



Proceso de Emisión de Tarjetas Innominadas – Capa Middleware Manual del Proceso

V.1.1.3.

Mayo de 2023

Content table

Change history	3
Executive brief.....	4
Property rights	5
Cards API Overview	6
Proceso de Emisión de Tarjetas Innominadas	7
Estructura de Datos.....	7
Repositorio del proyecto.....	9
Configuración de api-cards-multitenant-core-innominate-issuance.....	10
Solicitud de Tarjetas Innominadas	11
Descripción.....	11
Esquema Pre Autorizador / Full Autorizador	17
Curl	18
Request	18
Response	18
Carga de Tarjetas Innominadas.....	21
Descripción.....	23
Reintentos	29
Ejemplos de carga de archivos.....	30
Caso 1	30
Caso 2.....	30
Caso 3.....	31
Carga Manual de Tarjetas	31
Curl.....	32
Consulta de Procesos	35
Curl.....	35
Configuraciones	36

Change history

V.	Date	Description	Elab. por:	Rev. por:	Aprob. por:
0	30/05/2023	Document creation	Duilio Hernández	Eliana Chaverra	

Executive brief

NovoPayment is a technological services company that allows financial institutions, payment networks, telecommunication companies, retail chains, government entities and diverse enterprises to connect to the complexity of the financial ecosystem in an easy, efficient and cost-effective way.

Funded in 2014, NovoPayment offers highly integrated and interconnected solutions with cloud-based technology, enabling new payment ecosystems that comply with increased security requirements and standards mandated by the industry.

Differentiated by its firsthand institutional knowledge in banking operations, payment methods and Fintech sub-sectors, the company processes quick, safe and reliable electronic payments through a scalable, interoperable bank-grade platform under a Platform-as-a-Service (PaaS) model, enabling diverse digital and electronic payment schemes, stored financial value programs, debit banking and mobile wallets to facilitate the diverse financial management needs of companies and consumers.

This manual constitutes an essential instrument in the static code analysis process of the different solutions and products developed by NovoPayment.

Property rights

This document is confidential and contains information about NovoPayment, its subsidiaries, and/or commercial allies, which is for private and exclusive use and distribution of NovoPayment and constitutes intellectual property that represents an asset for the company.

By receiving this document, every receptor/reader accepts that the information contained in this document has that character, acquiring the obligation to not divulge the contents of it, keep the confidentiality of it, making sure that the subsidiaries, representatives and/or any other related third parties do not divulge and maintain the confidentiality of the document.

Additionally, the contained information of this document cannot be used or duplicated in any level (partial or full) without previous consent by writing of NovoPayment nor cannot be used to the detriment of NovoPayment or any of its subsidiaries and/or commercial allies.

As a part of this obligation, every receptor and/or reader compromises to take reasonable measures to guarantee that the information, or any part of it, cannot be used, revealed or divulged by the same receptor, reader and/or any unauthorized person.

This document, including all presented materials by or in name of NovoPayment, cannot be reproduced, published, electronically transmitted or any other way, nor be modified without the previous consent by writing of NovoPayment.

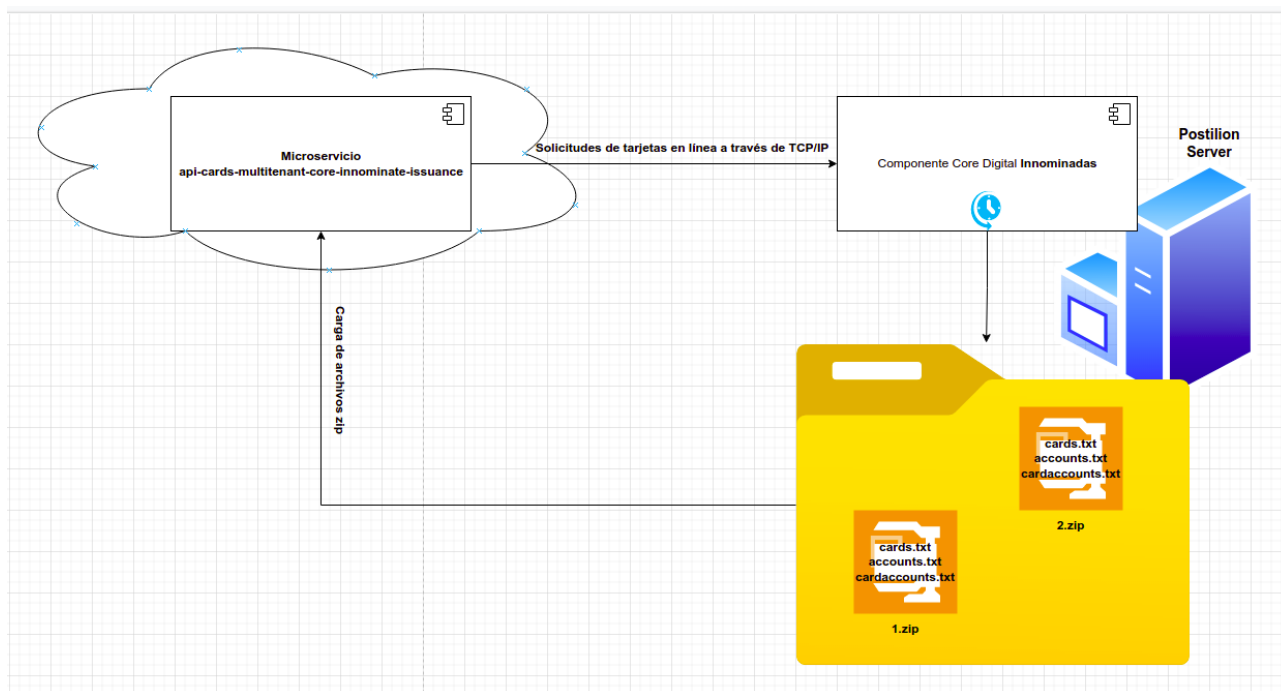
Cards API Overview

CARDS is the API in charge of the creation and enablement of a payment card or financial instrument that will allow your customer to execute transactions in physical stores (POS), ecommerce, pay merchants, use ATMs, among others. You can generate physical or virtual cards associated with an account, with a vast combination of characteristics, like close loop and open loop transactions, restrictions for domestic, international, cross border transactions, specific currency, etc. Thus, allowing you to deliver your unique solution.

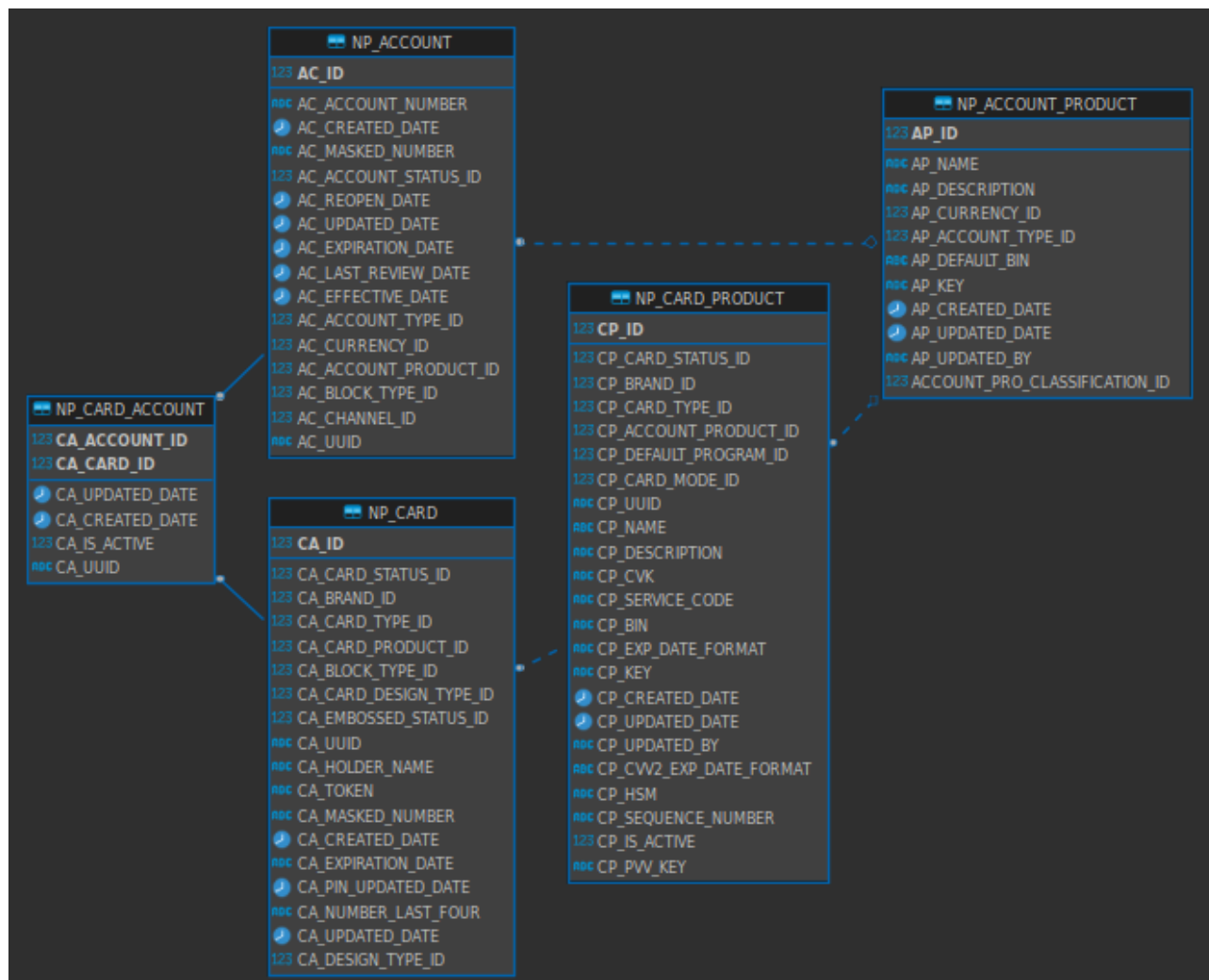
NovoPayment is a worldwide leader in the card enablement process and provides the most advanced security. This experience and capabilities have given NovoPayment the endorsement of the biggest card networks like Visa and Mastercard among world-class PCI compliance.

