-: Nab velocity wrapper API documentation:-

We have created test account:

ServiceID: 8D9DE00001

IdentityToken:



MerchantProfileId: EPX Retail Test

ApplicationProfileId: 71228

1-First deploy the nabwrapper.war on your environment.

2- Import the collections of Inovio and Nabvelocity actual api.

3- Import all the environment variable where we have added base url.

4- To test the Nab velocity actual API need to create test account as we have created as below.

**ServiceID**: 8D9DE00001

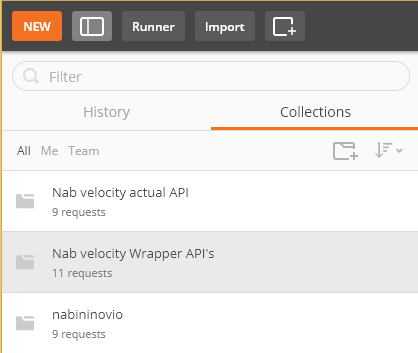
**IdentityToken**:



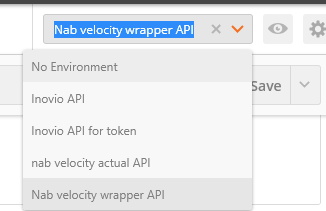
**MerchantProfileId**: EPX Retail Test

**ApplicationProfileId**: 71228

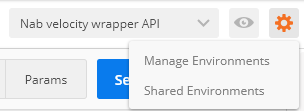
5- In postman now you are able to see the collections in left hand like below after import the collections.



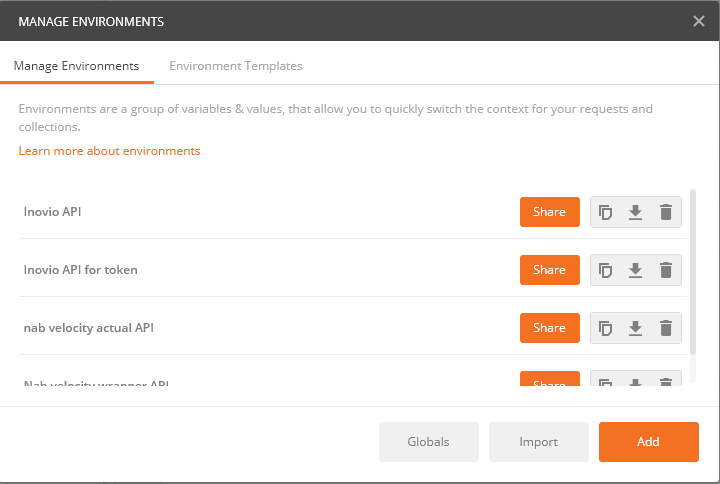
6- In postman now you are able to see all the environment variable after import the environment variable.



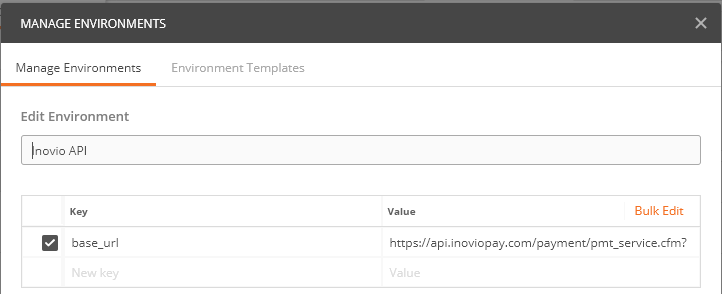
7- To edit the environment variable click on setting icon and click on Manage environment variable to edit he environment variable.



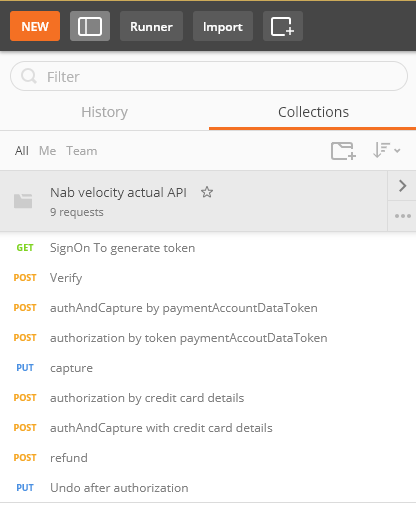
Double click on any environment to edit.



