AZUL | SERVICIOS DIGITALES POPULAR

Link de Pagos con Integración vía API

ÍNDICE

Link de Pagos AZUL con Integración vía API 3
Comercios sin Integración vía API 3
Conexión VPN Site-to-Site 4
Mutual Authentication (Autenticación mutua con Certificados Digitales) 6
Cómo invocar la Integración vía API 7
Autenticación Integración vía API 8
Comercios con Integración vía API 9
Transacciones para crear Link de Pago 9
Transacción para consultar Link de Pago 11

Link de Pagos AZUL con Integración vía API

Link de Pagos es una solución versátil que te permitirá realizar ventas y recibir pagos a través de un link con el cual tus clientes realizarán su pago móvil, esta funcionalidad ya se encuentra disponible para integrar a la plataforma de tu empresa a través de Integración vía API.

La Integración vía API permite procesar transacciones con tarjetas VISA, MasterCard, American Express, Discover y Diners. La implementación de esta integración es totalmente independiente de la plataforma tecnológica que tu empresa haya utilizado en sus sistemas. El comercio afiliado puede desarrollar una aplicación en cualquier lenguaje de programación que soporte integración vía API (ASP, .NET, PHP, Java, etc.). Es decir, que sin importar el tipo de servidores donde corra la aplicación de tu empresa o el lenguaje de programación o las tecnologías que se utilizaron para desarrollar, los sistemas pueden ser integrados a la plataforma de pagos de AZUL a través del consumo de la integración.

La Integración vía API está publicada en formato JSON, HTTPS Post Form y SOAP para la integración de tus aplicaciones con nuestra plataforma de pagos. La elección del formato a implementar depende del afiliado.

Comercios sin Integración vía API

Para disponer del método «**PaymentLinkCreate**» el comercio deberá integrarse a nuestra plataforma de Integración vía API, para esto debe cumplir con los siguientes procesos:

1. Establecer una conexión segura con nuestro ambiente desarrollo por medio de VPN o utilizando un certificado digital.
2. Iniciar desarrollo de integración vía API.
3. Certificación de la integración.
4. Recepción de accesos de producción para utilizar el método «**PaymentLinkCreate**».

Producción	
Ambiente	URL
SOAP	https://pagos.azul.com.do/webservices/SOAP/Default.asmx
JSON	https://pagos.azul.com.do/webservices/JSON/Default.aspx
HTTP	https://pagos.azul.com.do/webservices/HTTP/Default.aspx

Sitio alterno	
Ambiente	URL
SOAP	https://contpagos.azul.com.do/Webservices/SOAP/default.asmx
JSON	https://contpagos.azul.com.do/Webservices/JSON/default.aspx
HTTP	https://contpagos.azul.com.do/Webservices/HTTP/default.aspx

Ambiente de pruebas	
Ambiente	URL
SOAP	https://pruebas.azul.com.do/webservices/SOAP/Default.asmx
JSON	https://pruebas.azul.com.do/webservices/JSON/Default.aspx
HTTP	https://pruebas.azul.com.do/webservices/HTTP/Default.aspx

El desarrollo debe estar preparado para intentar por la URL del sitio alterno en caso de fallo de la URL principal.

En la integración vía API, la captura de datos para realizar la transacción se realiza en el sitio web o sistema del comercio.

Para brindar la seguridad necesaria en la captura y envío de los datos hacia AZUL, el comercio deberá tener un tipo de autentificación para el formulario de tarjetas e implementar la conexión TLS 1.2.

Antes de empezar el proceso de integración/implementación, se debe establecer una comunicación segura y confiable entre el site del afiliado y el site de la plataforma de pagos de AZUL; para lo cual debe de elegirse entre:

- VPN Site-to-Site
- Autenticación Mutual con Certificados Digitales emitidos localmente por AZUL

Esta configuración debe realizarse tanto para el ambiente de pruebas como para el ambiente de producción; es un requisito para AZUL proveerles los factores de autenticación indicados en la sección Autenticación Integración vía API.

Conexión VPN Site-to-Site

Una conexión VPN Site-to-Site permite conectar de manera segura dos redes LAN, a través de internet; permitiendo que la información viaje por un "túnel" seguro (cifrado) para prevenir intercepción de las transacciones.

Para establecer este tipo de conexión debe realizarse configuración en ambos puntos que se desean conectar. Los requerimientos son:

- El cliente debe contar con un equipo que tenga conexión a Internet y pueda ser configurado para el uso de túneles IPSec (Router, Firewall, etc)
- El Pre-share Key será comunicado vía telefónica al cliente
- El cliente debe completar el siguiente formulario con su información y enviarlo a AZUL.