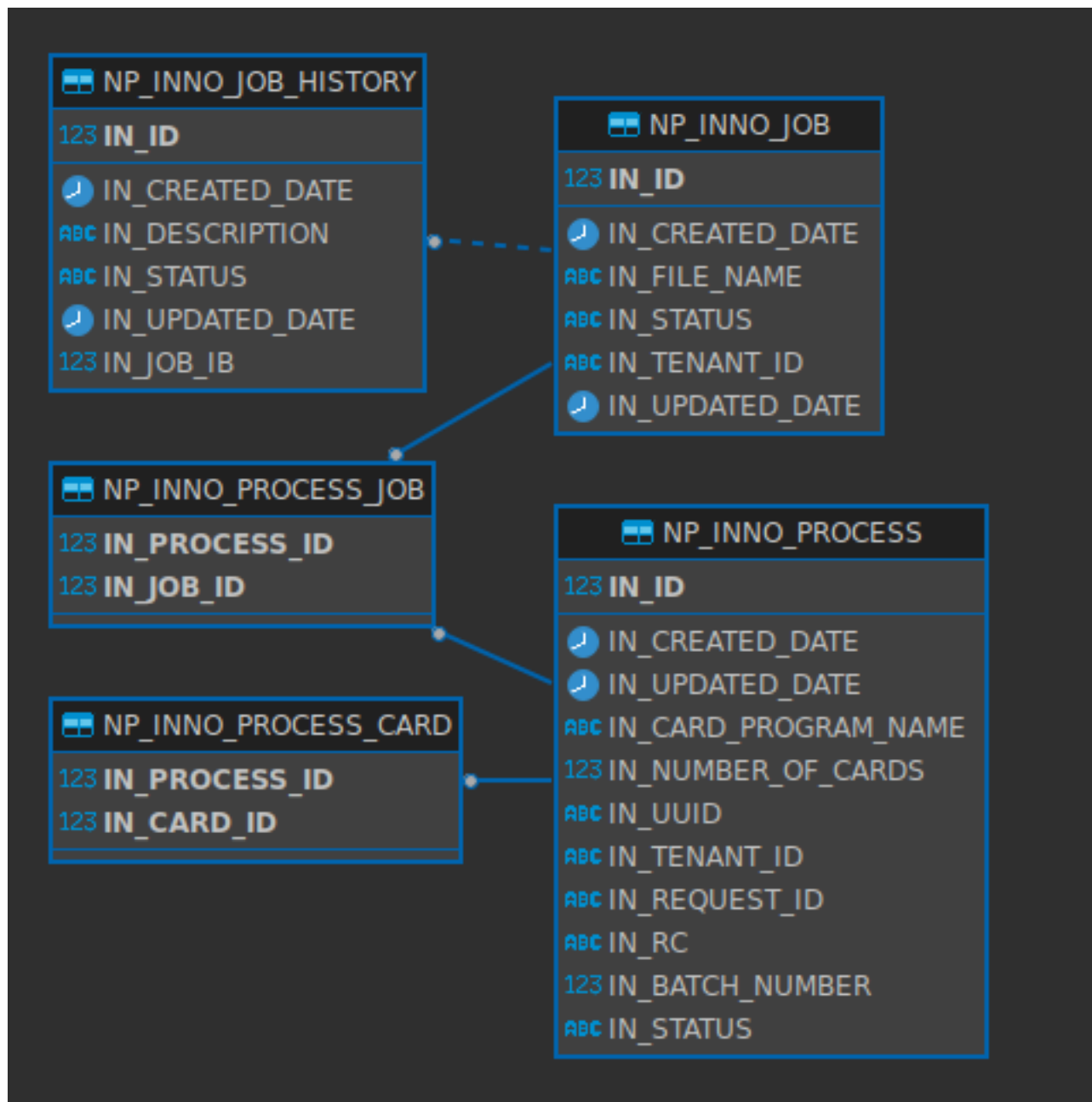
Complementing the card issuing services, NovoPayment offers processing services that work with credit, debit and prepaid cards, giving end to end solutions.

Proceso de Emisión de Tarjetas Innominadas



Estructura de Datos





Repositorio del proyecto

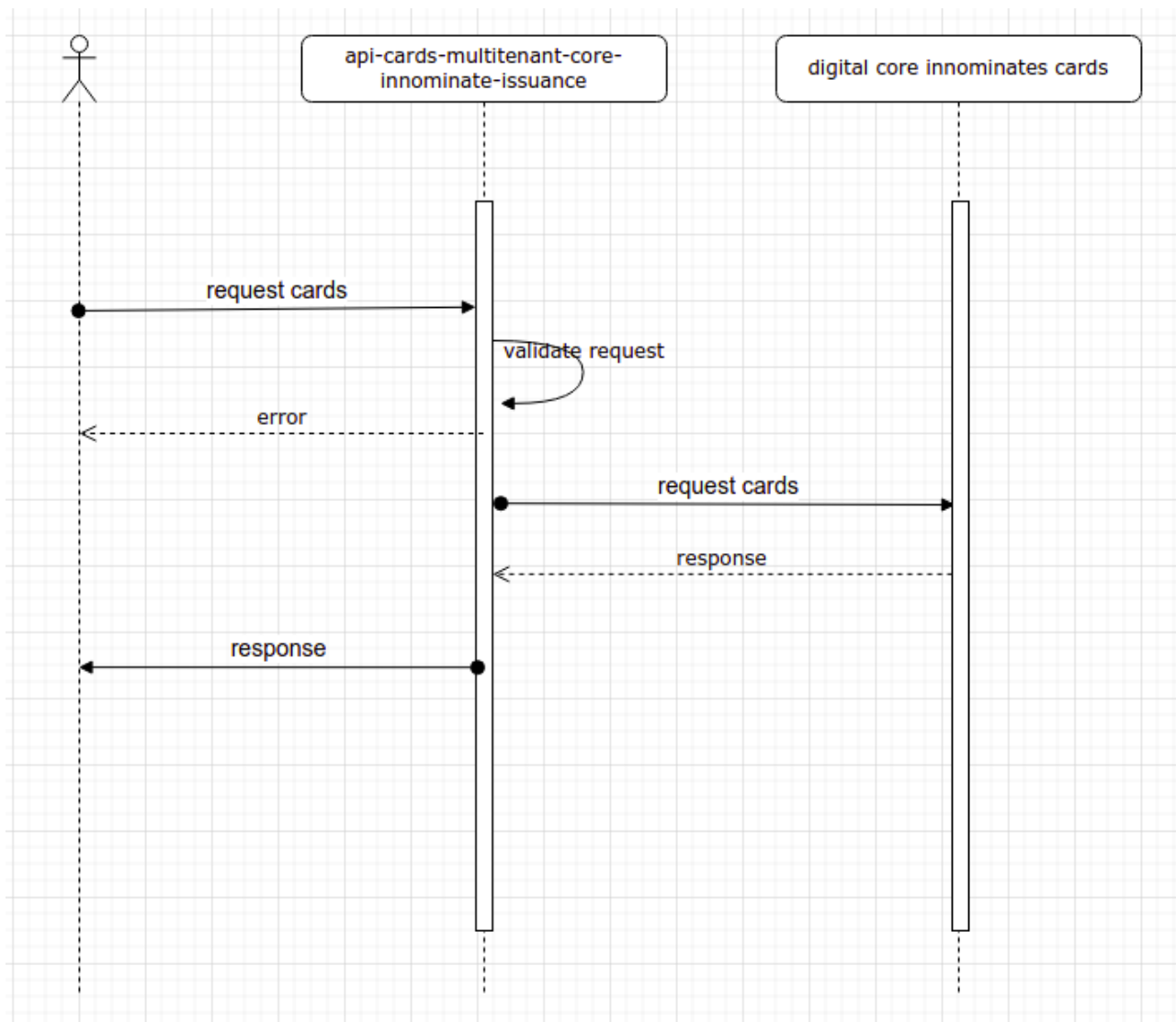
<https://code.novopayment.com/novopayment/api-cards-multitenant-core-innominate-issuance>

Configuración de api-cards-multitenant-core-innominate-issuance

```
1 {  
2   "certificates": [],  
3   "params": [  
4     {  
5       "name": "innominate.card.process.server", 1  
6       "value": "SRVPOSTINT2.novopayment.dom"  
7     },  
8     {  
9       "name": "innominate.card.process.server.port", 2  
10      "value": "9048"  
11    },  
12    {  
13      "name": "innominate.card.process.dir.source", 3  
14      "value": "/postilion-postcard-files/EbanxPeru/PartialCardAccountExtract_Standard3"  
15    },  
16    {  
17      "name": "innominate.card.process.issuer", 4  
18      "value": "06"  
19    },  
20    {  
21      "name": "innominate.card.process.max.cards.request.per.day", 5  
22      "value": "5000"  
23    }  
24  ]  
25 }
```

1. **"innominate.card.process.server"**: Nombre/IP del servidor de Postilion
2. **"innominate.card.process.server.port"**: Puerto del servidor de Postilion
3. **"innominate.card.process.dir.source"**: Directorio donde el microservicio irá a buscar los archivos
4. **"innominate.card.process.issuer"**: Emisor
5. **"innominate.card.process.max.cards.request.per.day"**: Cantidad máxima de tarjetas que se pueden solicitar por día

Solicitud de Tarjetas Innominadas



Descripción

- El microservicio recibe una solicitud de tarjetas a través de un endpoint HTTP. Se valida la solicitud y en caso de existir algún error no se procesa dicha solicitud y se registra el proceso con estado **FAILURE**.
- Si la solicitud es correcta la misma es enviada a un componente del Core Digital (**IssuingCardsAcq**) a través del protocolo **TCP/IP**.