Like we just clicked Inovio API and here you can edit as per requirement.

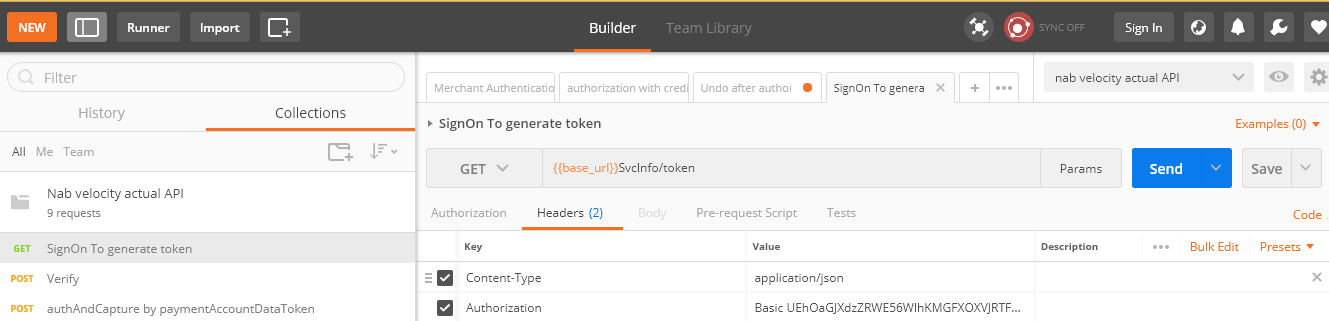


8- Now Click on Nab velocity actual API which is in left hand side with all the API’s.



9- First method is used to verify the merchant information using token id and for that we need to pass token id in header section which we got after register on Nab velocity site.

1. Once you select SignOn to generate token it will show like below.



1. Select environment according to collection like for Nab velocity select nab velocity actual api.
2. First convert token value with colon using base64. For example you can use online site <http://www.utilities-online.info/base64/#.XI9gJCIzbIX>
3. Token id as below we got after register on the site.
4. :

**Note**: we have added colon (**:)** after token**.**

You will got new value to pass in Header section of the value of authorization after convert into base64.



9- Now step 8 value will use in authorization in header as below with basic and select environment Nab velocity actual api.

Authorization Basic 

10- Now Hit the click button to receive new value like below.



11- Now we will convert the generated value from step 10 with colon:

:

**Note**: We have added colon (:)

Value is generated from base64.



12- Generated value of base64 from step 11 will use to set authorization value in header for another API.

13- Click on verify method and update the header information with generated base 64 value

Authorization : Basic 

**Note:** Basic always will be constant only base 64 value will be change.

13- Now hit the send button to get the response of verify method and the value will be like below.

<BankcardTransactionResponsePro xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard/Pro" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

<Status xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">Successful</Status>

<StatusCode xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">00</StatusCode>

<StatusMessage xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">NO MATCH</StatusMessage>

<TransactionId xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">435C9C139B5F44DCB0033B2790401BB2</TransactionId>

<OriginatorTransactionId xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">28579</OriginatorTransactionId>

<ServiceTransactionId xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">09KGN50MHDZF0VPH33D</ServiceTransactionId>

<ServiceTransactionDateTime xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">

<Date>2019-03-18</Date>

<Time>12:46:28.000</Time>

<TimeZone>-06:00</TimeZone>

</ServiceTransactionDateTime>

<Addendum xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">

<Unmanaged>

<Any xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

<a:string>ID:8034695</a:string>

</Any>

</Unmanaged>

</Addendum>

<CaptureState xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">CannotCapture</CaptureState>

<TransactionState xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">Verified</TransactionState>

<IsAcknowledged xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions">false</IsAcknowledged>

<Reference xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions"/>

<Amount xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">0</Amount>

<CardType xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">Visa</CardType>

<FeeAmount xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">0</FeeAmount>

<ApprovalCode xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard"/>

<AVSResult xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">

<ActualResult>N</ActualResult>

<AddressResult>NoMatch</AddressResult>

<CountryResult>NotSet</CountryResult>

<StateResult>NotSet</StateResult>

<PostalCodeResult>NoMatch</PostalCodeResult>

<PhoneResult>NotSet</PhoneResult>

<CardholderNameResult>NotSet</CardholderNameResult>

<CityResult>NotSet</CityResult>

</AVSResult>

<BatchId xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard"/>

<CVResult xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">Match</CVResult>

