

Integration between Anthology Student & Custom Payment Gateway Provider Technical Guide – Phase I

Overview	2
Limitations & Practice Recommendations	3
Process Flow	4
Anthology Student Configurations	5
Select Advanced Features Option	5
Add Credit Card/ACH Processor	5
Test Tool	8
API Authentication Method	9
Payment Info	10
Add New Credit Card	10
Add New Bank Account	12
Ledger Card	14
Make Payment With New Credit Card	14
Make Payment Using Saved Credit Card/Student Bank Account	16
Post Credit Card/ACH Payments for Reference Only	18

Overview

The purpose of the document is to provide clients with technical details for a successful integration between Anthology Student and any payment gateway providers that operate under the following concepts:

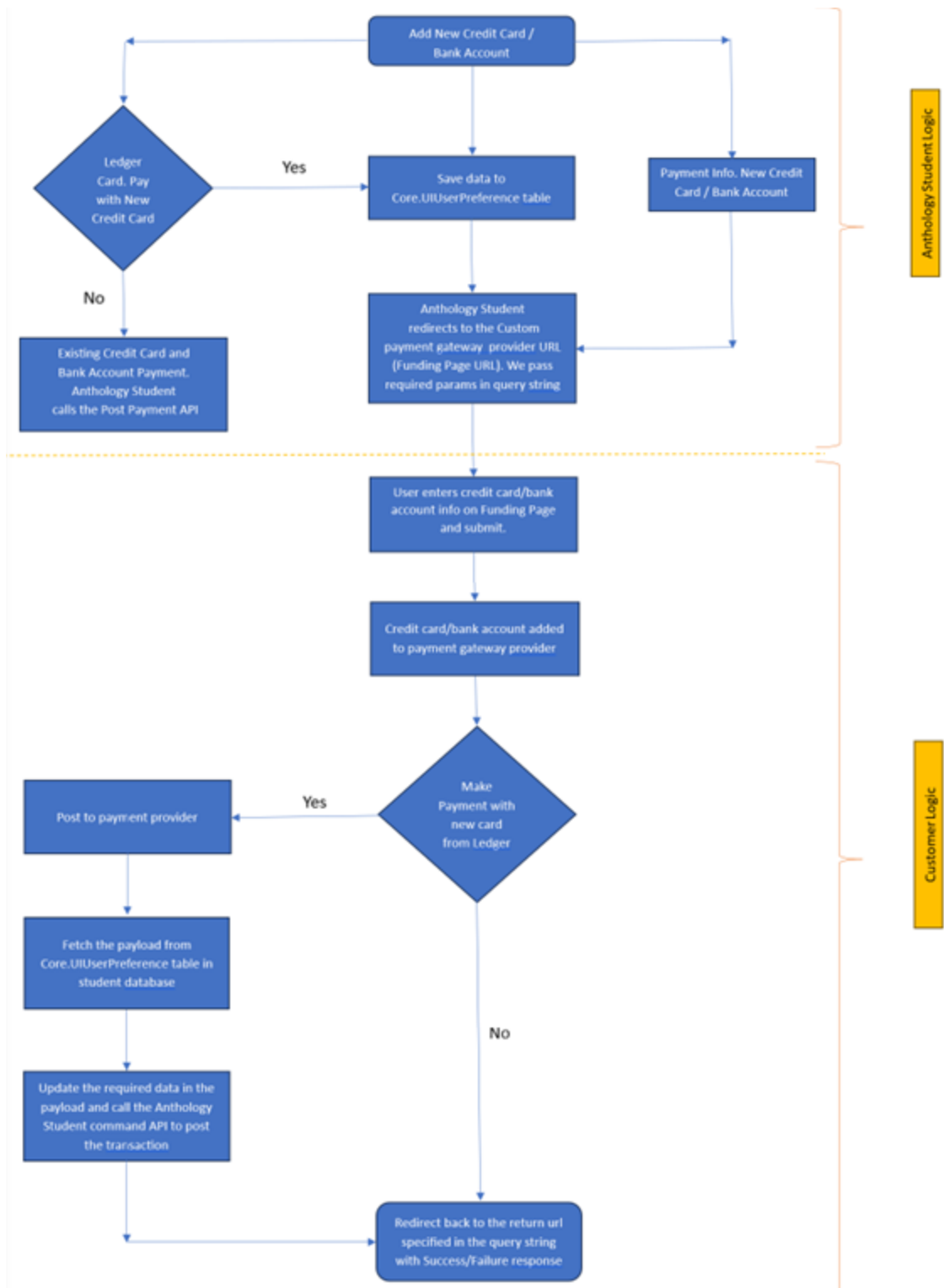
- Allow consumers to add a credit card or bank account on a dedicated funding page.
- The funding page returns a unique identifier for the host application to be stored.
- An endpoint (URL) exists to receive payment requests from the host application.

At a high level, this solution relies on existing payment services in Anthology Student to send transaction requests to the payment gateway providers to be processed. Once the requests are processed successfully, Anthology APIs can be used to post the processed transactions to Anthology Student to complete the process.

Limitations & Practice Recommendations

- This is phase 1 and the scope is limited to:
 - Allow configuration of Credit Card and ACH electronic processors with “Custom” payment gateway provider
 - Allow URL configuration (funding URL and Post URL) for the “Custom” payment gateway provider
 - Add students’ credit cards and bank accounts to the Anthology Student web app
 - Make single credit card/ACH payments through the student’s ledger card.
- Clients must develop their own funding page that can communicate with Anthology Student and be able to use Anthology APIs to insert data into the Anthology Student database.

