



Descripción técnica servicios de integración SISCOOP - NEO



siscoop

15
AÑOS
GENERANDO
VALOR

Detalle tema:

La integración de SISCOOP con NEO permite a las entidades financiera desarrollar sus APIs y transformar sus cajeros automáticos en multifuncionales mediante la implementación de servicios clave.

Contenido

Introducción.....	3
1. Token.....	4
1.1 Descripción de solicitud	4
1.2 Formato de la solicitud	4
1.3 Descripción de la respuesta.....	15
1.4 Formato de la respuesta	5
<u>2. Consulta cuenta</u>	<u>6</u>
2.1 Descripción de solicitud	6
2.2 Formato de la solicitud	6
2.3 Descripción de la respuesta.....	7
2.3.1 validaciones	8
2.4 Formato de la respuesta	8
<u>3. Depósito Cuenta/Pago Crédito</u>	<u>9</u>
3.1 Descripción de solicitud	9
3.1.1 validaciones de la solicitud	10
3.2 Formato de la solicitud	10
3.3 Estructura de la firma.....	11
3.3.1 Proceso de la firma	11
3.4 Ejemplos de las respuestas.....	12,13,14
4. Manejo de Excepciones en Swagger.....	15
5. Conclusión.....	16

Introducción

Este documento describe los servicios integrados en el cajero automático para realizar transacciones de depósito en efectivo y pago de crédito a través de una conexión con varios endpoints REST. El proceso está basado en la autenticación y autorización con el estándar OAuth2.0 para garantizar la seguridad de las transacciones. Se abordan los flujos de interacción entre el cajero y los servicios, especificando cómo el cajero obtiene un token de acceso, consulta las cuentas disponibles y realiza depósitos o pagos a crédito.

Los servicios cubiertos incluyen:

Token de autenticación: El cajero obtiene un token mediante el envío de las credenciales del usuario.

Consulta de cuenta: El cajero obtiene información detallada sobre la cuenta seleccionada por el usuario para realizar un depósito o pago.

Depósito a cuenta/Pago a crédito: El cajero envía los datos de la transacción para realizar el depósito o pago correspondiente.

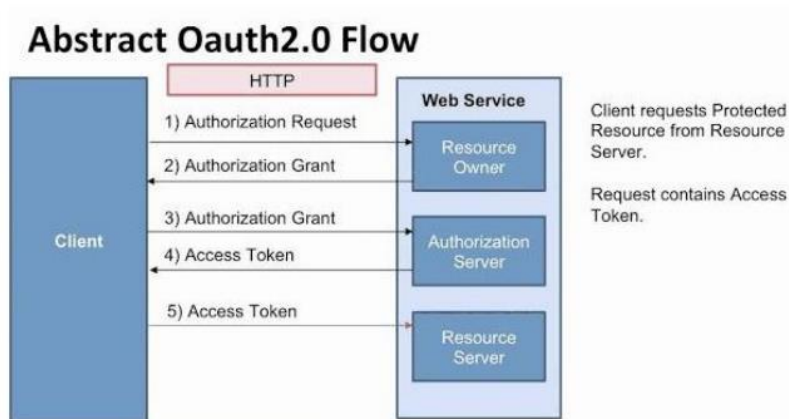
Este documento detalla los pasos necesarios para interactuar con cada uno de estos servicios, así como las validaciones que deben realizarse durante el proceso para garantizar la integridad y validez de las transacciones.

1. -Token

Cuando el Usuario selecciona el botón de Deposito en efectivo/Pago de crédito el cajero envía una petición al servicio token, el cajero se integrará con el endpoint REST Token enviando una solicitud POST ejemplo:

URL: <https://siscoop.test.mx/Token>

Este servicio implementa un flujo de autenticación basado en el estándar OAuth2.0.



1.1.- El cajero envía las credenciales en el cuerpo de la solicitud, y el servidor responde con un token de acceso.