Generalidades

La aplicación **IssuingCardsAcq** de Postilion es una interfaz para nodo Source de tipo Terminal Application. En la actualidad se soporta la siguiente funcionalidad:

- Manejo de mensajería JSON en la comunicación con la entidad remota.
- Inserción de lotes de tarjetas innominadas en la Base de Datos de postcard.
- Notificación transaccional de sólo almacenamiento en Transaction Manager.

Solicitud de generación de tarjetas innominadas

IssuingCardsAcq recibe un **mensaje** de petición con una estructura determinada desde la entidad remota, realiza la validación de la información recibida y ejecuta la inserción del lote de tarjetas innominadas en Base de Datos.

Flujo normal

1. Se recibe el mensaje desde la entidad remota.
2. Se valida la estructura del mensaje.
3. Se valida la presencia y formato de los elementos del mensaje.
4. Se valida la información de la solicitud contra la Base de Datos.
5. Se realiza la inserción del lote de tarjetas innominadas.
6. Se envía la respuesta del mensaje hacia la entidad remota (ver [Códigos de Respuesta](#)).
7. Se envía un mensaje de notificación de sólo almacenamiento hacia TM.

Flujo de excepción

- Paso 2: Si se recibe un mensaje no soportado, se declinará el mensaje con código 12 (Transacción inválida).
- Paso 3: Si se recibe un elemento con un formato diferente al establecido o está ausente, se declinará el mensaje con código 30 (Error de formato).
- Paso 4: Si no se cumple con las validaciones a nivel de Base de Datos de postcard, se declinará el mensaje con el código correspondiente (ver [Códigos de Respuesta](#)).
- Paso 5: Si ocurre un error de Base de Datos, se declinará el mensaje con código 06 (Error).
- Paso 7: Si ocurre un error construyendo el mensaje, no se enviará el mensaje hacia TM, pero el flujo normal continúa su curso.

Nota: La interfaz siempre intentará reportar cualquier fallo mediante evento para obtener mayor información.

Bases de Datos

Esta aplicación se conecta con la BD de postcard para crear y ejecutar el siguiente *Procedimiento Almacenado*:

- **issuingcards.create_anon_card_batch**: Valida la información correspondiente a la inserción de los lotes de tarjetas innominadas, si dicha información es correcta, la inserta en la tabla pc_anonymous_card_batches. Utilizado en los pasos 4 y 5 del flujo normal.

Mensajería

La mensajería soportada por esta interfaz es de tipo JSON, con los siguientes elementos:

Mensaje de solicitud

Campo	Formato	Condición	Observaciones
TransmissionDateTime	n10	O	MMddHHmmss
LocalTime	n6	O	HHmmss
LocalDate	n4	O	MMdd
RetrievalRefNumber	ans..32	M	Número de seguimiento de la transacción
TerminalId	ans8	O	Id de la terminal
CardAcceptorId	ans15	O	Id del establecimiento
CardAccepNameLocation	ans40	M	Nombre y ubicación del establecimiento
InstitutionId	n..11	M	Id del emisor
CardProduction	Array	M	Información de los lotes de tarjetas. Puede contener uno o más elementos

O: Opcional

M: Mandatorio

El **CardProduction** contiene los siguientes elementos:

Campo	Formato	Condición	Observaciones
CardProgName	ans..20	M	Nombre del programa de tarjetas
FromDate	n4	O	YYMM
FromDay	n2	O	dd
ExpiryDate	n4	M	YYMM
ExpiryDay	n2	O	dd
AcctProdName	ans..50	O	Nombre del producto de la cuenta
AcctType	n2	M	Para más información, remitirse a la documentación de Postilion
AcctCurrency	n3	O	Si no está presente, la interfaz establece el valor '000'
AcctBalance	n12	C	Mandatorio si <i>AcctCurrency</i> está presente
NumberOfCards	n..9	M	Número de tarjetas a emitir

O: Opcional
C: Condicional
M: Mandatorio

Mensaje de respuesta

Campo	Formato	Condición	Observaciones
ResponseCode	n2	C	Presente si ocurre un error de formato o peticiones no permitidas
RetrievalRefNumber	ans..32	M	Número de seguimiento de la transacción
CardProduction	Array	C	Información de respuesta por cada lote de tarjetas. Presente si <i>ResponseCode</i> está ausente

C: Condicional

M: Mandatorio

El **CardProduction** contiene la información de respuesta por cada lote de tarjetas:

Campo	Formato	Condición	Observaciones
CardProgName	ans..20	M	Nombre del programa de tarjetas
AcctProdName	ans..50	M	Nombre del producto de la cuenta
BatchNr	n..9	C	Número del lote de tarjetas a emitir. Presente si <i>ResponseCode</i> = 00
ResponseCode	n2	M	Código de respuesta de la inserción

C: Condicional

M: Mandatorio

- Ejemplo de JSON enviado desde el microservicio

```

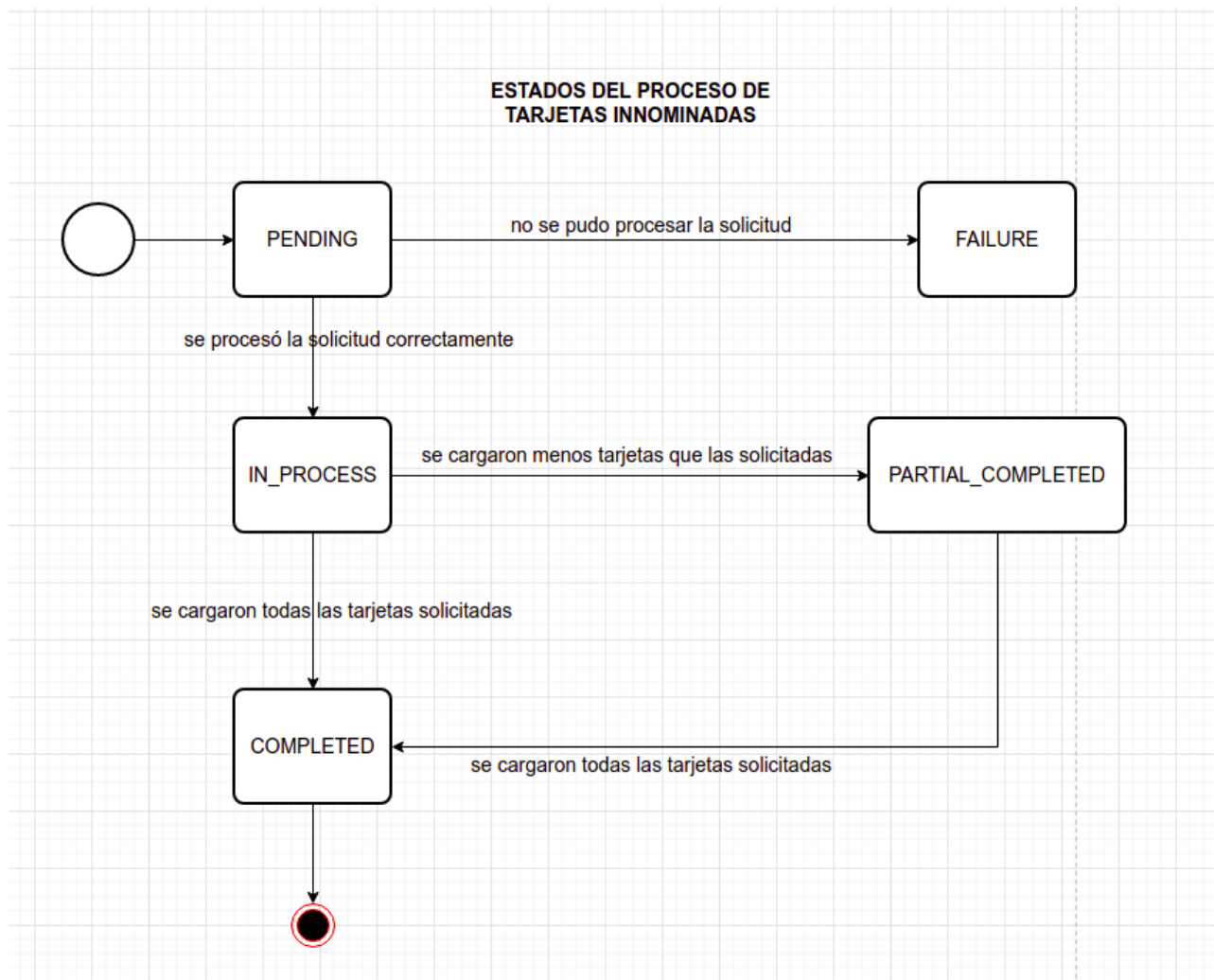
1 {
2   "TransmissionDateTime": "0602152115",
3   "RetrievalRefNumber": "6c13059fafb0439748621077c2770431",
4   "CardAccepNameLocation": "INNOMINATE CARD PROCESS",
5   "InstitutionId": "06",
6   "CardProduction": [
7     {
8       "CardProgName": "Pruebas M Cards API",
9       "ExpiryDate": "2704",
10      "AcctProdName": "EbanxAPIAP",
11      "AcctType": "10",
12      "AcctCurrency": "604",
13      "NumberOfCards": 2
14    }
15  ]
16 }
```

- Ejemplo de respuesta recibida desde IssuingCardsAcq

```

1 {
2   "CardProduction": [
3     {
4       "AcctProdName": "EbanxAPIAP",
5       "BatchNr": "65",
6       "CardProgName": "Pruebas M Cards API",
7       "ResponseCode": "00"
8     }
9   ],
10  "RetrievalRefNumber": "4eb112ce202b4f5daa2501aea83e2e4c"
11 }
```

- El microservicio al recibir la solicitud registra en la tabla **NP_INNO_PROCESS** el proceso con estado **PENDING**, luego se envía la solicitud al Core Digital. Si la respuesta del Core Digital es RC 00 entonces se pasa al estado **IN_PROCESS**, caso contrario se pasa al estado **FAILURE**.