Process Flow



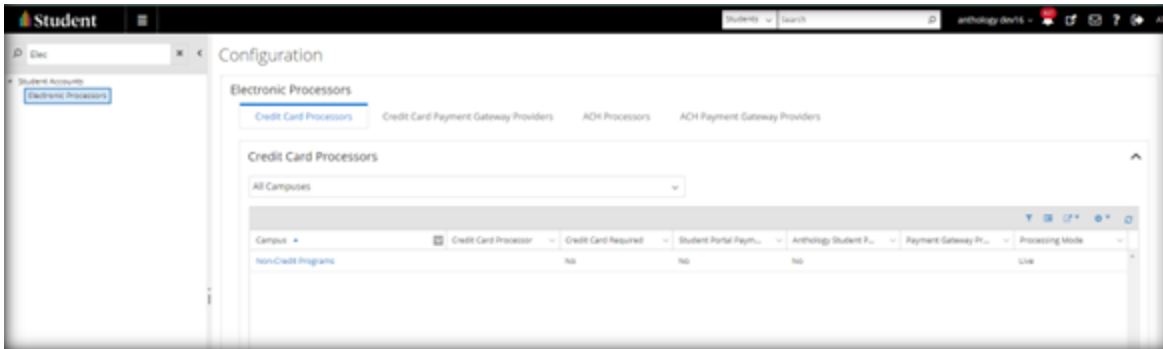
Anthology Student Configurations

Select Advanced Features Option

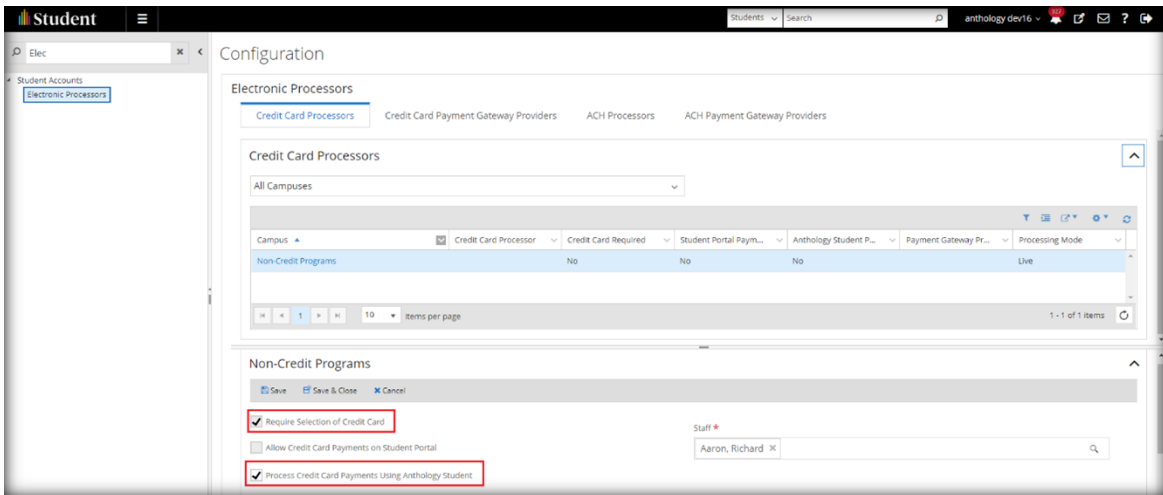
Under Settings > System > Advanced Features, the setting **Enable Additional Payment Gateway Provider for electronic processing?** should be **Yes**.

