



GOBIERNO DE LA  
CIUDAD DE MÉXICO

AGENCIA DIGITAL DE  
INNOVACIÓN PÚBLICA

Manual para creación del cliente del Servicio Web  
**Servicio Web del Panel de tianguis digital del Gobierno de la Ciudad de  
México**

Fecha: 09/11/2021



Control de versiones	3
Objetivo del documento	3
Información general	4
Objetivo del Servicio Web	4
Disponibilidad del servicio	4
Ambientes	4
Solicitud de acceso al servicio web	4
Consumo del Servicio Web	4
Seguridad del Servicio Web	4
Datos del servicio:	5
Parámetros de entrada:	5
Parámetros de salida:	5
Manejo de errores	6

---



## Control de versiones

Versión	Fecha	Descripción del cambio	Responsable
1.0	09/11/2021	Versión inicial del documento	
1.1	09/11/2021	Se agregan validaciones	

## Objetivo del documento

Contar con una guía para que los desarrolladores puedan implementar un cliente para el consumo del Servicio Web para el acceso (login) a la información del proveedor registrado en el Sistema denominado Panel de tianguis digital.



## Información general

### Objetivo del Servicio Web

Permitir el acceso (login) de la información de la cuenta creada en panel de tianguis digital.

### Disponibilidad del servicio

24 hrs, los 365 días del año.

En caso de mantenimiento, la Agencia digital de innovación pública notificará a la persona designada como responsable de las credenciales de acceso al servicio (Mesa de servicio).

Se sugiere contar con un mecanismo que guarde las peticiones que no se puedan realizar por algún evento inesperado, para que se puedan reenviar una vez que se restablezca el servicio.

### Ambientes

Para el consumo del servicio se cuenta con los ambientes de:

#### a) Desarrollo

Este ambiente tiene como fin realizar pruebas de funcionalidad e integración.

#### b) Producción

Este ambiente solo recibe peticiones de ambientes operativos. No se deben realizar ningún tipo de pruebas en este ambiente. Se dará el acceso una vez que se realicen las pruebas pertinentes en el ambiente de desarrollo.

### Solicitud de acceso al servicio web

#### a) Desarrollo

Para solicitar las credenciales de acceso al ambiente de desarrollo, se debe contactar con el equipo responsable de la Agencia digital de Innovación Pública de la Ciudad de México, en particular con la Dirección General de Operación Tecnológica <raul.soto@cdmx.gob.mx>.

#### b) Producción

Para solicitar las credenciales de acceso al ambiente de producción, se debe enviar un oficio a la Dirección General de Tecnologías y Comunicaciones de la Secretaría de Administración y Finanzas incluyendo , la justificación del uso del servicio, así como el número de peticiones que estima realizar mensualmente.



## Consumo del Servicio Web

### Seguridad del Servicio Web

El método de autenticación es mediante Bearer Token, para lo cual se entrega un usuario y contraseña, los cuales deberán resguardarse por el usuario responsable.

Para el uso del servicio y el envío de las credenciales se debe realizar mediante un canal seguro (https).

Así mismo, se debe garantizar que no se expongan las credenciales, por ejemplo no se deben realizar solicitudes desde el front/view sino desde el backend.

### Datos del servicio:

URI:	{URL}/v1/oauth/token/
Método:	POST
Headers:	multipart/form-data
Paso de parámetros	Los parámetros se envían vía form-data
Body:	Los parámetros envían como FORM-DATA, esto imita a un formulario

### Parámetros de entrada:

Parámetro	Descripción	Tipo de dato	Obligatorio
grant_type	Se recibe el texto "password"	string	Sí
username	Se recibe Registro Federal de contribuyentes (R.F.C.) de acuerdo a las reglas de operación bajo el esquemas REGEX: <pre>^[A-Z,Ñ,&amp;]{3,4}([0-9]{2})(0[1-9] 1[0-2])(0[1-9] 1[0-9] 2[0-9] 3[0-1])([A-Z \d]{3})\$</pre>	string	Sí
password	Se recibe la contraseña del usuario que previamente obtuvo su registro en dicho portal de panel de tianguis digital. Nota: "sin encriptar".	string	Sí
client_id	Dicho ID es generado mediante el sistema de encriptado de python utilizando la librería "oauthlib.common.py" de 40 caracteres de tipo UNICODE_ASCII_CHARACTER_SET = ('abcdefghijklmnopqrstuvwxyzABCDEFGH IJKLMNOPQRSTUVWXYZ0123456789'):	string	Sí



client_secret	Dicho ID es generado mediante el sistema de encriptado de python utilizando la librería "oauthlib.common.py" de 128 caracteres de tipo UNICODE_ASCII_CHARACTER_SET = ('abcdefghijklmnopqrstuvwxyzABCDEFGH IJKLMNOPQRSTUVWXYZ0123456789');	string	Sí
---------------	---	--------	----

**Parámetros de salida:**

Parámetro	Descripción	Tipo de dato
codigo_resultado	Código de resultado que indica la respuesta del servicio para el acceso a la información.  200 = OK { "access_token": "mye28gum2b4iUBb7RCYwtojvL90Y6n", "expires_in": 3600, "token_type": "Bearer", "scope": "read", "refresh_token": "8M3RC6h0DdeZjcZjbVJGkgzV1zYTrO", "refresh_token_expires_in": 604800 }	json
codigo_resultado	Código de resultado que indica si se creó o no la cuenta de usuario.  401 = Unauthorized { "code": "401", "message": "invalid_grant Invalid credentials given." }	json
codigo_resultado	Código de resultado que indica si se creó o no la cuenta de usuario.  401 = Unauthorized { "code": "401", "message": "invalid_client " }	json



codigo_resultado	Código de resultado que indica si se creó o no la cuenta de usuario.  401 = Unauthorized { "code": "401", "message": "unsupported_grant_type " }	json
mensaje	Mensaje asociado al código de respuesta	String

### Manejo de errores

En caso de un error inesperado, el servicio responderá con un HTTP 500.