Campo de Servicio	Descripción	Origen del Valor
User	String - Nombre de usuario para la autenticación	Proporcionado por el cliente
Pwd	String - Contraseña del usuario	Proporcionado por el cliente

1.2.-Formato de envío:

Las credenciales de autenticación (user, pwd) se envían en el cuerpo de la solicitud en formato JSON. El cuerpo se envía en RAW/JSON.

```
{
  "user": "Usuario",
  "pwd": "Password"
}
```

1.3- Respuesta de Autenticación

Campo de Servicio	Descripción	Origen del Valor
success	Indica si la autenticación fue exitosa (true/false)	Generado por el servidor
token	Token de acceso en formato JWT con el prefijo 'Bearer'	Generado por el servidor
expiresIn	Tiempo en segundos hasta que el token expira	Generado por el servidor

1.4.-Formato de respuesta:

El servidor devuelve una respuesta en JSON que incluye los detalles del token de acceso. Este token es necesario para acceder a los demás servicios.

```
{
  "success": true,
  "token": "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIyIiwianRpIjoibWJkMGFhYTdkYzE3YjIwNGU2ZjU4YzJjO
GJiOTAwMDhiNWZmNzBIMjg3NDk",
  "expiresIn": 527039
}
```

Uso del Token en Otros Servicios

El token proporcionado sigue el estándar OAuth2.0 y se utilizará en el encabezado Authorization en cada solicitud posterior a otros servicios. El token se envía en formato Bearer.

Ejemplo de encabezado de solicitud:

```
Authorization: Bearer <token>
```

- OAuth2.0 se utiliza para gestionar la autenticación y autorización.
- Datos de autenticación (user, pwd): Se envían en RAW/JSON en el cuerpo de la solicitud.
- Token de acceso: Devuelto en la respuesta de autenticación y se usa en el encabezado Authorization con el prefijo Bearer.

2.- Consulta cuenta

Una vez obtenido el token de autenticación, el cajero solicita al usuario ingresar el número de cuenta o crédito al que desea depositar o pagar. El usuario introduce los datos y selecciona el botón *Continuar*.

El cajero realiza una integración con el endpoint REST SrvConsultaCuenta enviando una solicitud POST con el token obtenido previamente en el encabezado.

Detalles del servicio

- **URL del servicio EJEMPLO:**
https://siscoop.mx/SrvConsultaCuenta
- **Método HTTP:**
POST
- **Encabezados requeridos:**
 - **Authorization:** Token generado en el servicio de autenticación.

2.1.- Cuerpo de la solicitud

A continuación, se desglosan los campos del cuerpo de la solicitud:

Campo	Descripción	Origen del valor
idCajero	String- Identificador único del cajero que realiza la transacción.	Proviene de la configuración del cajero.
fechaHora	Fecha y hora de la solicitud en formato ISO 8601.	Generado automáticamente por el cajero.
cuenta	String- Número de cuenta o crédito ingresado por el usuario.	Capturado por el usuario en la pantalla del cajero.
tipoCuenta	Integer - Tipo de cuenta (0 para cuentas regulares, 1 para créditos).	Definido según la operación solicitada por el usuario.

2.2.- Ejemplo del formato de envío (RAW/JSON)

El cuerpo de la solicitud enviado al endpoint tendrá el siguiente formato:

```
{
  "idCajero": "12345",
  "fechaHora": "2024-12-13T15:37:15.290Z",
  "cuenta": "9876543210",
  "tipoCuenta": 0
}
```