Add Credit Card/ACH Processor

1. Navigate to Configuration > Student Accounts > **Electronic Processors** tab.



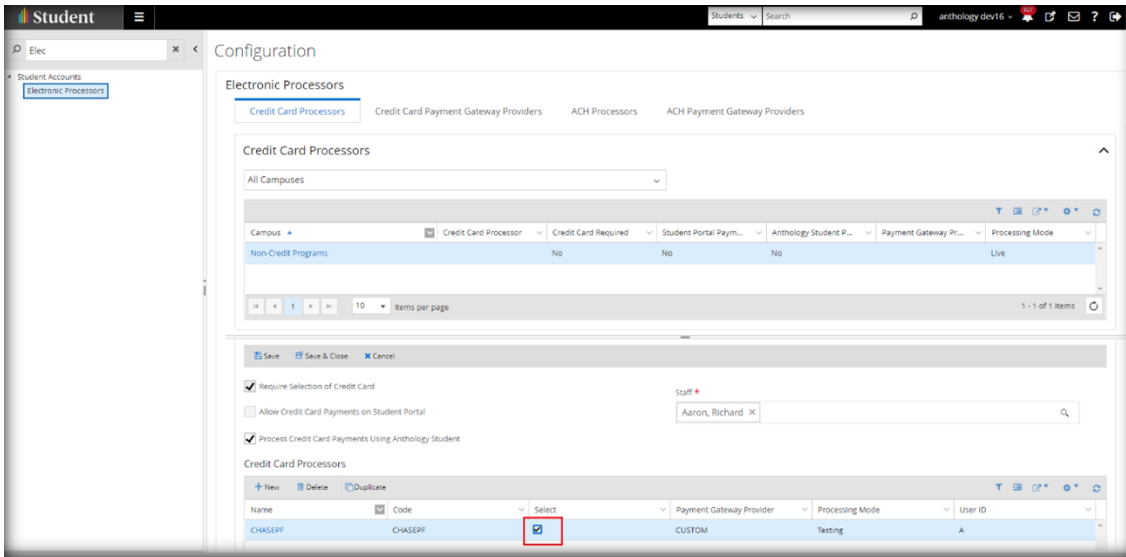
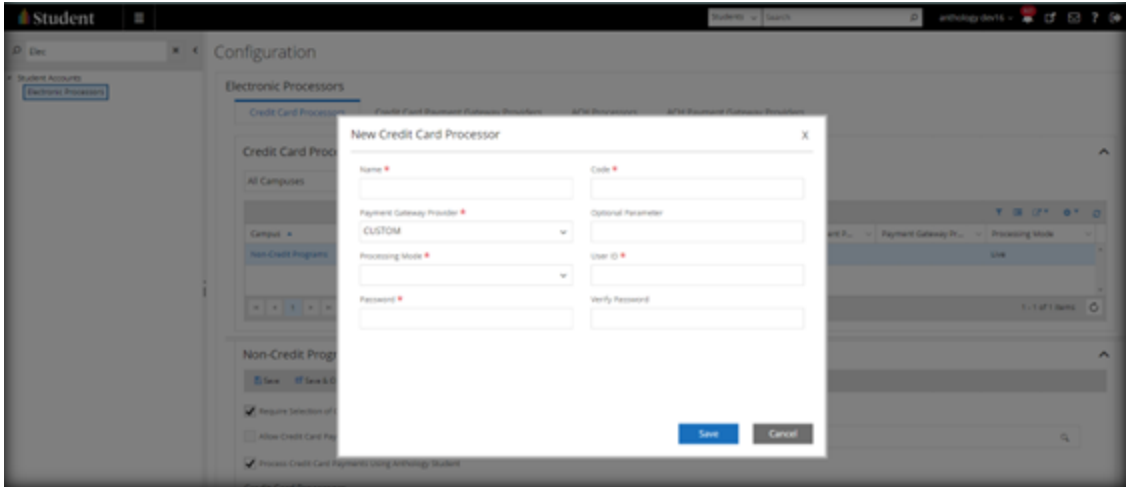
2. Click the **Campus** link to open the editable UI and select the check boxes **Require Selection of Credit Card** and **Process Credit Card Payments Using Anthology Student**.



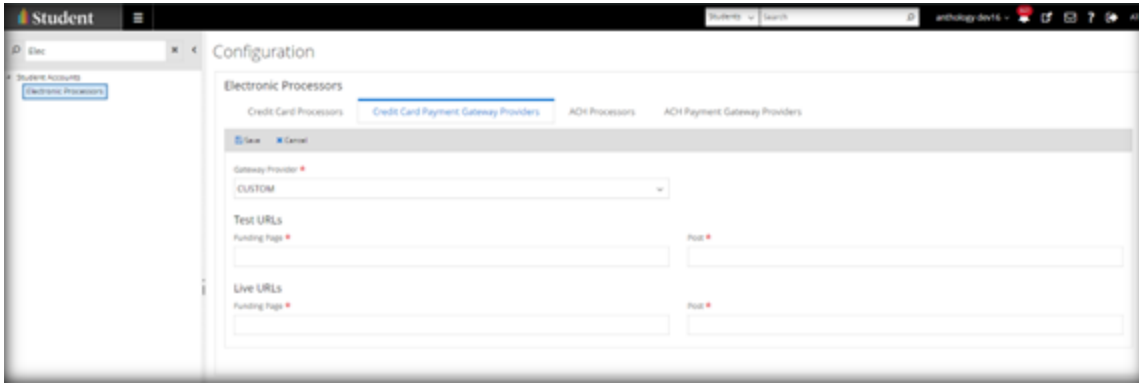
3. Click the **New** button to add the custom provider and select Payment Gateway Provider as **CUSTOM**.

If you don't have a user ID and Password, enter dummy text in the appropriate fields and click Save. The User ID and Password are not currently being used to log in to the funding page. The funding page authentication is handled by the client. Anthology authenticates only the Post URL by providing the API token in the request header. We create the API token from the Azure AD service using the Client ID and

ClientSecret, OAuthTokenUrl, and ResourceUrl. These values are provided by the client. More details are available in the [API Authentication Method](#) section.



4. Once the new payment gateway provider has been added to the grid, select the check box, and click the **Save** button.
5. Go the **Credit Card Payment Gateway Providers** tab and add valid URLs for the custom payment gateway provider. These URLs must be provided by the client.
 - **Funding Page URL** – This URL will redirect the user from Anthology Student to the client’s hosted funding page to add credit cards.
 - **Post URL** – This URL will be used to post payment requests to the payment gateway provider.



Follow the steps above to add an ACH processor if needed.

Test Tool

The test tool URLs for Anthology's internal 600045 environment are as follows:

- Credit Card/ACH Funding Page URL - <https://600045appvm1.int.campusnexus.dev:888/PaymentProvider>
- Credit Card/ACH Post URL - <https://600045appvm1.int.campusnexus.dev:888/Payments/PostPayment>

The test tool can be deployed into other internal environments upon request.

Once the deployment is done, change the URL for the **BaseServiceUrl** key on the web.config file to point to the same environment.

```
<configuration>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="PreserveLoginUrl" value="true" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <add key="BaseServiceUrl" value="https://600045appvm1.int.campusnexus.dev:9500/" />
  </appSettings>
```


API Authentication Method

Anthology authenticates the custom payment gateway APIs by providing the authentication token. We use Client ID, Client Secret, Token URL, and Resource URL to generate the token. We added 4 keys in the syRegistry table for this purpose with empty values by default. These keys should be updated with valid values.