<CardLevel xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard"/>

<DowngradeCode xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard"/>

<MaskedPAN xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">4XXXXXXXXXXXX111</MaskedPAN>

<PaymentAccountDataToken xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">435c9c13-9b5f-44dc-b003-3b2790401bb28612796c-7a50-4ef8-8620-270dfeb00d1e</PaymentAccountDataToken>

<RetrievalReferenceNumber xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard"/>

<Resubmit xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">Unspecified</Resubmit>

<SettlementDate xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">0001-01-01T00:00:00</SettlementDate>

<FinalBalance xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">0</FinalBalance>

<OrderId xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">7661</OrderId>

<CashBackAmount xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">0.00</CashBackAmount>

<PrepaidCard xmlns="http://schemas.ipcommerce.com/CWS/v2.0/Transactions/Bankcard">NotSet</PrepaidCard>

<AdviceResponse>NotSet</AdviceResponse>

<CommercialCardResponse>NotSet</CommercialCardResponse>

<ReturnedACI>NotSet</ReturnedACI>

</BankcardTransactionResponsePro>

**Wrapper API Explanation:**

Note: we followed the exact request and response of Nab velocity on Wrapper API.

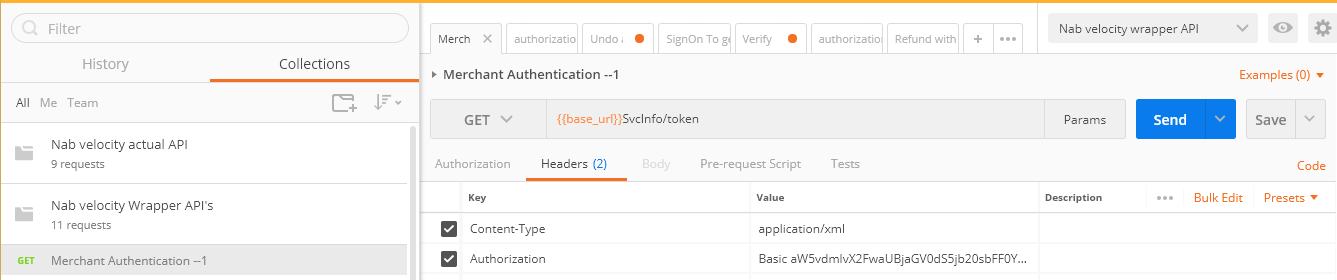
1. **Merchant Authentication**: Use to verify the merchant details
2. How to use: To use merchant authorization we converted in base64 value of Inovio API as follow.

Base64\_encoded(“Username,password,site\_id:”)=aW5vdmlvX2FwaUBjaGV0dS5jb20sbFF0YklibjIwTURNV0NOU05McEQsMjI4OTU6

1. Now will use converted value of base64 in header section with Basic value. So the value will be in header section.

Authorization = Basic aW5vdmlvX2FwaUBjaGV0dS5jb20sbFF0YklibjIwTURNV0NOU05McEQsMjI4OTU6

1. First select collection Nab velocity Wrapper API->select environment Nab velocity Wrapper API.
2. Click on Merchant authenticate and set header value as below.



**Note**: when you import the collection you will already get the value in header section.

1. Now click on postman **send button** to get the result.
2. **Generate card token:** It is use to generate card token number **paymentAccountDataToken.** Using paymentAccountDataToken you can make the payment without card details.

**Note: paymentAccountDataToken** will use only single time to make the payment.

1. **Authorization by Token-** It is use to make payment using paymentAccountDataToken.
2. **Auth and capture by token-** It is use to make payment using paymentAccountDataToken.
3. **Authorization-** It is use to make payment with authorization and after making authorization you can use undo the payment and capture the payment.
4. **Capture:** It is use to capture the payment after authorization using transaction-Id.
5. **Undo after authorization:** It is use to make the void or undo transaction after authorization using transaction-Id.
6. **Refund-** It is uses to refund the transaction after done the capture or auth and capture using transaction-Id.
7. **Auth and capture with credit card details-** It is use to make payment using credit card details.
8. **Authorize with credit card details:** It is use to make the payment using credit card details.