Información de Contacto		Cliente	AZUL
PRIMARIO	Nombre		
	Dirección de correo		
	Teléfono		
	Celular		
SECUNDARIO	Nombre		
	Dirección de correo		
	Teléfono		
	Celular		

Información de los Sites:

Información Peer	Cliente	AZUL
Dirección IP Site Producción		191.97.90.4
Dirección IP Host Producción Puerto		
Dirección IP Site Contingencia		201.221.126.193
Dirección IP Host Contingencia-Puerto		

Propiedades del Túnel

Propiedades del Túnel		Dispositivo Cliente	Dispositivo AZUL
PHASE 1	Método de Autenticación		Pre Share Key
	Esquema de encriptación		IKE
	Grupo Diffie-Hellman		Group2
	Algoritmo de Encriptación		3DES
	Algoritmo de Hashing		SHA1
	Lifetime (para renegociación)		86400 seconds
PHASE 2	Encapsulación		ESP
	Algoritmo de Encriptación		3DES
	Algoritmo de Autenticación		SHA1
	Perfect Forward Secrecy		DESHABILITADO
	Lifetime (para renegociación)		3600 seg. (1 HORA)

Mutual Authentication (Autenticación mutua con Certificados Digitales)

Para la implementación de una comunicación segura con autenticación mutual, el afiliado ha de presentar un Cliente Certificate[1] emitido por AZUL (no tiene costo). Las especificaciones con las cuales debe ser remitido el archivo CSR correspondiente son las siguientes:

Ambiente	URL
Producción	
Common Name*	Nombrecomercio.local (Ejemplo: tiendaenlinea.local)
Organization Name**	Servicios Digitales Popular
Organization Unit Name**	Producción
Country Name**	DO
State or Province**	Santo Domingo
Locality Name**	Distrito Nacional
Level Encryption**	4096

[1] AZUL emitirá certificados diferentes para cada uno de los ambientes (Pruebas y Producción)

Pruebas	
Common Name*	Nombrepruebas.local
Organization Name**	Servicios Digitales Popular
Organization Unit Name**	Desarrollo
Country Name**	DO
State or Province**	Santo Domingo
Locality Name**	4096 bits
Level Encryption**	

* Debes indicar la URL de tu negocio (según el ambiente que se trate)

** Los archivos CSR deben llegar con estas informaciones, los valores son fijos sin importar el país o provincia en que se encuentren.

Notas importantes:

- Los datos de pruebas y de producción nunca van a ser similares
- Los factores de autenticación de prueba y producción nunca van a ser los mismos
- Las URL de prueba y las de producción no son las mismas

Cómo invocar la Integración vía API

El consumo del servicio es igual a cualquier request POST. A continuación, puedes ver un ejemplo básico en C#. Ten en cuenta que debes incluir los headers «**ContentType**» apropiados según el canal y aplicar manejo de errores al request:

```
string wsMessageIn = "MENSAJE_A_ENVIAR";     string wsMessageOut = "";


HttpWebRequest webReq = (System.Net.HttpWebRequest)WebRequest.Create(UrlToQuery);
webReq.Method = "POST"; webReq.ContentType = "application/json";

webReq.Headers.Add("Auth1", "factor1");
webReq.Headers.Add("Auth2", "factor2");

StreamWriter bodyStream = new StreamWriter(webReq.GetRequestStream());
bodyStream.Write(wsMessageIn); bodyStream.Close();


HttpWebResponse webResp = (System.Net.HttpWebResponse)webReq.GetResponse();


if (webResp != null && webReq.HaveResponse)
{
        StreamReader receiveStream = new StreamReader(webResp.GetResponseStream());
        wsMessageOut = receiveStream.ReadToEnd();   receiveStream.Close();
        webResp.Close();
}
```

Para el servicio SOAP puedes consultar el WSDL directamente para formar el componente de acceso o si lo deseas puedes tomar el mensaje de requerimiento directamente de la sección Ejemplos de este documento y construir el requerimiento manualmente.

Debes aplicar el «**ContenType**» de la llamada según el tipo que vayas a utilizar:

- SOAP: "text/xml"
- JSON: "application/json"
- HTTP Forms: "application/x-www-form-urlencoded"

Autenticación Integración vía API

Los 3 tipos de servicios requieren ser provistos de dos valores de autenticación, los cuales serán entregados por AZUL durante el proceso de afiliación al servicio. El funcionamiento es similar al de un usuario/password.

Existen dos formas en que puedes enviar estos dos parámetros, dependiendo del tipo de servicio que estés utilizando y tu preferencia.