The key names are CustomPaymentProviderClientId, CustomPaymentProviderClientSecret, CustomPaymentProviderOAuthTokenUrl, and CustomPaymentProviderResourceUrl

(When a client registers its application with the authorization server, the authorization server will generate a unique Client ID and a Client Secret for that application. These credentials are typically provided by the OAuth provider (authorization server) as part of the registration process.)

CustomPaymentProviderClientId – This value will be added in the request header for “client_Id”.

CustomPaymentProviderClientSecret - This value will be added in the request header for “client_Secret”.

CustomPaymentProviderResourceUrl - This value will be added in the request header for “resource”.

CustomPaymentProviderOAuthTokenUrl – This is the endpoint that Anthology will call to generate the token.

```
<configuration>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="PreserveLoginUrl" value="true" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <add key="BaseServiceUrl" value="https://600045appvm1.int.campusnexus.dev:9500/" />
  </appSettings>
```

Example: Add a valid RegValue for all keys:

```
UPDATE syRegistry SET RegValue='xxxxxx' WHERE RegKey='CustomPaymentProviderClientId'
```

```
UPDATE syRegistry SET RegValue='xxxxxx' WHERE RegKey='CustomPaymentProviderClientSecret'
```

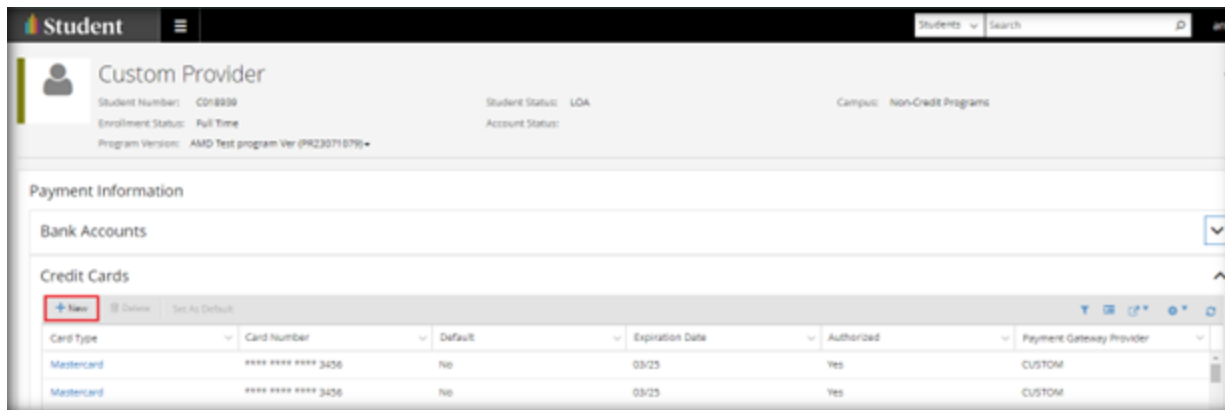
```
UPDATE syRegistry SET RegValue='xxxxxx' WHERE RegKey='CustomPaymentProviderOAuthTokenUrl'
```

```
UPDATE syRegistry SET RegValue='xxxxxx' WHERE RegKey='CustomPaymentProviderResourceUrl'
```

Payment Info

Add New Credit Card

Select the student > Student Accounts > Payment Information tile.



The New button in this UI will redirect to the specified custom payment provider Funding Page URL that is configured under Configuration > Student Accounts > Electronic Processors > Credit Card Payment Gateway Providers.

The userid, studentid, and returnurl will be passed as query strings. The passed URLs have been encoded.

Sample URL:

[https://\[custompaymentproviderurl\]?studentid=59655&userid=123&returnurl=http://siswebclienturl/#/students/59655/paymentInformation](https://[custompaymentproviderurl]?studentid=59655&userid=123&returnurl=http://siswebclienturl/#/students/59655/paymentInformation)

Save data to the SaCC table once the credit card has been successfully added to the payment provider.

Below is the sample URL and payload to save a new credit card using the StudentCreditCard command API:

[https://\[siswebclienturl\]/api/commands/StudentAccounts/StudentCreditCard/SaveNew](https://[siswebclienturl]/api/commands/StudentAccounts/StudentCreditCard/SaveNew)

```

"payload": {
  "isOnlinePayment": true,
  "authorizationNumber": "TST834", //AuthNumber returned from payment provider
  "cardHolderCity": "ADJUNTAS",
  "cardHolderFirstName": "Test",
  "cardHolderLastName": "User",
  "cardHolderName": "Test User",
  "cardHolderPostalCode": "00601",
  "cardHolderState": "FL",
  "cardHolderStreetAddress": "Address1",
  "correlationIdentifier": 288757, //Student ID
  "createdDateTime": "2023/08/31 03:20:40", //Current Date Time
  "creditCardNumber": "V786501VISA4542", //Token from payment provider
  "creditCardTypeId": 1, //creditCardTypeId from saCCType table
  "expirationDate": "2025/03/31 00:00:00",
  "isActive": true,
  "isCreditCardNumberStored": true,
  "isSingleUse": false,
  "isVerified": false,
  "lastFourCreditCardNumber": "3456",
  "lastModifiedDateTime": "2023/08/31 03:20:40", //Current Date Time
  "lastModifiedUserId": 32973,
  "paymentGatewayProvider": "CHASE", //Payment Provider Code
  "studentId": 288757
}

```

Using the following OData URL, the CreditCardTypeId can be fetched from the SaCCType table based on code.

[https://\[siswebclienturl\]/ds/campusnexus/CreditCardTypes?\\$filter=Code eq 'V'](https://[siswebclienturl]/ds/campusnexus/CreditCardTypes?$filter=Code eq 'V')

The sample credit card type codes are listed below.