### Datos del servicio:

URI:	{URL}/v1/profile
Método:	GET
Headers:	Authorization: Bearer `{TOKEN}`
Paso de parámetros	Los parámetros se vía Headers
Body:	Los parámetros envían Headers, este header se envía para detectar desde el backend la sesión del usuario y obtener la información requerida.

### Parámetros de entrada:

Parámetro	Descripción	Tipo de dato	de	Obligatorio
-----------	-------------	--------------	----	-------------

### Parámetros de salida:

Parámetro	Descripción	Tipo de dato
codigo_resultado	Código de resultado que indica la respuesta del servicio para el acceso a la información.  200 = OK { "full_name": "INVITADO INVITADO INVITADO", "rfc": "AAAA010101010", "email_oficial": "invitado@cdmx.gob.mx", "entity": 2, "is_staff": true, "is_active": true, "is_saf": false, }	json



	<pre>"is_superuser": true, "is_pending_request": false, "groups": [] }</pre>	
codigo_resultado	<p>Código de resultado que indica si se creó o no la cuenta de usuario.</p> <p>401 = Unauthorized</p> <pre>{   "message": "Las credenciales de autenticación no se proveyeron.",   "code": "401" }</pre>	json
codigo_resultado	<p>Código de resultado que indica si se creó o no la cuenta de usuario.</p> <p>401 = Unauthorized</p> <pre>{   "code": "401",   "message": "invalid_grant Invalid credentials given." }</pre>	json
codigo_resultado	<p>Código de resultado que indica si se creó o no la cuenta de usuario.</p> <p>401 = Unauthorized</p> <pre>{   "code": "401",   "message": "invalid_client " }</pre>	json
codigo_resultado	<p>Código de resultado que indica si se creó o no la cuenta de usuario.</p> <p>401 = Unauthorized</p> <pre>{   "code": "401",   "message": "unsupported_grant_type " }</pre>	json
mensaje	Mensaje asociado al código de respuesta	String





### Manejo de errores

En caso de un error inesperado, el servicio responderá con un HTTP 500.

### Datos del servicio:

URI:	{URL}/v1/oauth/token/
Método:	POST
Headers:	Content-Type: multipart/form-data
Paso de parámetros	Los parámetros se envían vía form-data
Body:	Los parámetros envían como FORM-DATA, esto imita a un formulario

### Parámetros de entrada:

Parámetro	Descripción	Tipo de dato	Obligatorio
grant_type	Se recibe el texto "refresh_token"	string	Sí
refresh_token	Token recibido con anterioridad del servicio anterior: ej. "refresh_token": "8M3RC6h0DdeZjcZjbVJGkqzV1zYTrO",	string	Sí
client_id	Dicho ID es generado mediante el sistema de encriptado de python utilizando la librería "oauthlib.common.py" de <b>40</b> caracteres de tipo UNICODE_ASCII_CHARACTER_SET = ('abcdefghijklmnopqrstuvwxyzABCDEFGH IJKLMNOPQRSTUVWXYZ0123456789');	string	Sí
client_secret	Dicho ID es generado mediante el sistema de encriptado de python utilizando la librería "oauthlib.common.py" de <b>128</b> caracteres de tipo UNICODE_ASCII_CHARACTER_SET = ('abcdefghijklmnopqrstuvwxyzABCDEFGH IJKLMNOPQRSTUVWXYZ0123456789');	string	Sí

### Parámetros de salida:

Parámetro	Descripción	Tipo de dato
codigo_resultado	Código de resultado que indica la respuesta del servicio para el acceso a la información.  200 = OK {  "access_token": "5uAuoUxdwf6qnkuuFX69Itr8pu7OGv",	json



	<pre>"expires_in": 3600, "token_type": "Bearer", "scope": "read", "refresh_token": "83cU3cdzV7u9eptHyLBQiBrHLW7BQg", "refresh_token_expires_in": 604800 }</pre>	
codigo_resultado	<p>Código de resultado que indica si se creó o no la cuenta de usuario.</p> <pre>401 = Unauthorized {   "code": "401",   "message": "invalid_grant Invalid credentials given." }</pre>	json
codigo_resultado	<p>Código de resultado que indica si se creó o no la cuenta de usuario.</p> <pre>401 = Unauthorized {   "code": "401",   "message": "invalid_client " }</pre>	json
codigo_resultado	<p>Código de resultado que indica si se creó o no la cuenta de usuario.</p> <pre>401 = Unauthorized {   "code": "401",   "message": "unsupported_grant_type " }</pre>	json
mensaje	Mensaje asociado al código de respuesta	String

### Manejo de errores

En caso de un error inesperado, el servicio responderá con un HTTP 500.