Esquema Pre Autorizador / Full Autorizador

La emisión de tarjetas innominadas soporta los dos esquemas (Pre Autorizador / Full Autorizador).

Cuando el esquema es **Full Autorizador** las tarjetas se crean con las cuentas en cambio cuando el esquema es **Pre Autorizador** sólo se crean las tarjetas.

La configuración del tipo de emisión se establece en **Card Product** (tabla **NP_CARD_PRODUCT**). En la tabla se definió una columna **CP_GEN_ANON_CARDS_WITH_ACCTS**, donde

- **CP_GEN_ANON_CARDS_WITH_ACCTS = 1**, entonces hace emisiones con cuentas (Full autorizador)
- **CP_GEN_ANON_CARDS_WITH_ACCTS = 0**, entonces hace emisiones sin cuenta (Pre autorizador)

Curl

```
curl --location --request POST 'https://t-cards-api.novopayment.net/api-cards-multitenant-core-innominate-issuance/v1/cards/request' \  
--header 'X-Tenant-Id: co-powwi' \  
--header 'X-Request-Id: 3ea0bec5-b83f-4a47-83a1-78fc35e50afc' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "cardProgramName": "Pruebas M Cards API",  
  "numberOfCards": 1,  
  "expiryDate": "2704"  
}'
```

Request

Atributo	Tipo de dato	Descripción	Requerido
cardProgramName	String	Nombre del programa	Si
numberOfCards	Number	Cantidad de tarjetas a solicitar	Si
expiryDate	YYMM	Fecha de expiración de las tarjetas a generar	Si

Response

Atributo	Tipo de dato	Descripción	Requerido
createdDate	Date	Fecha de creación	Si
updatedAt	Date	Fecha de actualización	No
jobs	Array	Colección de trabajos de carga de archivos asociados a dicha solicitud	No
requestId	uuid	Id del request asociado a la petición	Si
uuid	uuid	Identificador asociado al proceso	Si
tenantId	String	Id del tenant que hizo la solicitud	Si
responseCode	String	Código de respuesta emitido por el componente del Core Digital	No
status	String Valores posibles: IN_PROCESS, COMPLETED, PARTIAL_COMPLETED, FAILURE	Estado del proceso	Si

batchNumber	Number	Número de batch informado por el componente del Core Digital	No
numberOfCardsRequested	Number	Cantidad de tarjetas solicitadas para el proceso	Si
numberOfCardsLoaded	Number	Cantidad de tarjetas asociadas a dicho proceso	No

POST

{{scheme}}://{{server}}/{{context-root}}/cards/request ...

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "cardProgramName": "Pruebas-M-Cards-API",
3   "numberOfCards": 1,
4   "expiryDate": "2704"
5 }
```

Body

Cookies

Headers (16)

Test Results

Pretty

Raw

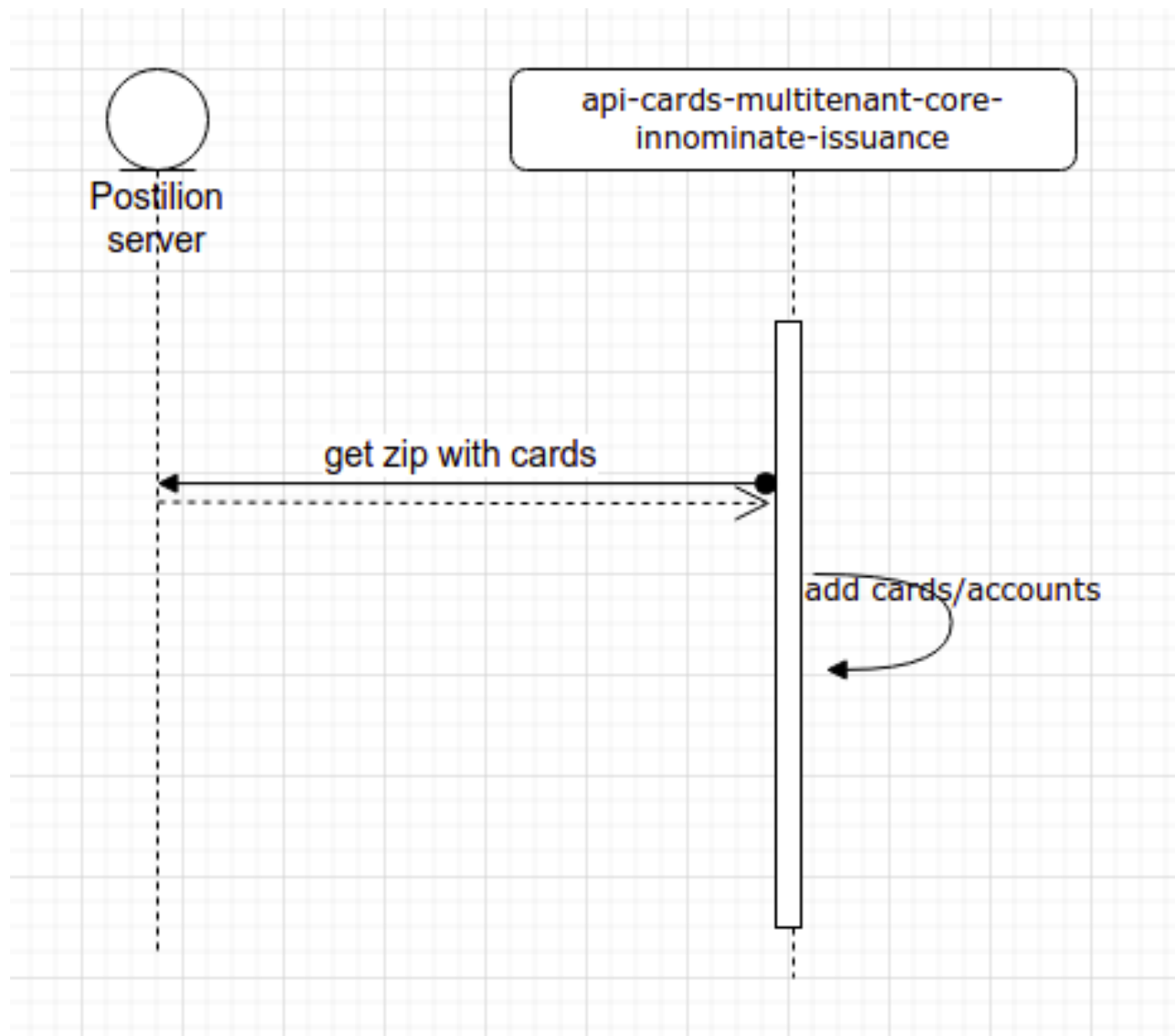
Preview

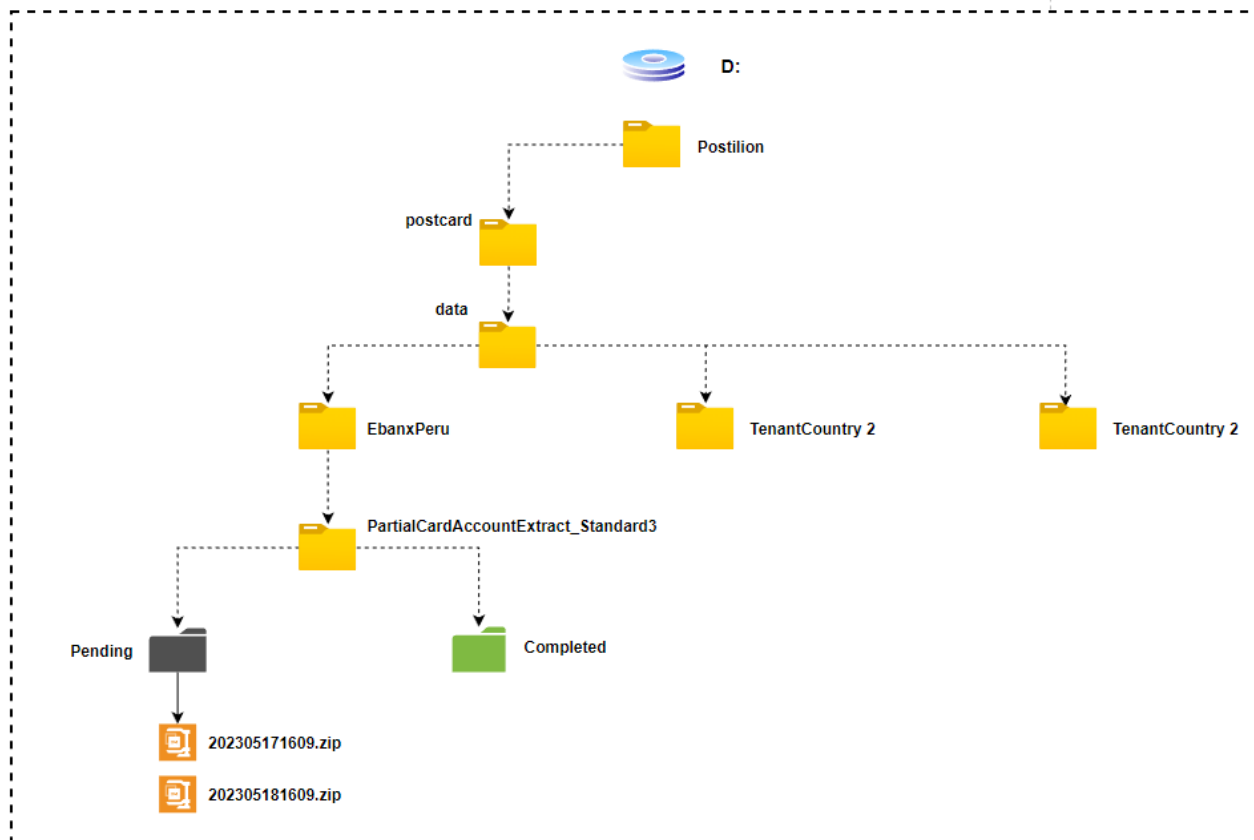
Visualize

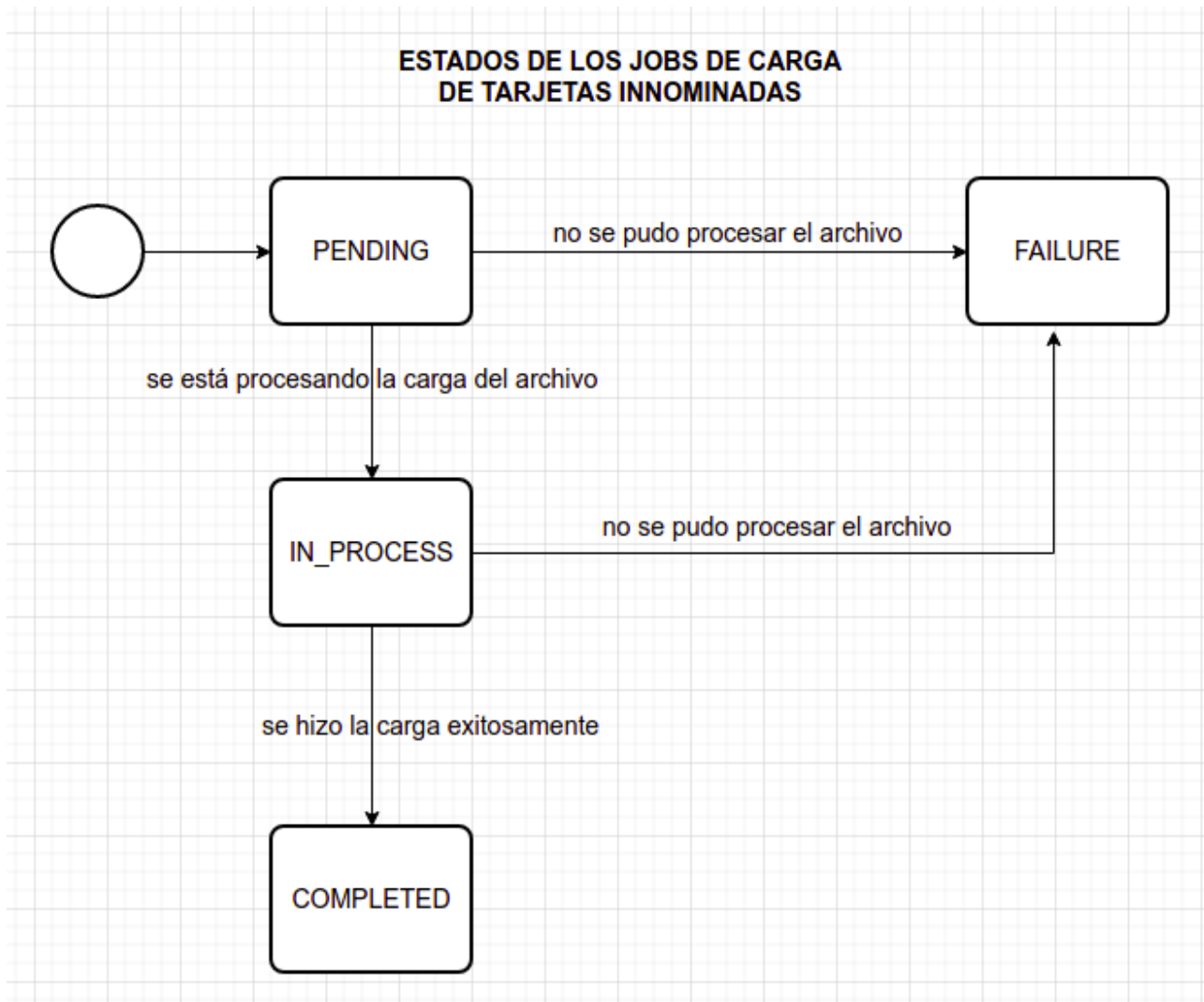
JSON