Credit Card Type	Code
Visa	V
MasterCard	M
Discover	D
Amex	A
American Express	A
Diner's Club	I
JCB	J
PayPath	P
Other	O

If card is added successfully, redirect to the returnUrl provided in the above query string with AddCard=true

Example:

[http://\[siswebclienturl\]/#/students/59655/paymentInformation?AddCard=true](http://[siswebclienturl]/#/students/59655/paymentInformation?AddCard=true)

If any error, then pass the error message in query string as below.

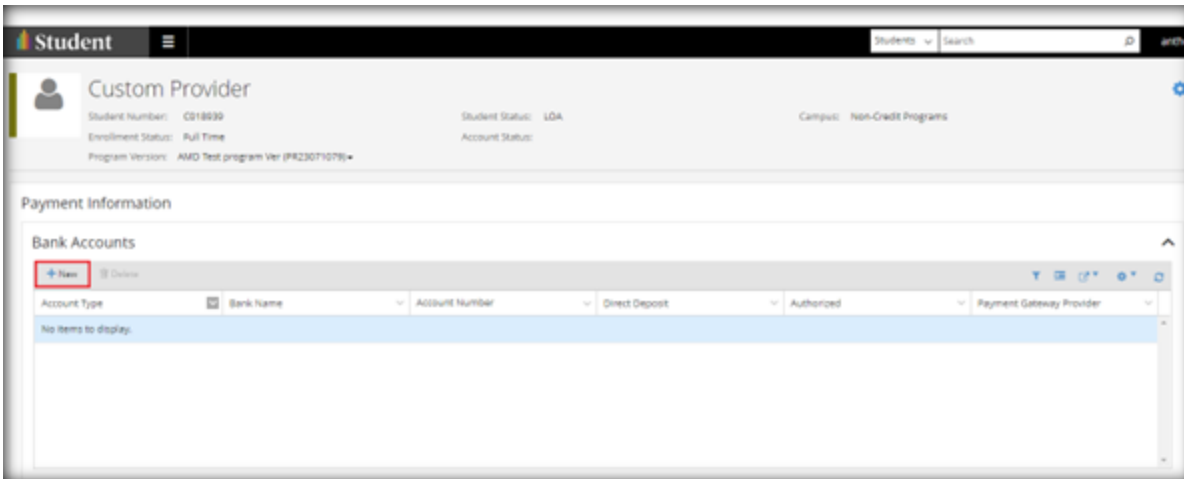
[http://\[siswebclienturl\]/#/students/59655/paymentInformation?errmsg=error](http://[siswebclienturl]/#/students/59655/paymentInformation?errmsg=error)

Cancel URL.

[http://\[siswebclienturl\]/#/students/59655/paymentInformation](http://[siswebclienturl]/#/students/59655/paymentInformation)

Add New Bank Account

Select the student > Student Accounts > Payment Information tile.



The New button in this UI will redirect to the specified custom payment provider Funding Page URL that is configured under Configuration > Student Accounts > Electronic Processors > ACH Payment Gateway Providers. The userid, studentid, and return URL will be passed as query strings. The passed URLs have been encoded.

Sample URL:

[https://\[custompaymentproviderurl\]?studentid=59655&userid=123&returnurl=http://siswebclienturl/#/students/59655/paymentInformation](https://[custompaymentproviderurl]?studentid=59655&userid=123&returnurl=http://siswebclienturl/#/students/59655/paymentInformation)

Save data to the SaStudentBankAccount table once the bank account has been successfully added to the payment provider.

Below is a sample URL and payload to save a bank account using the StudentBankAccount command API.

[https://\[siswebclienturl\]/api/commands/StudentAccounts/StudentBankAccount/SaveNew](https://[siswebclienturl]/api/commands/StudentAccounts/StudentBankAccount/SaveNew)

```

"payload": {
  "accountNumber": "12345678", //Token from payment provider
  "authorizationNumber": null, //AuthNumber returned from payment provider
  "createdDateTime": "2023/09/05 03:33:43", //Current Date Time
  "isDirectDeposit": false,
  "lastModifiedDateTime": "2023/09/05 03:33:43", //Current Date Time
  "lastModifiedUserId": 32973,
  "studentId": 288757,
  "paymentGatewayProvider": "CHASE", //payment gate way provider code. E.g ACI, PayPal, TouchNet, CHASE etc
  "isVerified": false,
  "bankName": "State Bank",
  "correlationIdentifier": 288757, //studentId
  "isActive": true,
  "isOnlinePayment": true,
  "LastFourNumber": "2345", //Last four account number
  "isPrimaryPayment": false,
  "isSingleUse": false,
  "routingNumber": "072403004",
  "accountType": "S" // "S" or "C"
}

```

The valid values for the account type will be one of codes from the list below.

Account Type Name	Code
Savings	S
Personal Savings	S
Business Savings	S
Checking	C
Personal checking	C
Business checking	C

If Bank Account is added successfully, redirect to the URL passed in query string with:

<http://siswebclienturl/#/students/49633/paymentInformation?AddAccount=true>

If any error occurs, pass the error message in query string

<http://siswebclienturl/#/students/59655/paymentInformation?errmsg=error>

Cancel URL:

<http://siswebclienturl/#/students/59655/paymentInformation>

Use the command below to read any previously saved bank accounts.