HTTP Headers:

Este método es el único disponible para JSON y HTTP Forms (también está disponible para SOAP). En este formato, los campos son enviados en el header del POST HTTP. Para esto debes incluir dos headers, uno llamado Auth1 y otro llamado Auth2:

```
POST /url_definida
ContentType: ....
Auth1: FACTOR#1
Auth2: FACTOR#2
...... ......
```

SOAP Headers:

Para el consumo vía SOAP puedes optar por no enviar los headers HTTP, sino utilizar el «**SOAPAuthHeader**» en el cuerpo del mensaje. Si utilizas una herramienta WSDL para generar la llamada a la integración, o un proxy/cliente para consumirlo, es muy probable que estos ya estén adicionados a su mensaje/componente. En todo caso, el servicio SOAP es capaz de detectar la presencia de los valores de autenticación en el header HTTP o en el header SOAP de manera automática y efectuar la autenticación:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SOAPAuthHeader xmlns="http://Merit/AzulIS/TransactionServices/">
       <Auth1>factor1</Auth1>
       <Auth2>factor2</Auth2>
    </SOAPAuthHeader>
  </soap:Header>
  <soap:Body>
    <ProcessPayment xmlns="http://Merit/AzulIS/TransactionServices/">
       <ProcessPaymentRequest>
         ....
       </ProcessPaymentRequest>
    </ProcessPayment>
  </soap:Body>
</soap:Envelope>
```

Los factores de autenticación para el ambiente de pruebas son diferentes a los que se les proveerá en el ambiente de producción[2].

Comercios con Integración vía API

Para los comercios que ya posean acceso a la Integración vía API en producción, se les estará enviando un nuevo acceso de producción (Auth1/Auth2), solo para utilizar el método «**PaymentLinkCreate**».

Una vez reciban este nuevo acceso y coloquen el mismo, podrán utilizar el método de producción.

Transacciones para crear Link de Pago

Campos de interés	
Parámetro	Descripción
ResponseCode	Estatus de la creación. "CREATED" cuando fue creado exitosamente o "NOT_CREATED" cuando existe un error/validación.
ErrorDescription	Mensaje de error/validación.
LinkId	ID del link de pago creado. Puede ser utilizado en el endpoint «PaymentLinkInfo» para buscar la información sobre el mismo.
LinkFullUrl	URL completa del link de pago.
ExpirationDate	Fecha de expiración del link.

[2] Los factores de acceso al ambiente de producción serán provistos luego de haberse certificado las pruebas en el ambiente de pruebas (ver sección: Pasos para Aceptar Pagos en tu web).

SOAP:

POST: https://server/WebServices/SOAP/default.asmx

SOAPAction: "http://Merit/AzulIS/TransactionServices/PaymentLinkCreate"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SOAPAuthHeader xmlns="http://Merit/AzulIS/TransactionServices/">
       <Auth1>testtv</Auth1>
       <Auth2>testtv</Auth2>
    </SOAPAuthHeader>
  </soap:Header>
  <soap:Body>
    <PaymentLinkCreate xmlns="http://Merit/AzulIS/TransactionServices/">
       <PaymentLinkCreateRequest>
         <Store>99999991</Store>
         <OrderNumber>WS1234</OrderNumber>
         <Amount>20123</Amount>
         <Itbis>102</Itbis>
         <Language>ES</Language>
         <ClientContractNumber>Client de tal</ClientContractNumber>
       </PaymentLinkCreateRequest>
    </PaymentLinkCreate>
  </soap:Body>
</soap:Envelope>
```

Respuesta:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SOAPAuthHeader xmlns="http://Merit/AzulIS/TransactionServices/">
       <Auth1>testtv</Auth1>
       <Auth2>testtv</Auth2>
    </SOAPAuthHeader>
  </soap:Header>
  <soap:Body>
    <PaymentLinkCreateResponse xmlns="http://Merit/AzulIS/TransactionServices/">
       <PaymentLinkCreateResult>
         <ResponseCode>CREATED</ResponseCode>
         <ErrorDescription/>
         <LinkId>b74ca034</LinkId>
         <LinkFullUrl>https://pagos.azul.com.do/b74ca034</LinkFullUrl>
         <ExpirationDate>2021-03-24T10:04:10.9240172-04:00</ExpirationDate>
       </PaymentLinkCreateResult>
</PaymentLinkCreateResponse>
</soap:Body>
</soap:Envelope>
```

JSON:

POST: https://pruebas.azul.com.do/Webservices/JSON/Default.aspx?paymentlinkcreate

```
{
        "Store":"99999991",
        "OrderNumber":"W12345S",
        "Amount":"20123",
        "Itbis":"102",
        "Language":"ES",
        "ClientContractNumber":"Nombre Cliente"
}
```

Respuesta:

```
{
        "ErrorDescription":"",
        "ExpirationDate":"\/Date(1616099927158-0400)\/",
        "LinkFullUrl":"https:\/\/pagos.azul.com.do\/3203042c"
        "LinkId":"3203042c","ResponseCode":"CREATED"
}
```

Transacción para consultar Link de Pago

Campos de interés	
Parámetro	Descripción
Found	Indica si fue encontrado un link de pago (true/false).
OriginalTransactionType	Tipo de transacción original (Sale/Hold).
ExpirationDate	Fecha de expiración del link en formato YYYYMMDDHHMMSS.
Status	Estatus del link (Generado/Abierto/Expirado/Procesado/Cancelado/Anulada).
Attempts	# intentos de pago.
Clicked	Si fue abierto por el usuario (true/false). Si es true, se mostrará la fecha en «ClickedDate».
Payed	Si fue pagado exitosamente por el usuario (true/false). Si es true, se mostrará la fecha en «PayedDate», al igual que se llenarán los campos «PayedCardNumber», «PayedAuthorizationNumber», «PayedRRN», «PayerdAzulOrderId».
Posted	Si fue postead/aplicado el pago (true/false). Aplica solamente para transacciones Hold. Si es true, se mostrará la fecha en «PostedDate» y el campo «PostedAzulOrderId» vendrá llenado.

SOAP:

SOAPAction: "http://Merit/AzulIS/TransactionServices/PaymentLinkInfo"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SOAPAuthHeader xmlns="http://Merit/AzulIS/TransactionServices/">
       <Auth1>testtv</Auth1>
       <Auth2>testtv</Auth2>
    </SOAPAuthHeader>
  </soap:Header>
  <soap:Body>
    <PaymentLinkInfo xmlns="http://Merit/AzulIS/TransactionServices/">
       <PaymentLinkInfoRequest>
         <Store>99999991</Store>
         <LinkId>a6852849</LinkId>
       </PaymentLinkInfoRequest>
    </PaymentLinkInfo>
  </soap:Body>
</soap:Envelope>
```

Respuesta

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <SOAPAuthHeader xmlns="http://Merit/AzulIS/TransactionServices/">
       <Auth1>testtv</Auth1>
       <Auth2>testtv</Auth2>
    </SOAPAuthHeader>
  </soap:Header>
  <soap:Body>
    <PaymentLinkInfoResponse xmlns="http://Merit/AzulIS/TransactionServices/">
       <PaymentLinkInfoResult>
         <Found>true</Found>
         <LinkId>a6852849</LinkId>
         <ResponseCode>FOUND</ResponseCode>
         <ErrorDescription/>
         <OriginalTransactionType>Sale</OriginalTransactionType>
         <MerchantId>99999991</MerchantId>
         <OrderNumber>12</OrderNumber>
         <Amount>12300</Amount>
         <Itbis>2300</Itbis>
         <Currency>$</Currency>
         <ClientContractNumber/>
         <ExpirationDate>20200714220450</ExpirationDate>
         <Status>Expirado</Status>
         <Attempts>0</Attempts>
         <Clicked>true</Clicked>
         <ClickedDate>20200707222038</ClickedDate>
         <Payed>false</Payed>
         <PayedDate/>
         <PayedCardNumber/>
         <PayedAuthorizationNumber/>
         <PayedRRN/>
         <PayedAzulOrderId>0</PayedAzulOrderId>
         <Posted>false</Posted>
         <PostedDate/>
         <PostedAzulOrderId/>
       </PaymentLinkInfoResult>
    </PaymentLinkInfoResponse>
  </soap:Body>
</soap:Envelope>
```

JSON:

POST: https://pruebas.azul.com.do/Webservices/JSON/Default.aspx?PaymentLinkInfo

```
{
       "Store":"99999991",
       "OrderNumber":"3203042c"
}
```

Respuesta:


```
{
        "Amount":"20123",
        "Attempts":"0",
        "Clicked":false,
        "ClickedDate":"",
        "ClientContractNumber":"Client de tal b",
        "Currency":"$",
        "ErrorDescription":"",
        "ExpirationDate":"20210318163847",
        "Found":true,
        "Itbis":"102",
        "LinkId":"3203042c",
        "MerchantId":"390385400050",
        "OrderNumber":"W12345S",
        "OriginalTransactionType":"Sale",
        "Payed":false,
        "PayedAuthorizationNumber":"",
        "PayedAzulOrderId":"0",
        "PayedCardNumber":"",
        "PayedRRN":"",
        "Posted":false,
        "PostedAzulOrderId":"",
        "PostedDate":"",
        "ResponseCode":"FOUND",
        "Status":"Generado"
}
```