```
1 {
2   "code": "200.00.000",
3   "datetime": "2023-05-31T11:48:09.677Z",
4   "message": "Process Ok",
5   "data": {
6     "batchNumber": 41,
7     "createdDate": "2023-05-31T11:48:06.874519Z",
8     "jobs": [],
9     "numberOfCardsLoaded": 0,
10    "numberOfCardsRequested": 1,
11    "requestId": "c1483404-5441-4f12-8f87-065be8a50575",
12    "responseCode": "00",
13    "status": "IN_PROCESS",
14    "tenantId": "co-powwi",
15    "uuid": "6ce8e8e26830465d8014b749c93b32c4"
16  }
17 }
```

Carga de Tarjetas Innominadas







Descripción

- El microservicio cuando levanta tiene un Job configurado para procesar los archivos zips cada cierta cantidad de tiempo (configurable en el archivo `application.yml` del microservicio, actualmente son cinco (5) minutos). Los archivos los va a buscar en la carpeta `${innominate.card.process.dir.source}`/**Pending** para cada Tenant configurado. Para ello se conecta con el **Core Multitenant** y obtiene todos los Tenants que el componente tiene configurado (<https://d-core-multitenant.novopayment.net/configurations/v1/components/f1951398-d963-11ed-afa1->

[0242ac120002/tenants](#)).

```
load-innominate-cards-job:
  cron: "0 0/5 * ? * *" # corre cada 5 min
  retries: 3
  encryption-key: ${ENV_INNO_FILE_KEY:ivkBy4ypxxhi8B54V8yAetcXR0e0gWMd}
```

¡IMPORTANTE!: Es sumamente importante validar que la carpeta Pending, Error, Completed estén creadas.

- Al ejecutarse el Job el mismo carga todos los archivos **.zip** que encuentre en la carpeta **Pending**.
- Los archivos zips estarán cifrados con una clave y contendrán los archivos:
 - **Cards.txt**
 - **Accounts.txt**
 - **Cardaccounts.txt**
 - **Output.txt**

necesarios para hacer la carga de las tarjetas innominadas.

- La clave de cifrado se configura en el archivo **application.yml** del microservicio:

```
jobs:
  load-innominate-cards-job:
    retries: 3
    encryption-key: ${ENV_INNO_FILE_KEY:ivkBy4ypxxhi8B54V8yAetcXR0e0gWMd}
```

- Dicha clave debe ser seteada al momento de hacer el despliegue del microservicio con la variable de entorno **ENV_INNO_FILE_KEY**.

¡IMPORTANTE!: Es sumamente importante que dichas claves estén sincronizadas con el equipo del Core cuando configure el componente de innominadas en el servidor de Postilion. Dado que el proceso del Core generará los zips cifrados y el

componente del MiddleWare (api-cards-multitenant-core-innominate-issuance) procesará dichos archivos.

- Cuando el proceso tenga que procesar un archivo .zip, el mismo registrará la carga en las tablas **NP_INNO_JOB**, **NP_INNO_JOB_HISTORY**.
- En la tabla **NP_INNO_JOB** se registra el archivo que se está procesando cuyo nombre de archivo es único para la tabla.
- Se definió con el equipo del Core Digital que el nombre del archivo es **yyyyMMddHHmm.zip** por ejemplo: **202305301200.zip**.
- Como ya se mencionó anteriormente, dichos archivos estarán cifrados y contendrán los archivos **cards.txt**, **accounts.txt**, **cardaccounts.txt**, **output.txt**.
- En la tabla **NP_INNO_JOB_HISTORY** se registran los intentos de proceso de los archivos. Actualmente está configurado para hacer tres (3) intentos: (ver application.yml)

