BANCO PLAZA

Tú cuentas

Referencia de Uso: API Banco Plaza

Clientes

Enero 2026

Versión 4.0.0

Tabla de Contenido

Datos de Dominio

Todos los recursos descritos en el presente documento son relativos al siguiente dominio:

URL para consumo API

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	openapiqa.bancoplaza.com	N/A
Productivo	HTTPS	openapi.bancoplaza.com	N/A

URL para generación de token de autenticación API

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	portalapiqa.bancoplaza.com/oauth2/token	N/A
Productivo	HTTPS	portalapi.bancoplaza.com/oauth2/token	N/A

Procedimiento para la autenticación de APIs con Oauth 2.0

OAuth 2.0 es un marco de autorización (Authorization Framework) basado en principios arquitectónicos y estándares de seguridad globales. Su objetivo principal es delegar el acceso a recursos protegidos sin la necesidad de compartir las credenciales directas del usuario. En el ecosistema de WSO2, el endpoint /oauth2/token actúa como la puerta central para la gestión de acceso. Su función es realizar el intercambio de credenciales de autorización (como códigos de autorización o credenciales de aplicación) por un Access Token válido.

Requisitos Obligatorios de la Petición

Para garantizar la integridad y seguridad del proceso, todas las solicitudes al endpoint /oauth2/token deben contemplar los valores del `client_id` y `client_secret` emitidos por el WSO2 APIM

Definición: CLIENT_ID

Es el identificador único que permite al servidor reconocer qué aplicación específica está solicitando el acceso.
Ejemplo: client_id

VgV9jtzb6ozPr_mqwlckJFR5C1oa

Definición: CLIENT_SECRET

Es la clave secreta o contraseña de la aplicación que verifica su identidad y asegura que solo entidades autorizadas puedan interactuar con la API.
Ejemplo: client_secret

FJ2lciJ3xXW7LhFa2f7oZ_HKNZsa

Definición: URL TOKEN

Es el recurso de red (punto de acceso) dedicado exclusivamente a la emisión de tokens de acceso. En el modelo de OAuth 2.0, este endpoint actúa como el servidor de autorización.

Características Principales:

- Método Obligatorio: Es estrictamente de tipo POST. Esto es así porque el cuerpo de la petición contiene credenciales sensibles y porque la operación no es "idempotente" (estás creando una sesión o recurso nuevo).
- Seguridad (SSL/TLS): Al manejar credenciales, este endpoint solo debe estar disponible a través de HTTPS.
- Procesamiento: Cuando la URL recibe la petición, el Key Manager de WSO2 realiza tres pasos internos:
- Valida la autenticidad del client_id y client_secret.
- Verifica que la aplicación esté activa.
- Genera un token firmado con un tiempo de expiración definido.

Procedimiento para la solicitud de token

Antes de consumir el endpoint, necesitas las credenciales emitidas por Banco Plaza.

- client_id: Tu identificador único
- client_secret: Tu contraseña de aplicación.

La solicitud se envía mediante un método POST a la URL del consumo del token. Este paso es el intercambio formal de identidad por acceso a continuación se describen la información de acuerdo con el ambiente.

URL de consumo del token

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	portalapiqa.bancoplaza.com/oauth2/token	N/A
Productivo	HTTPS	portalapi.bancoplaza.com/oauth2/token	N/A

Para realizar el proceso de consumo de las APIs se requiere la emisión del token, para ello se ha incorporado un script basado en el concepto de encadenamiento de peticiones, su funcionalidad es obtener un token y posteriormente enviarlo como parámetro al header: Authorization.

A continuación, se mencionan los parámetros a considerar remplazar su valor para consumo de las APIS en un código de ejemplo en Javascript y la explicación de este

Los valores que se deben de cambiar son:

tokenUrl = Url del token

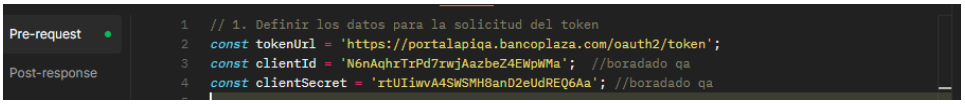
Ejemplo: https://portalapiqa.bancoplaza.com/oauth2/token

clientId = valor del client_id

Ejemplo: N6nAqhrTrPd7rwjAazbeZ4EWpWMa

clientSecret = valor del client_secret

Ejemplo: rtUIiwvA4SWSMH8anD2eUdREQ6Aa



Nota: cada uno de los recursos se deberá hacer el remplazo de la información que anteriormente se menciona

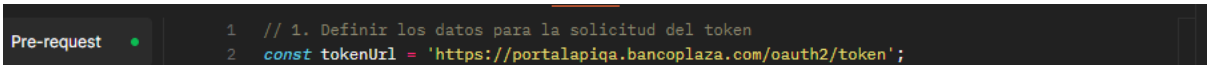
La función principal del script es mostrar el proceso a seguir para automatizar la obtención de un token de acceso (OAuth 2.0) antes de ejecutar la petición principal de la colección.