Example:

<http://siswebclienturl/api/commands/StudentAccounts/StudentBankAccount/Get>

```

{
  "payload": {
    "id": "4"
  }
}

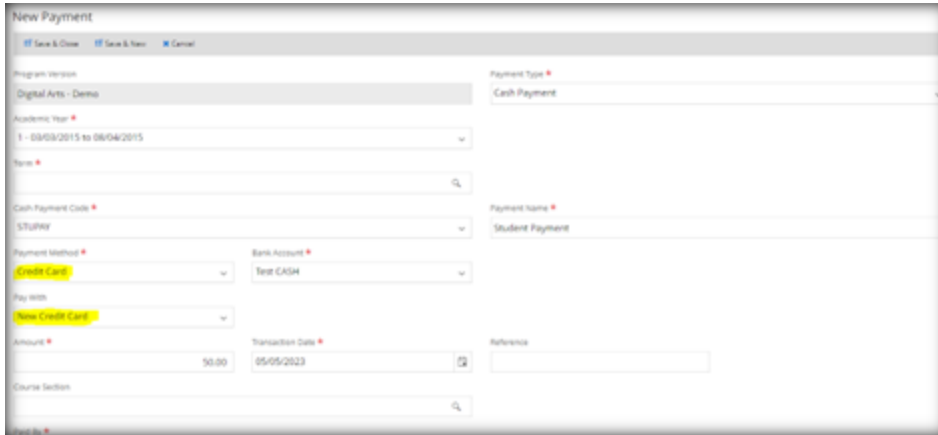
```

The decrypted bank account number will be included in the results with other details.

Ledger Card

Make Payment With New Credit Card

The Anthology Student web app will redirect to the configured custom payment gateway provider Funding URL when the user selects the Payment Method as “Credit Card” and Pay With as “New Credit Card” on this UI.

A screenshot of a web application form titled "New Payment". The form is divided into several sections. At the top left, there are buttons for "Save & Close", "Save & New", and "Cancel". The form fields include: "Program Version" (Digital Arts - Demo), "Academic Year" (1 - 08/08/2015 to 08/04/2015), "Payment Type" (Cash Payment), "Cash Payment Code" (STUPAY), "Payment Method" (Credit Card), "Bank Account" (Test CASH), "Payment Name" (Student Payment), "Pay With" (New Credit Card), "Amount" (50.00), "Transaction Date" (05/05/2023), and "Reference". The "Payment Method" and "Pay With" fields are highlighted in yellow.

Sample URL:

[https://\[chaseurl\]?studentid=59655&userid=2&userpreferenceguid=690746db-d1e3-4509-8cc9-e1e36837&returnurl=http://siswebclienturl/#/students/59655/ledgercard/transactions](https://[chaseurl]?studentid=59655&userid=2&userpreferenceguid=690746db-d1e3-4509-8cc9-e1e36837&returnurl=http://siswebclienturl/#/students/59655/ledgercard/transactions)

The PostAccountTransactionPayment payload will be saved in the core.UIUserPreference table before redirecting for future use.

Once card is added and payment is processed successfully in the payment provider, you need to save the credit card with the details shown below.

<http://localhost:90/api/commands/StudentAccounts/StudentCreditCard/SaveNew>

```

"payload": {
  "isOnlinePayment": true,
  "authorizationNumber": "TST834", //AuthNumber returned from payment provider
  "cardHolderCity": "ADJUNTAS",
  "cardHolderFirstName": "Test",
  "cardHolderLastName": "User",
  "cardHolderName": "Test User",
  "cardHolderPostalCode": "00601",
  "cardHolderState": "FL",
  "cardHolderStreetAddress": "Address1",
  "correlationIdentifier": 288757, //Student ID
  "createdDateTime": "2023/08/31 03:20:40", //Current Date Time
  "creditCardNumber": "V786501VISA4542", //Token from payment provider
  "creditCardTypeId": 1, //creditCardTypeId from saCCType table
  "expirationDate": "2025/03/31 00:00:00",
  "isActive": true,
  "isCreditCardNumberStored": true,
  "isSingleUse": false,
  "IsVerified": false,
  "lastFourCreditCardNumber": "3456",
  "lastModifiedDateTime": "2023/08/31 03:20:40", //Current Date Time
  "lastModifiedUserId": 32973,
  "paymentGatewayProvider": "CHASE", //Payment Provider Code
  "studentId": 288757
}

```

After card is saved, post the transaction using the PostAccountTransactionPayment command API.

(api/commands/StudentAccounts/StudentAccountTransaction/PostAccountTransactionPayment)

The payload for this can be fetched using the userpreferenceguid passed in the query string.

This data is saved in the core.UIUserPreference table. The value column contains the payload.

Example

```
select [value] from core.UIUserPreference where [Key]='690746db-d1e3-4509-8cc9-e1e36837'
```

Sample OData to fetch records from UIUserPreference table would be as follows.

[http://siswebclienturl/ds/views/UserPreferences/CampusNexus.GetUserPreferenceItemByKey\(key='690746db-d1e3-4509-8cc9-e1e36837'\)](http://siswebclienturl/ds/views/UserPreferences/CampusNexus.GetUserPreferenceItemByKey(key='690746db-d1e3-4509-8cc9-e1e36837'))

The value for the Key will be passed as query string parameter (userpreferenceguid)/

Once this value is fetched, modify the following property values in the request before posting.

Example:

```

var payload = value;
payload.StudentCreditCardId = [new credit card id added]
payload.IsNewPaymentSuccessful = true;
payload.IsStudentCreditCardAuthorizeRequired = true;
payload.AuthorizationNumber = [Auth Number from payment provider]

```

Once payment is posted, delete the record from the core.UIUserPreference table.

The Id for the Delete command can be fetched using the following OData query.