```
jobs:|
  load-innominate-cards-job:
    retries: 3
    encryption-key: ${ENV_INNO_FILE_KEY:ivkBy4ypxxhi8B54V8yAetcXR0e0gWMD}
```

- El proceso sólo va a insertar las tarjetas cuyo **Batch Number** tenga valor (anonymous cards). También va a insertar las cuentas asociadas en caso de existir. Cuando NovoPayment es **pre-autorizador** las tarjetas **NO** tienen cuentas asociadas. En

cambio, cuando es **full-autorizador** se crean las tarjetas/cuentas.

Card-account extract CARDS.TXT					
		<ul style="list-style-type: none"> One character length of the length indicator of the extended field name Length indicator of the extended field name Extended field name One character length of the length indicator of the extended field value Length indicator of the extended field value Extended field value <p>No validation will be performed on individual extended fields.</p>		or custom reporting.	[this represents security_code = 999]
Expiry day	n2_DD	Exactly 2 chars, in the form DD, in the range 01 - 31. If empty, the expiry day is unknown or not applicable (in which case a card expires on the last day of the expiry date's month).	yes	Card expiry day	01
From date	n4_YYMM	Exactly 4 chars, in the form YYMM, where MM is in the range 01 - 12. If empty, the from date is unknown or not applicable.	yes	Card from date	0202
From day	n2_DD	Exactly 2 chars, in the form DD, in the range 01 - 31. If empty, the from day is unknown or not applicable.	yes	Card from day	01
Contactless Discretionary Data	ans..13	Discretionary data field for contactless cards, for example MasterCard PayPass. Various token characters can be used to represent different items of data in the Track2 of contactless cards, like S (sequence number), U (Unpredictable number) and others. The tokens that are available depends on the type of contactless scheme used (VISA or MasterCard). Up to 13 characters.	yes	The structure of discretionary data for track 2 for a contactless card.	AAAACCCSSS
Dynamic CVV Key Index	n..10	Key Index for the DCVV Key.	yes	Used to select a particular key out of a Group key.	1
Batch Number	n..10		yes	For an anonymous card, this indicates the batch in which this card was produced.	17
Last updated date	YYYY-MM-DD hh:mm:ss	Exactly 19 characters.	no	The date and time this record was last updated.	2006-11-30 15:05:10

- **¡IMPORTANTE!** Las tarjetas se generan desde el Core con estado **INACTIVA**.
- Finalmente va a registrar en la tabla **NP_OUTPUT_EMBOSED** todos los registros del archivo output.txt y luego va a invocar al Stored Procedure **EMBOCE** para generar el archivo para la embozadora.

¡IMPORTANTE! El Stored Procedure **EMBOCE** debe ser configurado por cada cliente porque el mismo generará el archivo con el formato especificado por la embozadora.

Adicionalmente a la tabla **NP_OUTPUT_EMBOSED** se le añade el **Card UUID**. El Card UUID se obtiene del campo “**extended card fields**” del archivo **output.txt**. En dicho campo se envían los campos tipo clave/valor. Por ejemplo:

"210card_token2366e64d696-058e-453a-8f61-

ae1bd1ac6710213customer_name212Brielle

Dare"

Entonces el **Card UUID** es **6e64d696-058e-453a-8f61-ae1bd1ac6710** (36 dígitos)

En caso de no venir el Card UUID en el archivo output.txt entonces el mismo es generado en el MiddleWare por el componente **api-cards-multitenant-core-innominate-issuance**

Las tarjetas se asocian con el proceso a través del campo **Batch Number**.

xCHM v. 1.33: PostCard interface Specification

File View Edit History Help

Open... Print... Fonts... Contents Copy Find Fullscreen Back Forward Home About

Contents Index Search

Card-account extract CARDS.TXT

Copyright

Concepts

File formats

File formats overview

Data element format

Loads

Load types

Loader robustness

Load file formats

Extracts

Extract types

Balances extract file formats

Card-account extract file formats

Standard 4

Standard 3

CARDS.TXT

TEMPCARDS.TXT

CARDOVERRIDE LIMITS.TXT

ACCOUNTS.TXT

ACCOUNTBALANCES.TXT

Bookmarks

Add Remove

		Various token characters can be used to represent different items of data in the Track2 of contactless cards, like S (sequence number), U (Unpredictable number) and others.			
		The tokens that are available depends on the type of contactless scheme used (VISA or MasterCard).			
		Up to 13 characters.			
Dynamic CVV Key Index	n..10	Key Index for the DCVV Key.	yes	Used to select a particular key out of a Group key.	1
Batch Number	n..10		yes	For an anonymous card, this indicates the batch in which this card was produced.	17
Last updated date	YYYY-MM-DD hh:mm:ss	Exactly 19 characters.	no	The date and time this record was last updated.	2006-11-30 15:05:10
Last updated user	ans..20		no	The user that last updated this record.	Mary

* Trailing spaces will be stripped from this field.
If the length of this field is incorrect, it will be right-justified and padded with zeroes.

¡IMPORTANTE! LAS TARJETAS INFORMADAS EN EL ARCHIVO CARDS.TXT QUE NO TENGAN REGISTRADO UN NÚMERO DE BATCH EN LA TABLA NP_INNO_PROCESS SERÁN DESCARTADAS.

SÓLO SE CARGAN TARJETAS CUYO BATCH NUMBER ESTÉ REGISTRADO EN LA TABLA NP_INNO_PROCESS

```
[m] #32[Tenants found [co-bancolombia, m-lynk, sv-chiva, co-lifemiles, bn-dilo, co-powwi, bn-tengo] #32[m]
[32>Loading postcard file /postilion-postcard-files/Tengo/PartialCardAccountExtract_Standard3/DEV/Pending/202306141450.zip #x1b[m]
[32>Request from /api-cards-multipayment-nom-innominate-issuance/v1/cards/load #x1b[m]
[32>FileReader #107: #x1b[m] #32>11 cards to be processed #x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004517. Skipping record-#x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004525. Skipping record-#x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004533. Skipping record-#x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004541. Skipping record-#x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004558. Skipping record-#x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004566. Skipping record-#x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004574. Skipping record-#x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004582. Skipping record-#x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004590. Skipping record-#x1b[m]
[32>CardBuilderDelegate #125: #x1b[m] #32Batch number 21 not found for pan 4073180010004608. Skipping record-#x1b[m]
[32>AlgorithmFactory #41: #x1b[m] #32Algorithm type: AES-#x1b[m]
[32>InnominateProcessService #106: #x1b[m] #32m1 cards/accounts to be inserted-#x1b[m]
[32>CardService #59: #x1b[m] #32>associating 1 cards to batch process number 20-#x1b[m]
[32>EmbossedCardService #51: #x1b[m] #32>Saving 11 cards to NP OUTPUT #8055ED #x1b[m]
[32>InnominateJobService #146: #x1b[m] #32>Moving file /postilion-postcard-files/Tengo/PartialCardAccountExtract_Standard3/DEV/Pending/202306141450.zip to /postilion-postcard-files/Tengo/P...
[32>InnominateCardLoadService #87: #x1b[m] #32>Success loading innominate cards, response: 000-#x1b[m]

[32>ServiceCommonHttpResponseDecorator #67: #x1b[m] #32>Headers: [transfer-encoding:[chunked], Content-Type:[application/json;charset=UTF-8], Content-Length:[868], X-Request-Id:[]]-#x1b[m]
[32>ServiceCommonHttpResponseDecorator #68: #x1b[m] #32>Response type [MonoContextWrite] from URL [null] with state code [200 OK]-#x1b[m]
[32>ServiceCommonHttpResponseDecorator #69: #x1b[m] #32>Response: {"code":"200.00.000","datetime":"2023-06-14T22:40:30.244Z","message":"Process OK","data":{"createdDate":"2023-06-14T22:40:22.467281Z","f...
[32>Success load innominate cards, response: {"code":"200.00.000","datetime":"2023-06-14T22:40:30.244Z","message":"Process OK","data":{"createdDate":"2023-06-14T22:40:22.467281Z","fileName":"/postilion-p...
```

Cuando se hace una emisión se registra en la tabla **NP_INNO_PROCESS** el número de batch asociado a dicha petición.

Results	Meta data	Info	Overview / Charts	Rotated table	Results as text	Positive Row 1, Col		
IN_CARD_PROGRAM_N...	IN NUMBER OF CARDS	IN UUID	IN_TENANT_ID	IN_REQUEST_ID	IN_BATCH_N...	IN STATUS		
Pruebas M Cards API	7	e3c06514ce0e4133b7ab43a64d8e88fc	co-powwi	629011e2-eded-4a0b-b...	00	61	IN PROCESS	
Pruebas M Cards API	3	6c13059fafb0439285755077c2770431	co-powwi	5a93bd1d-3882-40d5-b...	00	59	IN PROCESS	
Pruebas M Cards API	5	8f585a8d396c4e828df77c792f65d486	co-powwi	0918c3df-4400-435d-9...	00	58	IN PROCESS	
Pruebas M Cards API	4	dbbdef2b17e446f8b17b16bbb9f73df	co-powwi	1ec43cef-7437-44a5-a...	00	57	COMPLETED	
Pruebas M Cards API	2	f3c23f96f1c2430bbcb3a21dcf5946aa	co-powwi	35eb63a8-d270-464c-b...	00	56	COMPLETED	
Pruebas M Cards API	1	1a379ed9d7b434e038284d6112217e...	co-powwi	012be295-847a-4698-b...	00	55	COMPLETED	

Una vez que se procesan los archivos, las tarjetas se asocian al proceso a través de la tabla **NP_INNO_PROCESS_CARD**. Finalmente se actualiza el estado del proceso:

- Si tarjetas asociadas >= tarjetas solicitadas => **COMPLETED**
- Si tarjetas asociadas < tarjetas solicitadas => **PARTIAL_COMPLETED**

Por ejemplo:

Nro. de proceso	Tarjetas solicitadas	Tarjetas asociadas	Estado
1	4	4	COMPLETED
2	5	3	PARTIAL_COMPLET ED
3	8	0	IN_PROCESS
4	50	0	IN_PROCESS

Las cargas de los archivos se asocian con los procesos a través de la tabla **NP_INNO_PROCESS_JOB**

Reintentos

- Cada vez que se procesa un archivo se registra el Job en la tabla **NP_INNO_JOB** (en caso de no existir).
- En la tabla **NP_INNO_JOB** hay un Job por archivo y se registra el intento de carga en la tabla **NP_INNO_JOB_HISTORY**.
- Si hay al menos n registros con estado **FAILURE** en la tabla **NP_INNO_JOB_HISTORY**, entonces, el Job pasa a estado **FAILURE** y no se procesa más el archivo (el mismo es movido de la carpeta **Pending** a la carpeta **Error**).
- La cantidad de reintentos se configuran en el archivo **application.yml** del microservicio:

```
jobs:
  load-innominate-cards-job:
    retries: 3
    encryption-key: ${ENV_INNO_FILE_KEY:ivkBy4ypxxhi8B54V8yAetcXR0e0gWMd}
```

Por ejemplo:

1. Primer intento con error:

NP_INNO_JOB

Número de Job	Archivo	Estado
1	202305311200.zip	PENDING

NP_INNO_JOB_HISTORY

Número de Job	Estado
1	FAILURE

2. Segundo intento con error:

NP_INNO_JOB

Número de Job	Archivo	Estado
1	202305311200.zip	PENDING

NP_INNO_JOB_HISTORY

Número de Job	Estado
1	FAILURE
1	FAILURE

3. Tercer intento con error:

NP_INNO_JOB

Número de Job	Archivo	Estado
1	202305311200.zip	FAILURE

NP_INNO_JOB_HISTORY

Número de Job	Estado
1	FAILURE
1	FAILURE
1	FAILURE

Ejemplos de carga de archivos

Caso 1

- Se crea un archivo zip en `${innominate.card.process.dir.source}/Pending`. Ejemplo `${innominate.card.process.dir.source}/Pending/202305311230.zip`
- Se registra el Job en la tabla **NP_INNO_JOB**, en caso de no existir, con estado **IN_PROCESS**
- Se registra el Job en la tabla **NP_INNO_JOB_HISTORY** con estado **IN_PROCESS**
- Se cargan las tarjetas/cuentas
- Se relacionan las tarjetas con los procesos **NP_INNO_PROCESS_CARD**
- Se relacionan los procesos con el Job **NP_INNO_PROCESS_JOB**
- Se actualizan los estados de los procesos **NP_INNO_PROCESS**
- Se actualizan los estados de **NP_INNO_JOB**, **NP_INNO_JOB_HISTORY** a **COMPLETED**
- Se mueve el archivo a `${innominate.card.process.dir.source}/Completed`. Ejemplo `${innominate.card.process.dir.source}/Completed/202305311230.zip`

Caso 2

- Se crea un archivo zip en `${innominate.card.process.dir.source}/Pending`. Ejemplo `${innominate.card.process.dir.source}/Pending/202305311230.zip`

- Se registra el Job en la tabla **NP_INNO_JOB**, en caso de no existir, con estado **IN_PROCESS**
- Se registra el Job en la tabla **NP_INNO_JOB_HISTORY** con estado **IN_PROCESS**
- Hubo un error en la carga
- Se actualiza el estado de **NP_INNO_JOB_HISTORY** a **FAILURE**
- Se actualiza el estado de **NP_INNO_JOB** a **PENDING**

Caso 3

- Se crea un archivo zip en `${innominate.card.process.dir.source}/Pending`. Ejemplo `${innominate.card.process.dir.source}/Pending/202305311230.zip`
- Se registra el Job en la tabla **NP_INNO_JOB**, en caso de no existir, con estado **IN_PROCESS**
- Se registra el Job en la tabla **NP_INNO_JOB_HISTORY** con estado **IN_PROCESS**
- Hubo un error en la carga y se alcanzó a la cantidad máxima de reintentos (`select count(*) from NP_INNO_JOB_HISTORY where status='FAILURE' >= CANTIDAD_MAX_REINTENTOS`)
- Se actualiza el estado de **NP_INNO_JOB_HISTORY** a **FAILURE**
- Se actualiza el estado de **NP_INNO_JOB** a **FAILURE**
- Se mueve el archivo a `${innominate.card.process.dir.source}/Error`. Ejemplo `${innominate.card.process.dir.source}/Error/202305311230.zip`

Carga Manual de Tarjetas

El microservicio **api-cards-multitenant-core-innominate-issuance** tiene un endpoint HTTP para hacer una carga manual de un archivo zip en caso de requerirlo.

Para hacer una carga manual se valida que la ruta del archivo este contenida dentro del parámetro de configuración [innominate.card.process.dir.source](#)

Por ejemplo:

Se configura el siguiente parametro para el cliente Ebanx

```
"innominate.card.process.dir.source" = "/postilion-postcard-files/ EbanxPeru  
/PartialCardAccountExtract_Standard3"
```

Los siguientes archivos son válidos:

- "/postilion-postcard-files/EbanxPeru/PartialCardAccountExtract_Standard3/DEV/Pending/202305291647.zip"
- "/postilion-postcard-files/EbanxPeru/PartialCardAccountExtract_Standard3/Pending/202305291647.zip"
- "/postilion-postcard-files/EbanxPeru/PartialCardAccountExtract_Standard3/202305291647.zip"

Los siguientes archivos no son válidos:

- "/postilion-postcard-files/Tengo/PartialCardAccountExtract_Standard3/202305291647.zip"
(Tenant Tengo en lugar de EbanxPeru)
- "202305291647.zip"
- "/postilion-postcard-files/202305291647.zip "

Curl

```
curl --location --request POST 'https://d-cards-api.novopayment.net/api-cards-multitenant-  
core-innominate-issuance/v1/cards/load' \  
--header 'X-Tenant-Id: co-powwi' \  
--header 'X-Request-Id: 08a6c634-f8fa-4840-a735-f13ba4105926' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "zipFile": "/postilion-postcard-  
files/EbanxPeru/PartialCardAccountExtract_Standard3/DEV/Pending/202305291647.zip"  
}'
```

Acá se exhibe un ejemplo de carga.

The screenshot shows a text file named 'cards.txt' with a large volume of data. The data is organized in columns, including card numbers, API status, and timestamps. A red box highlights a specific section of the file.

Se cargó el archivo **/postilion-postcard-files/EbanxPeru/PartialCardAccountExtract_Standard3/Pending/202306021539.zip** que tenía tarjetas para 4 procesos (Bach Number: 58, 59, 60, 61). Las tarjetas del proceso 60 se descartaron porque no existe dicho proceso en **NP_INNO_PROCESS**

Rows 29; select * from NP_INNO_PROCESS;

Selected Rows: 0, Cols: 0

Position: Row: , Col:

Results

Meta data

Info

Overview / Charts

Rotated table

Results as text

NAME	IN_NUMBER_OF_CARDS	IN_UUID	IN_TENAN...	IN_REQUEST_ID	IN...	IN_BATCH_NUMBER	IN_STATUS
3	rd/348dbb...	co-powwi	4c531b82-ec81-4052...	00	146	COMPLETED	
3	288739d9c...	co-powwi	c33d99ea-a9d7-410f-b...	00	145	COMPLETED	
1	bead2904...	co-powwi	72fd7dc0-4920-4d01-9...	00	144	COMPLETED	
1	bd76ec4a2...	co-powwi	7e2ac159-d051-4107...	00	143	COMPLETED	
7	e3c06514c...	co-powwi	629011e2-eded-4a0b...	00	61	COMPLETED	
3	6c13059faf...	co-powwi	5a93bd1d-3882-40d5...	00	59	COMPLETED	
5	8f585a8d3...	co-powwi	0918c3df-4400-435d-9...	00	58	COMPLETED	
4	dbbdef2b1...	co-powwi	1ec43cef-7437-44a5-a...	00	57	COMPLETED	
2	f3c23f9f1...	co-powwi	35eb63a8-d270-464c...	00	56	COMPLETED	
1	a379ed9d...	co-powwi	012be295-847a-4698...	00	55	COMPLETED	
10	9dfc37dffe...	co-powwi	276c3c9b-5af3-41cb-9...	00	54	COMPLETED	

```

▼ object {4}
  code : 200.00.000
  datetime : 2023-06-05T13:00:29.741Z
  message : Process Ok
▼ data {5}
  createdAt : 2023-06-02T21:09:01.865022Z
  fileName : /postilion-postcard-files/EbanxPeru/PartialCardAccountExtract_Standard3/Pending/202306021539.zip
  processes [3]
    ▼ 0 {11}
      batchNumber : 58
      createdAt : 2023-06-02T18:20:49.386563Z
      jobs [1]
        numberOfCardsLoaded : 5
        numberOfCardsRequested : 5
        requestId : 0918c3df-4400-435d-92c6-a7dbf0a91884
        responseCode : 00
        status : COMPLETED
        tenantId : co-powwi
        updatedAt : 2023-06-05T13:00:21.969637Z
        uuid : 8f585a8d396c4e828df77c792f65d486
    ▼ 1 {11}
      batchNumber : 59
      createdAt : 2023-06-02T18:21:15.880134Z
      jobs [1]
        numberOfCardsLoaded : 3
        numberOfCardsRequested : 3
        requestId : 5a93bd1d-3882-40d5-b943-ae669364e4bc
        responseCode : 00
        status : COMPLETED
        tenantId : co-powwi
        updatedAt : 2023-06-05T13:00:18.558726Z
        uuid : 6c13059fafb0439285755077c2770431
    ▼ 2 {11}
      batchNumber : 61
      createdAt : 2023-06-02T18:33:30.339790Z
      jobs [1]
        numberOfCardsLoaded : 7
        numberOfCardsRequested : 7
        requestId : 629011e2-eded-4a0b-b6d0-c4d1a64e2f18