se agrega el encabezado Authorization, debe incluir el token con el prefijo Bearer, ejemplo:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6I...
```

2.3- Respuesta esperada

Campo	Descripción	Origen del valor
ObjetoInformacion	Contiene los datos detallados de la cuenta consultada.	Calculado por el servicio con base en la solicitud y las validaciones aplicadas.
DatosCuenta	Detalles de la cuenta, incluyendo información específica de depósitos y cambios.	Consultado en la base de datos de la entidad.
cuenta	String- La cuenta ingresada por el cliente.	Capturada en la solicitud desde el cajero.
descripcionCuenta	String- Descripción de la cuenta.	Obtenida de la entidad con base en el número de cuenta.
detalleCuenta	String- Información adicional de la cuenta (puede incluir estatus u otros metadatos).	Calculada por el servicio.
montoProximoPago	Double.-Monto correspondiente al próximo pago, solo aplicable para cuentas de crédito.	Obtenido de las reglas del negocio de la entidad.
minDeposito	Double .-Monto mínimo permitido para realizar depósitos en la cuenta.	Definido por las reglas de negocio de la entidad.
maxDeposito	Double.- Monto máximo permitido para realizar depósitos en la cuenta.	Definido por las reglas de negocio de la entidad.
cuentaDefault	String- Cuenta a la que se depositará en caso de que el cajero no pueda dar cambio.	Proporcionada por la entidad o enviada como la misma cuenta ingresada si no aplica.
descripcionCuentaDefault	String- Descripción asociada a cuentaDefault.	Proporcionada por la entidad.
cuentaCambio	String- Cuenta integradora donde se depositará el cambio si el cajero no puede entregarlo físicamente.	Proporcionada por la entidad o enviada como la misma cuenta ingresada si no aplica.
descripcionCuentaCambio	String- Descripción asociada a cuentaCambio.	Proporcionada por la entidad.
maxCambio	Double .-Monto máximo permitido para cambio en la cuenta integradora.	Regla fija: 1000.
EstatusProceso	Int .-Indica si la solicitud fue exitosa (0 = éxito, otros valores = error).	Calculado por el servicio con base en las validaciones realizadas.
IdTransaccion	String- Identificador único para la transacción procesada.	Generado automáticamente por el servicio.
Mensaje	String- Mensaje de respuesta, útil para mostrar errores en caso de fallas.	Calculado por el servicio según el estatus de la solicitud.

2.3.1- Validaciones

- **idCajero:** Debe existir en la entidad antes de procesar la solicitud.
- **fechaHora:** Debe estar en formato UTC ISO 8601.
- **cuenta:**
 - Debe ser válida dentro del sistema de la entidad.
 - El tipo de cuenta debe coincidir con el campo tipoCuenta:
 - 0: Cuentas de ahorro.
 - 1: Cuentas de crédito.
- **tipoCuenta:** Solo permite valores 0 (ahorro) o 1 (crédito).
- **maxCambio:** Valor fijo de 1000.
- **minDeposito y maxDeposito:** Definidos por las reglas de negocio de la entidad.
- **montoProximoPago:** Aplica únicamente para cuentas de crédito

2.4- Ejemplo de respuesta exitosa (JSON RAW)

```
{
  "ObjetoInformacion": [
    {
      "DatosCuenta": {
        "cuenta": "9876543210",
        "descripcionCuenta": "Beneficiario:\n ***1567-Ahorro\n M*** A*** A*** O***",
        "detalleCuenta": "Cuenta activa para depósitos.",
        "montoProximoPago": 1500,
        "minDeposito": 50,
        "maxDeposito": 10000,
        "cuentaDefault": "9876543210",
        "descripcionCuentaDefault": "Cuenta principal",
        "cuentaCambio": "1234567890",
        "descripcionCuentaCambio": "Cuenta integradora",
        "maxCambio": 1000
      }
    }
  ],
  "EstatusProceso": 0,
  "IdTransaccion": "TX12345678",
  "Mensaje": "Consulta realizada con éxito."
}
```

El formato de presentación de descripcionCuenta incluye los últimos 4 dígitos de la cuenta y debe enmascarar información sensible para proteger al usuario, Fijar de la siguiente manera:

"descripcionCuenta": "Beneficiario:\n *1567-Ahorro\n M*** A*** A*** O***"**

3.- DepositoCuenta/PagoCredito

3.1- Desglose de la solicitud

Campo	Descripción	Origen del valor
idCajero	String- Identificador único del cajero desde donde se realiza la transacción.	Capturado en el sistema y validado en la base de datos de la entidad.
fechaHora	Fecha y hora de la solicitud en formato UTC ISO 8601.	Generado por el cajero al momento de la transacción.
tipoTX	Integer .-Tipo de transacción:	Determinado por la acción del usuario:
	- 0: Depósito a cuenta de ahorro.	- 1: Pago a crédito.
	- 3: Depósito de cambio no entregado.	- 4: Depósito a cuenta para retiro en sucursal.
referencia	String- Identificador de la transacción anterior asociada a la actual.	Capturado desde el sistema para relacionar transacciones.
cuenta	String- Número de cuenta ingresado por el usuario o la cuenta default seleccionada si aplica.	Ingresado por el cliente o calculado en el servicio según la lógica de cambio no entregado.
secuencia	Integer .-Número de secuencia generado por el cajero para llevar control interno de las operaciones.	Proporcionado por el cajero.
montoIngresado	Double .-Monto físico ingresado por el cliente en el cajero.	Capturado por el hardware del cajero.
montoDeposito	Double .-Monto que el cliente desea depositar en la cuenta seleccionada.	Determinado por la selección del cliente en el cajero.
montoCambio	Double .-Monto de cambio que no pudo ser entregado físicamente al cliente.	Calculado por el sistema del cajero.
firma	String- Hash o firma electrónica de la transacción para garantizar la integridad y autenticidad de los datos.	Calculado utilizando datos de la transacción y una clave de la entidad; se detalla más adelante.

3.1.1- Validaciones

1. **idCajero:**
 - El cajero debe estar registrado y activo en la entidad.
2. **fechaHora:**
 - Debe estar en formato UTC ISO 8601.
3. **tipoTX:**
 - 0: Depósito a cuenta de ahorro.
 - 1: Pago a crédito.
 - 3: Depósito de cambio no entregado.
 - 4: Depósito a cuenta para retiro en sucursal.
4. **referencia:**
 - Debe ser un identificador válido que relacione esta transacción con una previa.
5. **cuenta:**
 - Validar que corresponda a una cuenta registrada en la entidad.
 - Para tipos 3 y 4, validar que se utilice la cuenta default o la cuenta ingresada por el cliente.
6. **secuencia:**
 - Número único generado por el cajero para evitar duplicidad en las transacciones.
7. **montoIngresado, montoDeposito, montoCambio:**
 - $\text{montoDeposito} \leq \text{montoIngresado}$.
 - Validar que montoCambio solo aplique para tipos 3 y 4.
8. **firma:**
 - Debe generarse correctamente utilizando el algoritmo especificado

3.2.- Ejemplo de solicitud en formato JSON RAW