A continuación, se muestra el desglose paso a paso:

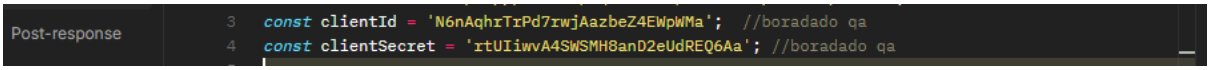
1. Configuración de Credenciales

   Se definen las variables necesarias para identificarse ante el servidor de Banco Plaza:

- **URL del Token:** La dirección del servidor de autorización (tokenUrl).



- **Credenciales:** Define el clientId y el clientSecret.



- **Codificación:** Convierte estas credenciales a un formato **Base64** usando la función btoa(). Esto es un requisito estándar para el encabezado de "Authorization: Basic".

2. Definición del Objeto de Petición (requestOptions)

Se construye la estructura de la solicitud HTTP que se enviará:

- Método: POST.
- Headers: Configura el tipo de contenido como application/x-www-form-urlencoded e incluye el token de autorización básica que se generó en el paso anterior.
- Cuerpo (Body): Define que el flujo de autenticación es de tipo client_credentials, que es el usado para comunicación de servidor a servidor (sin intervención de un usuario final).

3. Ejecución de la Petición (pm.sendRequest)

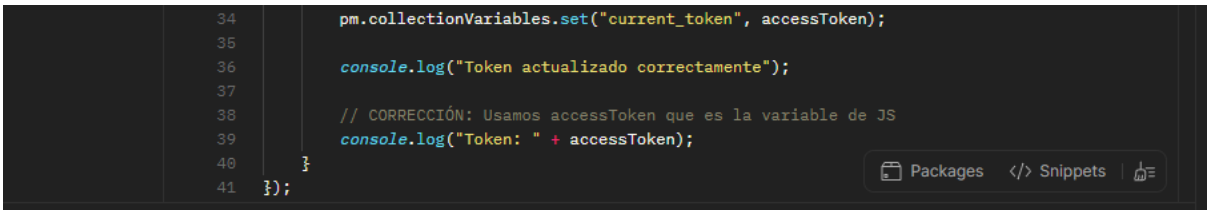
El script envía la solicitud de forma asíncrona:

- Manejo de Errores: Si algo sale mal (problemas de red, URL incorrecta), imprime el error en la consola.
- Procesamiento de Respuesta: Si la petición es exitosa, convierte la respuesta del servidor en un objeto JSON para poder leerla.

4. Extracción y Almacenamiento del Token

Esta es la parte más importante para el flujo de trabajo:

- Extracción: Toma el valor del campo access_token que devolvió el banco.
- Persistencia: Guarda ese valor en una variable de colección llamada current_token.
- Log: Imprime el token en la consola para que el desarrollador pueda verificar que se recibió correctamente.



Resumen de Operaciones

Clientes

Requiere autenticación	Endpoint	Método HTTP	Permisos
Sí	/clientes/v1/v0/clientes/{id}	HEAD	READ
Sí	/clientes/v1/v0/clientes/{id}	GET	READ

Recurso: `/clientes/v1/v0/clientes/{id}`

Argumentos de Entrada

	Argumento	Tipo	Descripción
Argumentos Requeridos	`id`	`String`	*Identificación*. Se refiere al documento de identidad del **cliente** que se desea consultar. Debe incluir el tipo de persona. Por ejemplo, para un persona natural el valor sería `V13759368`. En contraste, para un comercio sería `J00378944781`. Longitud de doce (`12`) caracteres.
Argumentos Opcionales	`telefono`	`String`	*Teléfono*. Se refiere al número telefónico que el cliente posee registrado en la plataforma de Banco Plaza.
	`email`	`String`	E-mail. Se refiere al correo electrónico que el cliente posee registrado en la plataforma de Banco Plaza.
Permisos	`READ`		
Método HTTP	`HEAD`		

HTTP Status Code			`204 No Content`
			`400 Bad Request`
			`401 Unauthorized`
			`404 Not found`
			`500 Internal Server Error`

Descripción

Permite verificar la existencia de un cliente Banco Plaza haciendo uso de su documento de identificación.

A continuación, un ejemplo de la petición:

`GET /clientes/v1/v0/clientes/V0016411999 HTTP/1.1`

`Request Header`

```
Content-Type: application/json
Authorization: Bearer {{current_token}}
```

Estructura de Salida

Se refiere a los datos que componen la estructura de salida para el recurso. Básicamente, es la respuesta a la petición. Se describe en la siguiente tabla:

Campo	Tipo	Descripción
codigoRespuesta	String	*Código de Respuesta*. Se refiere a un código de cuatro (4) dígitos que identifica la respuesta generada por el servicio. **Este campo será agregado a la cabecera de la respuesta HTTP.**
descripcionCliente	String	*Descripción para el Cliente*. Es la descripción asociada al *Código de Respuesta* diseñada para ser mostrada o desplegada a la capa front. **Este campo será agregado a la cabecera de la respuesta HTTP.**
descripcionSistema	String	*Descripción del Sistema*. Es la descripción asociada al Código de Respuesta diseñada para albergar información técnica sobre la respuesta a la petición. **Este campo será agregado a la cabecera de la respuesta HTTP.**
fechaHora	Datetime	*Fecha y Hora*. Es una marca de tiempo o timestamp del momento exacto en que se entrega la respuesta a la petición. **Este campo será agregado a la cabecera de la respuesta HTTP.**