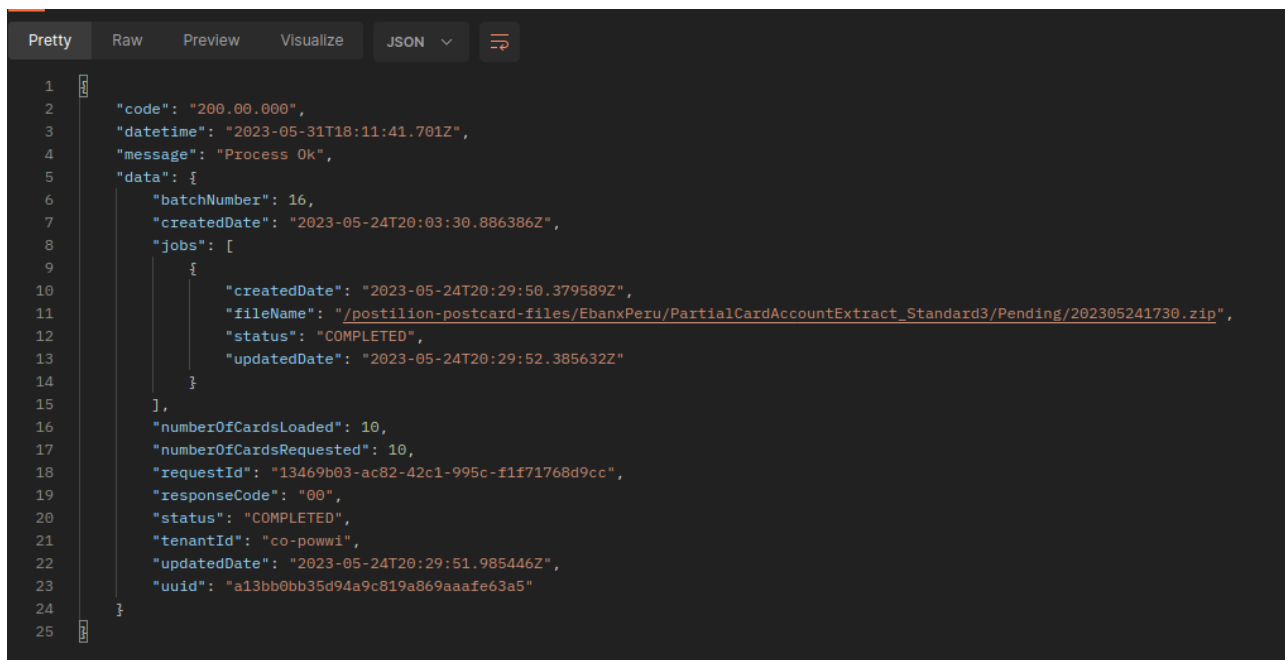
```

Consulta de Procesos

El microservicio **api-cards-multitenant-core-innominate-issuance** expone un endpoint HTTP para consultar los procesos por uuid

Curl

```
curl --location --request GET 'https://t-cards-api.novopayment.net/api-cards-multitenant-core-innominate-issuance/v1/processes/050680c125944fb58252da8a68a6f49a' \
--header 'X-Tenant-Id: co-powwi' \
--header 'X-Request-Id: d85960c3-59d3-477a-a9d3-cd1464a34688' \
--header 'Content-Type: application/json'
```



```
1  {
2    "code": "200.00.000",
3    "datetime": "2023-05-31T18:11:41.701Z",
4    "message": "Process Ok",
5    "data": {
6      "batchNumber": 16,
7      "createdAt": "2023-05-24T20:03:30.886386Z",
8      "jobs": [
9        {
10         "createdAt": "2023-05-24T20:29:50.379589Z",
11         "fileName": "/postilion-postcard-files/EbanxPeru/PartialCardAccountExtract_Standard3/Pending/202305241730.zip",
12         "status": "COMPLETED",
13         "updatedAt": "2023-05-24T20:29:52.385632Z"
14       }
15     ],
16     "numberOfCardsLoaded": 10,
17     "numberOfCardsRequested": 10,
18     "requestId": "13469b03-ac82-42c1-995c-f1f71768d9cc",
19     "responseCode": "00",
20     "status": "COMPLETED",
21     "tenantId": "co-powwi",
22     "updatedAt": "2023-05-24T20:29:51.985446Z",
23     "uuid": "a13bb0bb35d94a9c819a869aaafe63a5"
24   }
25 }
```

- Se puede observar que del archivo "/postilion-postcard-files/EbanxPeru/PartialCardAccountExtract_Standard3/Pending/202305241730.zip" se cargaron 10 tarjetas y se asociaron al proceso con batchNumber 16.
- La carga de archivos comenzó a las "2023-05-24T20:29:50.379589Z" y finalizó a las "2023-05-24T20:29:52.385632Z"
- Para el proceso se solicitaron 10 tarjetas y se cargaron 10 tarjetas por lo que el proceso quedó con estado COMPLETED

Configuraciones

- El microservicio se configura por Tenant a través del archivo [api-cards-multitenant-core-innominate-component.json](#)
- Dicho archivo se configura en el repositorio <https://code.novopayment.com/novopayment/api-microservices-configurations/tree/uat/api-microservices/configurations/cards-solution/cards-product> dependiendo de cada ambiente.
- El archivo tiene el servidor/puerto para hacer las solicitudes con el componente de Innominadas del Core Digital y la ruta donde ir a buscar los archivos para cargar las cuentas/tarjetas entre otras configuraciones

```
1 {
2   "certificates": [],
3   "params": [
4     {
5       "name": "innominate.card.process.server",
6       "value": "SRVPOSTINT2.novopayment.dom"
7     },
8     {
9       "name": "innominate.card.process.server.port",
10      "value": "9048"
11    },
12    {
13      "name": "innominate.card.process.dir.source",
14      "value": "/postilion-postcard-files/Tengo/PartialCardAccountExtract_Standard3"
15    },
16    {
17      "name": "innominate.card.process.issuer",
18      "value": "26"
19    },
20    {
21      "name": "innominate.card.process.max.cards.request.per.day",
22      "value": "5000"
23    }
24  ]
25 }
```

- Para que se puedan hacer emisiones al equipo del Core Digital es necesario habilitar el puerto en dicho Servidor.
- Para que los archivos sean visibles desde el microservicio desplegado en la Nube se debe **configurar un NFSⁱⁱ** -mediante ticket generado al equipo de infraestructura-; para tener acceso a las carpetas del Servidor del Core Digital.
- Una vez configurado el NFS es necesario **configurar un Persistent Volume Claim (PVC)ⁱⁱⁱ** para que apunte a D:\Postilion\postcard\data\. Para este caso sería /postilion-postcard-files => D:\Postilion\postcard\data\

```
postilion-postcard-files      Bound      postilion-postcard-files      100Gi      RWO      storage-nfs      16s
```

- Finalmente, en el Pod hay que configurar el NFS en el archivo template de helm.

```
root@api-cards-multitenant-core-innominate-issuance-k8s-dpmt-679j2k: /# ls
bin boot configurations dev elastic etc home lib lib64 logs media mnt opt postilion-postcard-files proc root run sbin srv sys tmp usr var
root@api-cards-multitenant-core-innominate-issuance-k8s-dpmt-679j2k: /# cd postilion-postcard-files/
root@api-cards-multitenant-core-innominate-issuance-k8s-dpmt-679j2k: /postilion-postcard-files# ls
AstroPay      CardProduction_BancodeGuayaquil_Standard4  CardProduction_GrupoGenteCR_Standard4  CardProduction_TigoMoneyNHL_Standard4  GrupoGenteCRUSD  SodexoColombia
BNPFullAuth   CardProduction_BancodeGuayaquil_Standard4  CardProduction_LuloBank_Standard4        CardProduction_TigoMoneyPA_Standard4  GrupoGente6      TebcaPeru
BNPPreAuth    CardProduction_Banreservas_Standard4       CardProduction_MFTECH_Standard4          CardProduction_TuPanaDOP_Standard4  GrupoGente8      Tengo
Balances      CardProduction_Chivo_Standard4              CardProduction_MIO_Standard4             Chivo                                GrupoGenteSFS    TigoMoneyNHL
BancodeGuayaquil  CardProduction_Coopcentral_Standard4       CardProduction_Ontop_Standard4           Coopcentral                        LuloBank         TigoMoneyPA
BancodeGuayaquil  CardProduction_Dale_Standard4              CardProduction_ProdubancoEcuador_Standard4  Dale                               MFTECH           TuPanaDOP
Banreservas    CardProduction_Dilo_Standard4              CardProduction_ServitebcaPeru_Standard4  Dilo                               MIO
CardProduction_AstroPay_Standard4  CardProduction_EbanxPeru_Standard4       CardProduction_SodexoColombia_Standard4  EbanxPeru                      Ontop
CardProduction_BNPFullAuth_Standard4  CardProduction_Fihogar_Standard4        CardProduction_TebcaPeru_Standard4       Fihogar                        ProdubancoEcuador
CardProduction_BNPPreAuth_Standard4  CardProduction_GrupoGenteCRUSD_Standard4  CardProduction_Tengo_Standard4          GrupoGenteCR                  ServitebcaPeru
```

Acá se listan los tickets requeridos para configurar el entorno de DEV

- <https://jira4novo.atlassian.net/browse/NP-17104>
- <https://jira4novo.atlassian.net/browse/NP-17105>
- <https://jira4novo.atlassian.net/browse/NP-17273>
- <https://jira4novo.atlassian.net/browse/NP-17274>

ⁱ <https://www.rfc-es.org/rfc/rfc1180-es.txt>

ⁱⁱ <https://datatracker.ietf.org/doc/html/rfc7530>

ⁱⁱⁱ <https://kubernetes.io/docs/concepts/storage/persistent-volumes/>