```
{
  "idCajero": "CAJ123",
  "fechaHora": "2024-12-13T16:14:05.169Z",
  "tipoTX": 0,
  "referencia": "TX987654",
  "cuenta": "1234567890",
  "secuencia": 42,
  "montoIngresado": 2000,
  "montoDeposito": 1500,
  "montoCambio": 500,
  "firma": "ABC123XYZ456"
}
```

3.3.- Estructura de la Cadena a Validar para la Firma

IdCajero|FechaHora:rfc1123|TipoTX|Referencia|Cuenta|Secuencia|MontoIngresado:0.00|MontoDeposito:0.00|MontoCambio:0.00|_secret

3.3.1- Proceso firma

1. **Construcción de la Cadena:**
 - Los campos originales se concatenan en el formato especificado utilizando | como separador.
 - Al final de la cadena, se agrega el secreto compartido (|_secret).
2. **Codificación en UTF-8:**
 - La cadena completa (incluyendo el secreto) se convierte a un arreglo de bytes usando la codificación **UTF-8**.
3. **Firma con SHA256 y Padding PKCS#1:**
 - El hash de la cadena UTF-8 se firma utilizando:
 - Algoritmo: **RSA**.
 - Hash: **SHA256**.
 - Padding: **PKCS1**.
 - La firma generada se incluye en el campo firma de la solicitud enviada.
4. **Verificación del Hash en el Servicio:**
 - El servicio reconstruye la cadena utilizando los mismos datos recibidos y el secreto compartido (|_secret).
 - Se firma esta cadena reconstruida y se compara el hash generado contra el recibido en el campo firma.
5. **Resultado:**
 - Si los hashes coinciden, la transacción se aprueba.
 - Si no coinciden, se rechaza por inconsistencia.

3.4.- Ejemplo 1: Solicitud con tipoTX=0 y tipoTX=3

Solo cambia el valor del campo tipoTX entre las solicitudes (tipoTX=0 y tipoTX=3), las respuestas permanecen exactamente iguales, ya que el cambio en tipoTX no impacta en otros campos de la respuesta.

Solicitud

```
{
  "idCajero": "string",
  "fechaHora": "2022-04-11T14:25:04.720Z",
  "tipoTX": 0,
  "referencia": "",
  "cuenta": "508792",
  "secuencia": 0,
  "montoIngresado": 100,
  "montoDeposito": 100,
  "montoCambio": 0,
  "firma": "VpQgxRaoYqJdscGYrk4vgopZWDQDeTMJ4m28RQ5UMxAETKy4/Y6ql5ehq0S1ioluSYeq9OrBFyrSjP/0cJ49Km4097XEWDqME3eT/LUji5fDe5=="
}
```

Respuesta

```
{
  "descripcionCuenta": "Beneficiario:\n000751021 \nGA****VA*****CE*****",
  "monto": 100,
  "montoNoDepositado": 100,
  "fechaHora": "2022-04-11T14:25:57.8616401+00:00",
  "detalleComprobante": [
    {
      "columns": [
        {
          "aligment": "L",
          "decoration": null,
          "text": "DEPOSITO EN CAJERO string",
          "len": 1
        }
      ],
      "columnsLen": 1
    }
  ],
  "noAutorizacion": "42757128"
}
```

Ejemplo: Solicitud con tipoTX=1

Solicitud