A continuación, se muestra un ejemplo de una respuesta para este recurso:

```
Response Header
HTTP/1.1 204 No Content
```

```
codigoRespuesta: 0000
descripcionCliente: Transaccion Exitosa
descripcionSistema: Transaccion Exitosa
fechaHora: 2026-01-28 09:12:45
```

Códigos de Respuesta

La lista de los posibles Códigos de Respuesta se presenta a continuación:

Código de Respuesta	Descripción Sistema	Tipo de Resultado	HTTP Status Code
0000	TRANSACCIÓN EXITOSA	Transacción exitosa.	204
0002	PARÁMETRO [parametro] OBLIGATORIO	Validación de sistema	400
0003	PARÁMETRO [parametro] NO CUMPLE FORMATO	Validación de sistema	400
0010	CLIENTE NO EXISTE	Validación de sistema	404
0011	TELÉFONO NO COINCIDE	Validación de sistema	400
0012	EMAIL NO COINCIDE	Validación de sistema	400
9990	SERVICIO NO DISPONIBLE	Validación de sistema	400
9999	ERROR INTERNO	Error de sistema	500
A001	FIRMA DIGITAL INVÁLIDA	Validación de sistema	400
A002	FIRMA DIGITAL VENCIDA	Validación de sistema	400
A003	RECURSO NO AUTORIZADO	Validación de sistema	401
A004	API-KEY INVÁLIDA O REVOCADA	Validación de sistema	401

Recurso: `/clientes/v1/v0/clientes/{id}`

Argumentos de Entrada

	Argumento	Tipo	Descripción
Argumentos Requeridos	`id`	`String`	*Identificación*. Se refiere al documento de identidad del **cliente** que se desea consultar. Debe incluir el tipo de persona. Por ejemplo, para persona natural el valor sería `V13759368`. En contraste, para un comercio sería `J00378944781`. Longitud de doce (`12`) caracteres.
Argumentos Opcionales	`telefono`	`String`	*Teléfono*. Se refiere al número telefónico que el cliente posee registrado en la plataforma de Banco Plaza.
	`email`	`String`	E-mail. Se refiere al correo electrónico que el cliente posee registrado en la plataforma de Banco Plaza.
Permisos	`READ`		
Método HTTP	`GET`		

HTTP Status Code			`200 OK`
			`400 Bad Request`
			`401 Unauthorized`
			`404 Not found`
			`500 Internal Server Error`

Descripción

Permite recuperar la información de un cliente a partir del documento de identificación indicado en `{id}`.

A continuación, un ejemplo de la petición:

`GET /v0/clientes/V0016411999 HTTP/1.1`

Request Header

```
Content-Type: application/json
Authorization: Bearer {{current_token}}
```

Estructura de Salida

Se refiere a los datos que componen la estructura de salida para el recurso. Básicamente, es la respuesta a la petición. Se describe en la siguiente tabla:

Campo	Tipo	Descripción
codigoRespuesta	String	*Código de Respuesta*. Se refiere a un código de cuatro (4) dígitos que identifica la respuesta generada por el servicio. **Este campo será agregado a la cabecera de la respuesta HTTP.**
descripcionCliente	String	*Descripción para el Cliente*. Es la descripción asociada al *Código de Respuesta* diseñada para ser mostrada o desplegada a la capa front. **Este campo será agregado a la cabecera de la respuesta HTTP.**
descripcionSistema	String	*Descripción del Sistema*. Es la descripción asociada al Código de Respuesta diseñada para albergar información técnica sobre la respuesta a la petición. **Este campo será agregado a la cabecera de la respuesta HTTP.**
fechaHora	Datetime	*Fecha y Hora*. Es una marca de tiempo o timestamp del momento exacto en que se entrega la respuesta a la petición. **Este campo será agregado a la cabecera de la respuesta HTTP.**
id	String	*Identificación*. Se refiere al documento de identidad del **cliente** que fue indicado como argumento de entrada.
nombreCompleto	String	*Nombre Completo.* Se refiere a los nombres y a apellidos del cliente consultado.

Un ejemplo de cómo luce una respuesta es el siguiente:

```
Response Header
HTTP/1.1 200 OK
```

```
codigoRespuesta: 0000
descripcionCliente: Transaccion Exitosa
descripcionSistema: Transaccion Exitosa
fechaHora: 2026-01-28 09:12:45
```

Body


```
{
        "id": "V0016411999"
      , "nombreCompleto": "JAVIER ENRIQUE, MARAVER SALAS"
}
```


Códigos de Respuesta

Los códigos de respuesta son exactamente los mismos que los que se presentan en el recurso `/clientes/v1v0/clientes/{id}`, con la salvedad que el código de estatus `HTTP` para las respuestas exitosas es `200` y no `204`.