[http://siswebclienturl/ds/views/UserPreferences/CampusNexus.GetUserPreferenceItemByKey\(key='690746db-d1e3-4509-8cc9-e1e36837'\)](http://siswebclienturl/ds/views/UserPreferences/CampusNexus.GetUserPreferenceItemByKey(key='690746db-d1e3-4509-8cc9-e1e36837'))

The following URL can be used to delete rows from the UIUserPreference table. For example, pass the Id (3695) that is returned from above URL to the Delete command and use the DELETE verb.

[https://siswebclienturl/ds/views/UserPreferences\(3695\)](https://siswebclienturl/ds/views/UserPreferences(3695))

Success URL:

<http://siswebclienturl/#/students/59655/ledgercard/transactions?paymentsuccess=true>

Additionally, Anthology will pass a few more query string parameters based on business functionality and the same should be returned to us to with the return URL (isFromStudentAccountTransaction, isFromProcessPostPayment, IsSaveAndNew, IsApplyPayment, IsPrint, etc.).

For example, we will redirect to the end point with the following query string parameters.

[https://\[chaseurl\]?studentid=59655&userid=2&userpreferenceguid=690746db-d1e3-4509-8cc9-e1e36837&isFromStudentAccountTransaction=true&isFromProcessPostPayment=false&IsSaveAndNew=false&IsApplyPayment=true&IsPrint=false&returnurl=http://siswebclienturl/#/students/59655/ledgercard/transactions](https://[chaseurl]?studentid=59655&userid=2&userpreferenceguid=690746db-d1e3-4509-8cc9-e1e36837&isFromStudentAccountTransaction=true&isFromProcessPostPayment=false&IsSaveAndNew=false&IsApplyPayment=true&IsPrint=false&returnurl=http://siswebclienturl/#/students/59655/ledgercard/transactions)

Anthology expects the return URL as: [http://siswebclienturl/#/students/59655/ledgercard/transactions?paymentsuccess=true&isFromStu-](http://siswebclienturl/#/students/59655/ledgercard/transactions?paymentsuccess=true&isFromStudentAccountTransaction=true&isFromProcessPostPayment=false&IsSaveAndNew=false&IsApplyPayment=true&IsPrint=false)

[dentAccountTransaction=true&isFromProcessPostPayment=false&IsSaveAndNew=false&IsApplyPayment=true&IsPrint=false](http://siswebclienturl/#/students/59655/ledgercard/transactions?paymentsuccess=true&isFromStudentAccountTransaction=true&isFromProcessPostPayment=false&IsSaveAndNew=false&IsApplyPayment=true&IsPrint=false)

These are required in the UI to load the proper controls.

Cancel URL:

<http://siswebclienturl/#/students/59655/ledgercard/transactions>

Error URL

<http://siswebclienturl/#/students/59655/ledgercard/transactions?errmsg=error>

Make Payment Using Saved Credit Card/Student Bank Account

On this screen, when the user selects the Payment Method as Credit Card/ACH and selects the existing Credit Card/Student Bank Account, the web client service will call the API endpoint (Post URL) that is configured under Configuration → Student Accounts → Electronic Processors (Credit Card Payment Gateway Providers / ACH Payment Gateway Providers). The Funding Page URL is used for redirection and the Post URL is used for posting the payment to the payment gateway provider.

Anthology Student sends a request to post a payment to the custom payment gateway provider. The following parameters are sent in the request.

```

Credit Card Request
{
  "IsAutoPay": false,
  "CampusCode": "TRAD",
  "Amount": 100,
  "CreditCardType": "VISA",
  "VaultId": "DP3ECH367", //Reference number
  "CreatedBy": "test user",
  "Payor": {
    "StudentId": "137"
  }
}

Bank Account Request
{
  "IsAutoPay": false,
  "CampusCode": "TRAD",
  "Amount": 100,
  "AccountType": "Savings",
  "VaultId": "DP3ECH341", //Reference number
  "CreatedBy": "test user",
  "Payor": {
    "StudentID": "1133"
  }
}

```

Note: We might have to update the values in the Anthology Student database if the expected CreditCardType/AccountType in the request does not match the values in the Anthology Student database. For example, if the CreditCardType expected in the request is "Visa", but the value in the saCCType table is "V", we need to update the code in the saCCType table to match the request.

SELECT * from SaCCType

SaCCTypeID	Code	Descrip	Active	SystemCode	UserID	DateAdded	DateLastMod	ts	SyCampusGpID
1	M	Mastercard	1	1	1	2000-12-18 18:09:36.930	2009-02-12 14:16:10.000	0x000000000000AF294	2
2	A	AMEX	1	1	1	2000-12-18 18:09:37.000	2009-02-12 14:15:26.000	0x000000000000AF274	3
3	V	VISA	1	1	1	2000-12-18 18:09:37.070	2013-10-22 11:57:11.000	0x00000003F9FEDC58	4
4	D	Discover	1	1	1	2000-12-18 18:09:37.153	2010-07-15 09:42:58.000	0x000000000008E7E98	5
5	I	Diners Club	0	1	1	2000-12-18 18:09:37.223	2000-12-18 18:09:37.223	0x000000000000AF27F	6
6	C	Carte Blanche	0	1	1	2000-12-18 18:09:37.293	2000-12-18 18:09:37.293	0x000000000000AF27E	7
7	J	JCB	1	1	1	2018-10-29 00:41:06.270	2018-10-29 00:41:06.270	0x000000035996BFED	1
8	O	Other	1	1	1	2018-10-29 00:41:06.270	2018-10-29 00:41:06.270	0x000000035996BFEE	1

UPDATE SaCCType SET Code = 'VISA' WHERE Code = 'V'

If the AccountType does not match, the Code in the SyCode table needs to be updated to match the expected values in the request.