```
{
  "idCajero": "string",
  "fechaHora": "2022-04-11T14:25:04.720Z",
  "tipoTX": 1,
  "referencia": "",
  "cuenta": "688211",
  "secuencia": 0,
  "montoIngresado": 100,
  "montoDeposito": 100,
  "montoCambio": 0,
  "firma":
  "of+MawK22ck795TL1JP6H0m/sDH5hcs2otnxo5K7ovuhqunLyVD4j9/6umxopnbQfB96AtLKfEO6Fs7tgcvEb4
  hSZ85WWr+qWVI=="
}
```

Respuesta

```
{
  "descripcionCuenta": "Beneficiario:\n688211 ACREINÓMINA PLATINO MXN.\nLE****AR*****UR****",
  "monto": 100,
  "montoNoDepositado": 0,
  "fechaHora": "2022-04-11T15:25:36.6532986+00:00",
  "detalleComprobante": [
    {
      "columns": [
        {
          "aligment": "L",
          "decoration": null,
          "text": "CAPITAL",
          "len": 3
        },
        {
          "aligment": "R",
          "decoration": null,
          "text": "$100.00",
          "len": 1
        }
      ],
      "columnsLen": 3
    }
  ],
  "noAutorizacion": "43415647"
}
```

Ejemplo: Solicitud con tipoTX=4

Solicitud

```
{
  "idCajero": "632546",
  "fechaHora": "2022-03-09T08:35:07.823Z",
  "tipoTX": 4,
  "referencia": "string",
  "cuenta": "string",
  "secuencia": 0,
  "montoIngresado": 0,
  "montoDeposito": 0,
  "montoCambio": 190,
  "firma":
  "a/ohwPGE0dc8P/vWcQX2iDrskX/q2iSoEESnZNBj/ysr3iwV26ewsK/OTcAqgeGJ4oLL761NGIhohbgMkaWyrJn
  asmTxByA07Q9FBEBoli0KqFQpKtx5tx2Tnrujonac489n56lfvzID0j4CZY9zCIEKcP7VirHrDrRHs5Cr/k98u8dqvQ
  PtWOnmwWTn9jvxpARY0C079E8plVCV8Jsc4aW8vCTK0STnVXmZTVZC2El2FaKKMnDz2u7gkZTu51CTwKEOY
  Xo3euWx/hDpa1jsdP6eHijOdIpljwsyC0k4GWbMvgdj5aoLhJk68CD5XozlPmSwzOd0pvgg8z67uLz30cg=="
}
```

Respuesta

```
{
  "descripcionCuenta": "RETIRO EN SUCURSAL",
  "monto": 0,
  "montoNoDepositado": 0,
  "fechaHora": "2022-04-06T21:39:43.2915807+00:00",
  "detalleComprobante": [
    {
      "columns": [
        {
          "aligment": "L",
          "decoration": null,
          "text": "CAJERO",
          "len": 1
        },
        {
          "aligment": "L",
          "decoration": null,
          "text": "632546",
          "len": 1
        }
      ],
      "columnsLen": 2
    }
  ],
}
```

```
{
  "columns": [
    {
      "aligment": "L",
      "decoration": null,
      "text": "ORDEN PAGO",
      "len": 1
    },
    {
      "aligment": "L",
      "decoration": null,
      "text": "259371",
      "len": 1
    }
  ],
  "columnsLen": 2
},
{
  "columns": [
    {
      "aligment": "L",
      "decoration": null,
      "text": "IMPORTANTE: CUENTA CON 30 DIAS PARA COBRAR ESTE COMPROBANTE EN CUALQUIERA DE
NUESTRAS SUCURSALES.",
      "len": 1
    }
  ],
  "columnsLen": 1
}
],
"noAutoizacion": "259371"
}
```

4.- Manejo de Excepciones Swagger

[Swagger UI](#)

5.- Conclusión

En conclusión, el flujo de integración entre el cajero automático y los servicios REST para la autenticación de usuarios, consulta de cuentas y la realización de transacciones, garantiza un proceso eficiente y seguro para la realización de depósitos y pagos. La implementación del estándar OAuth 2.0 para la autenticación mediante token, junto con los mecanismos de validación en cada etapa de la transacción, asegura la integridad de los datos y la correcta ejecución de las operaciones. Además, las respuestas detalladas proporcionadas por los servicios permiten una adecuada gestión de los procesos y un manejo transparente de las transacciones. Este enfoque no solo mejora la seguridad, sino que también optimiza la experiencia del usuario al realizar operaciones en el cajero automático.