

# Consulta Billetera

Autor: Henry Penuela.  
Fecha: 2025-10-23

Se expone la información necesaria y su descripción para poder realizar operaciones de consulta de saldo dentro de la billetera de Infomedia.

## Tipos de Cuenta

Al momento de inscribir un miembro en la billetera son creadas múltiples cuentas con el fin de facilitar la administración y uso de acuerdo a la transacción a realizar.

Código	Tipo	Descripción
8	Principal	Cuenta principal de un miembro. Refiere a la cuenta donde se realizan cargas y pagos para las acciones <b>"normales"</b> dentro de este sistema.
9	Manilla	Cuenta que refiere a las transacciones usando dispositivos con capacidad de transporte de información, como son tarjetas NFC o similares.
10	Comercio	Cuenta principal de un comercio. Refiere a la cuenta que recibe pagos.
11	Comisiones	Cuenta que recibe comisiones. Refiere a la cuenta que recibe las comisiones para vendedor o comercio, después de un cierre de operación.
12	Conciliación	Cuenta para recibir traslados, devoluciones y aquellas transacciones que tengan como objeto el <b>cuadre</b> de cuentas.
26	Temporal	Cuenta para uso en test, demostraciones y pruebas de concepto.

# Identificación de Miembro

La forma como un miembro es identificado dentro de este sistema.

Campo	Descripción
phoneNumber	Corresponde al numero de teléfono
phoneCode	Corresponde al código de Area
documentNumber	Corresponde al numero de documento
documentType	Corresponde al tipo de documento
memberType	Corresponde al tipo de miembro.
memberId	Identificación <b>única</b> dentro del sistema..

## Tipos de Documento

Campo	Descripción
citizen	Refiere a tipo de identificación de ciudadanía y entregada en forma exclusiva a mayores de edad.
passport	Refiere a tipo de documento de extranjería
company	Refiere a identificación de comercio.
licence	Tipo de identificación relacionado a permisos o identificación temporal.
under_age	Tipo de documento para menores de edad.
none	No relaciona a tipo de documento o es genérico.

## Tipos de Miembro

Campo	Descripción
member	Miembro Refiere a aquel que usa el sistema como cliente sin privilegios especiales.
owner	Propietario. Refiere a empresa cliente o responsable de operaciones.
admin	Administrador. Refiere a aquel con privilegios dentro del sistema.
supplier	Refiere a aquel con rol de proveedor.
deliver	Refiere a aquel con rol de transportador.
operator	Refiere a terceros con funciones de operación y/o mantenimiento.
store	Refiere a que corresponde a una tienda o comercio

# Consulta Saldo

Para realizar la consulta de un saldo tenga en cuenta que este se compone de:

- Tipo de cuenta a consultar.
- Identidad del miembro propietario de la cuenta.

Para la identificación del miembro, puede utilizar uno de los siguientes:

- Numero de Teléfono
- Numero de Documento
- ID único

## Usando Numero de Teléfono

```
{
  "phoneNumber": "123456789",
  "phoneCode": "+57",
  "memberType": "member", // Optional. Default: "member"
  "accountType": "8"      // Optional. "Principal" by default.
}
```

## Usando Numero de Documento

```
{
  "documentNumber": "123456789",
  "documentType": "citizen", // Optional. Default: "citizen"
  "memberType": "member",    // Optional. Default: "member"
  "accountType": "8"         // Optional. "Principal" by default.
}
```

## Usando ID único

```
{
  "memberId": "1234",
  "accountType": "8" // Optional. "Principal" by default.
}
```

# Requerimiento HTTP

- *path*: /wallet/balance
- *método*: POST
- *host*: https://iot.infomediasservice.com/iot\_stores

## Requerimiento

Usando uno de los expuestos.

## Respuesta

```
{
  "balance": 100.00,      // Balance for selected account type.
  "accountType": 8,      // Selected account type
  "accountName": "C02_00000100", // Account Name/Number
  "name": "Member's name",
  "lastName": "Member's last name",
  "phoneNumber": "123456789",
  "phoneCode": 2,
  "documentNumber": "123456789",
  "documentType": 2,
  "email": "member@mail",
  "memberId": 234,
  "phoneCodeName": "+57",
  "phoneCodeNick": "Colombia", // Used by frontend for "human" representation
  "documentTypeName": "CITIZEN",
  "documentTypeNick": "Cédula" // Used by frontend for "human" representation
}
```

## Error

```
{
  "error": "error description"
}
```

- member not found
- account type not found
- member account not found
- bad parameters

# Librería

Se dispone de una librería para el entorno `nodejs`, para el manejo de comunicaciones seguras en requerimientos de servidor a servidor.

## Instalación

La instalación de la librería usa `npm` como intermediario.

Dentro del directorio de su proyecto realice los siguientes pasos para la instalación:

```
npm i -D git+https://iot.infomediaservice.com/git/openbeer/iot-lib-server.git
```

### Dependencias

```
npm i -D js-yaml
```

O bien agregue al archivo `package.json` de su proyecto:

```
"devDependencies": {
  "@openbeer/iot-lib-server": "git+https://iot.infomediaservice.com/git/openbeer/iot-lib-server.git",
  "js-yaml": "^4.1.0"
}
```

# Ejemplo de Uso

Aqui un ejemplo de uso. Es importante que tenga en cuenta las definiciones de tipo de cuenta e identidad de usuario.

```
/** Http Secure Issuer
 * Used for Server to Server request.
 *
 * Example of use. */
import { HTTPSecureIssuer } from "@openbeer/iot-lib-server";

/** Define **host**
 *
 * A good practice is to define this as an environment variable
 * or in a configuration file outside the scope of your project.
 * @type {String} */
const IOT_HOST = "https://iot.infomediaservice.com/iot_stores";

/** **Secure Issuer for Server to Server requests**.
 *
 * It uses ***strong encryption*** to ensure end-to-end security
 *
 * For database "read" operations no public key is required.
 * @type {HTTPSecureIssuer} */
const issuer = new HTTPSecureIssuer(null, {
  host: IOT_HOST,
});

function main() {
  const fasync = async () => {
    let resp = await issuer.doRequest(
      "/wallet/balance", {
        body: {
          phoneNumber: "3138750391",
          phoneCode: "+57",
          accountType: "26",
        },
      },
    );
    let body = await resp.json();
    console.log(JSON.stringify(body, null, "  "));
    //-----
    // Closing is only necessary when the application ends.
    // As in this case.
    await issuer.close();
  };
  setImmediate(() => fasync());
}

main();
```

# Consideraciones.

---

De nada sirve tener conexiones seguras con la billetera, ***si usted no establece una política de seguridad en su aplicación.***

No olvide que de manejo de dinero se trata.