SELECT * FROM SyCode WHERE TableName = 'saStudentBankAccount' and ColumnName = 'AccountTypeCode'

	TableName	ColumnName	Code	Descrip	SyCodeID	ts	TooltipDesc
1	SaStudentBankAccount	AccountTypeCode	C	Checking	3590	0x00000003F9E4B5A3	NULL
2	SaStudentBankAccount	AccountTypeCode	S	Saving	3591	0x00000003F9E4B5A4	NULL

UPDATE syCode SET Code = 'Checking' WHERE Code = 'C'
 AND TableName = 'saStudentBankAccount' AND ColumnName = 'AccountTypeCode'

In addition, make sure to update the AccountTypeCode column with the same value when the records are inserted into the saStudentBankAccount table.

Anthology expects the following properties in the response when calling the API to post payment.

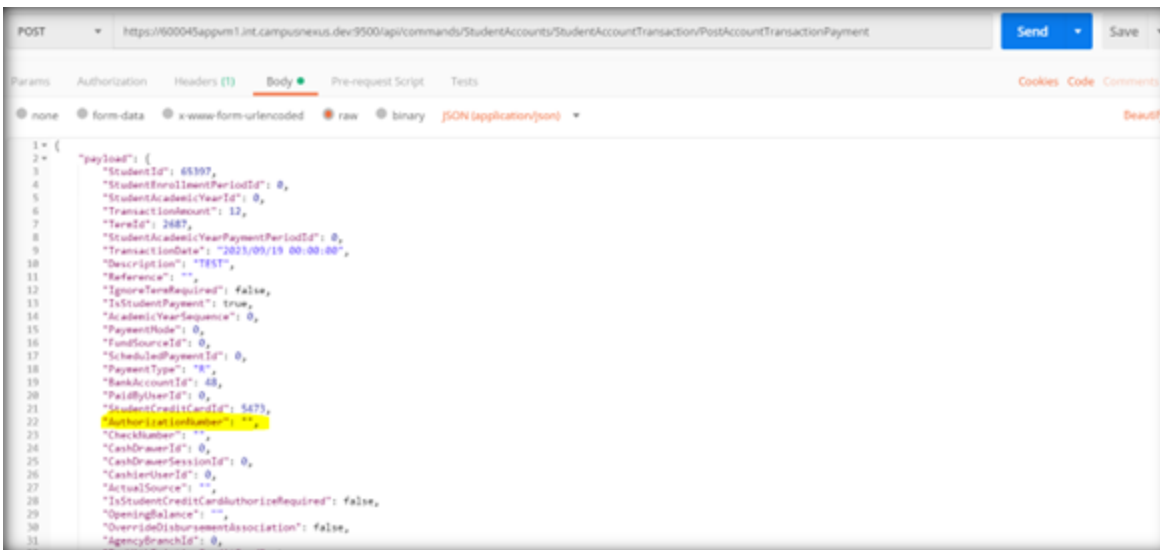
- Success – true/false. Indicates whether the transaction is failure or success.
- AuthCode – [Authorization number returned from payment gateway provider]
- Token – [Token returned from payment gateway provider]
- Messages – If failure, then this should contain the error message.
- Error Type – Provide any other information related to the error, if available, or Empty.

```
{
  "success": true,
  "errorType": "",
  "authCode": "TEST574",
  "token": "TESTVISA4242",
  "messages": [
    "Approved",
  ]
}
```

Post Credit Card/ACH Payments for Reference Only

Clients can use the command API (api/-commands/StudentAccounts/StudentAccountTransaction/PostAccountTransactionPayment) to post Credit Card/ACH transactions directly to the ledger card without sending them to payment gateway provider to be processed. This can be done by populating the AuthorizationNumber in the payload. So, if the Authorization Number in the payload is blank, the transaction will be sent to the custom payment gateway provider; but if it has a value, the request will not be sent to the custom payment gateway provider and will be posted to the ledger card only.

For example, the following request will be sent to custom payment gateway provider.



The following request will not be sent to the custom payment gateway provider because the AuthorizationNumber is available.

```
POST https://60045appw1.int.campusnexu.dev:9500/api/commands/StudentAccounts/StudentAccountTransaction/PostAccountTransactionPayment
Body
JSON (application/json)
{
  "payload": [
    {
      "StudentId": 65397,
      "StudentEnrollmentPeriodId": 0,
      "StudentAcademicYearId": 0,
      "TransactionAmount": 12,
      "TermId": 2087,
      "StudentAcademicYearPaymentPeriodId": 0,
      "TransactionDate": "2022/09/19 00:00:00",
      "Description": "TEST",
      "Reference": "",
      "IgnoreItemRequired": false,
      "IsStudentPayment": true,
      "AcademicYearSequence": 0,
      "PaymentMode": 0,
      "FundSourceId": 0,
      "SchedulePaymentId": 0,
      "PaymentType": "R",
      "BankAccountId": 48,
      "PaidUserId": 0,
      "StudentCreditCardId": 5433,
      "AuthorizationNumber": "A3440576",
      "CheckNumber": "",
      "CashDrawerId": 0,
      "CashDrawerSessionId": 0,
      "CashierUserId": 0,
      "ActualSource": "",
      "IsStudentCreditCardAuthorizeRequired": false,
      "OpeningBalance": "",
      "OverrideDisbursementAssociation": false,
      "AgencyBranchId": 0
    }
  ]
}
```