager be changed?

Yes!

- Can the effective date be changed?

Yes!

- Can a loan be moved into a different store?

Yes!

Pending Application Management

How do you change values on a pending application?

PUT /api/v1/loans/{loanId}/pendingApplication/recalculation

Change program, interest rate, approved principal amount, number of installments on a loan

NEW

- Change Program
- Change Principal Amount
- Change Interest Rate
- Change Number of Installments
- A payment schedule recalculation will automatically occur

Pending Application Management

Can the assigned loan manager be changed?

PUT

/api/v1/loans/{loanId}/agent

Update current employee assignments on a loan

- When a loan is a pending application, changing the assignment will update the **assignedLoanManagerId**
- You can also unset any assigned agent

Pending Application Management

Can the effective date be changed?

PUT

/api/v1/loans/{loanId}/effectiveDate

Update the effective date of a loan

- Provide a new effective date to update the loan

Pending Application Management

Can a loan be moved into a different store?

PUT

/api/v1/loans/{loanId}/changestore/{storeId}

Changes the store for a given loan id

- Provide a new store Id
- Great for 'sorting' or 'managing' pending applications if criteria has changed after the lead post
- Ability to perform underwriting when moving to the new store

Collections Management

When working with collections, we now offer the ability to move loans into different collection queues:

PUT

/api/v1/loans/{loanId}/changeQueue/{queueId}

Change the assigned collection queue for a loan

NEW

And the assigned collections agent can be updated using the update agent API:

PUT

/api/v1/loans/{loanId}/agent

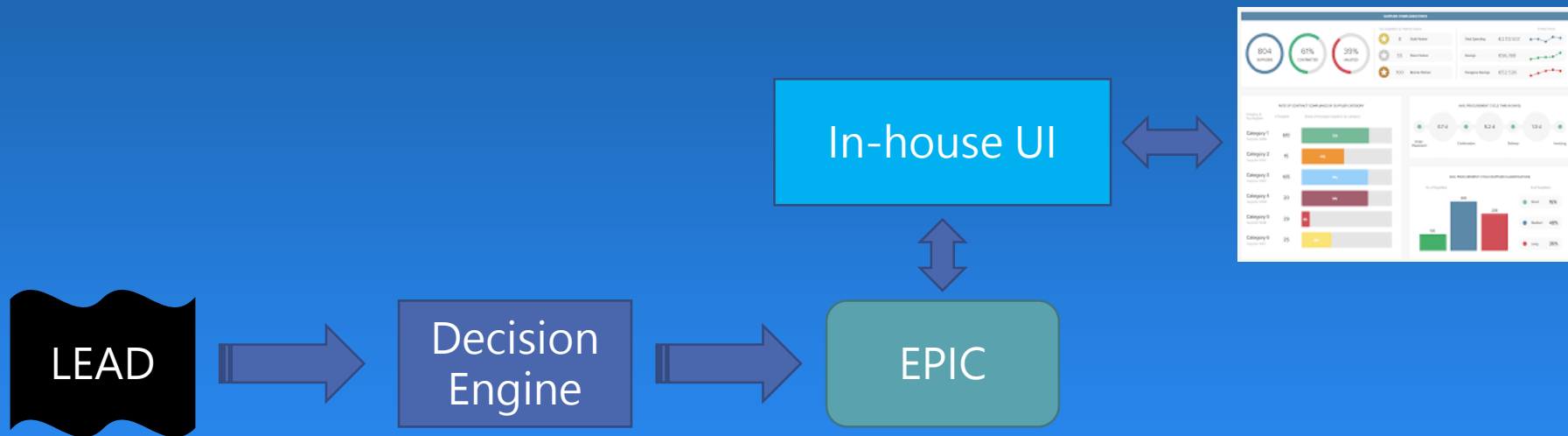
Update current employee assignments on a loan

- Only loans in active collections can be updated
- Employee must be in a collections department to be assigned

... how can I get data out quickly? #speed_demon

Frequent data updates

How keep dashboards or in-house queues updated?



Working with large amounts of data



How to retrieve large amount of API data?

How identify changes without constant polling?

How retrieve it quickly?

Retrieving data by date updated

Using the **dateChanged** field in the customer and loan records, you can zero in on only records with changes that occurred in the timeframe you specified.

Possible use cases:

- More efficient local data synchronization
- Run batch processes that take action on changed records
- Update local dashboards or applications more efficiently

Implementation:

- Available on GET for /customer and /loan as dateChanged
- Either a single date in the format yyyy-MM-ddTHH:mm:ss or a date range in the format yyyy-MM-ddTHH:mm:ss,yyyy-MM-ddTHH:mm:ss to specify the start and end date ranges

More efficient data retrieval

When trying to retrieve data, most common API's invoked to get a full view of a customer:

- Customer record
- Loan record
- Payment information
- Signature information

This results in a minimum of 4 API calls, but most likely more than that due to payment data across multiple retrievals.

GET /api/v1/loans/{loanId}/summary

New summary call retrieves all this information in ONE (1) API call:

- Customer record
- Loan record
- Payment information
- Signature information

The result is at least a 75% savings (1 call vs 4).
The structure returned:

Loan

- Customer
- Signature
- Payments (array)

... what would a workflow look like? #put_it_all_together



Managing the user end-to-end

Some Lenders want to control the entire user experience from landing page to signing.

- Users do not need to log into a different system to sign an agreement
- One unified, branded experience
- Create “dynamic” agreements to present options to users

With the EPIC API's:

- Retrieve and display loan agreement (using our template or yours)
- Target a different store
- Change pending app elements
- Internal or external PDF generation
- Signing the agreement



Example : Display and sign loan agreement

Using the EPIC API, the customer experience can be completely controlled externally from agreement presentation to signing.

- Display local agreement
- Generate loan agreement locally (PDF)
- Sign loan via signing API
- Upload agreement via document API

Let's walk through this in more detail...

Example : Display and Sign Agreement

Display local agreement	GET /api/v1/loans/{loanId}/preloanagreement GET /api/v1/loans/{loanId}/loanagreement
Change payment mode (ACH,Wire,etc)	PUT /api/v1/loans/{loanId}/paymentMode
Change due date	PUT /api/v1/loans/{loanId}/dueDate
Refresh agreement as necessary	GET /api/v1/loans/{loanId}/loanagreement
Update customer opt-in status	PUT /api/v1/customers/{customerId}
Capture consumer signature	(local)
Generate loan agreement locally (PDF)	(local)
Sign loan via signing API	PUT /api/v1/loans/{loanId}/signature
Upload agreement via document API	POST /api/v1/customers/{customerId}/documents
Add notes	POST /api/v1/customers/{customerId}/notes
Update flags	PUT /api/v1/customers/{customerId}/flags/{flagId}

Portal Activities



Go Build!

Keep us posted on what you
are doing!

Thank You
Questions?

Ken Maier



kmaier@epicloansystems.com



api@epicloansystems.com