

API

Descripción

Una de las características importantes que notarás en el sistema es que todo proceso ofrece el concepto de apertura, esto significa que vas a poder actualizar la información del sistema a través de una interfaz estándar que hemos definido.

Esta interfaz, conocida comúnmente por sus siglas en inglés API (Application Programming Interface), ofrece un conjunto de funciones en las que ofrecemos funcionalidad de cada uno de los procesos del sistema.

Si leíste el párrafo anterior y no te entendés de qué estamos hablando, probablemente no tengas un perfil técnico, pero no te preocupes, en el siguiente párrafo te lo explicaremos de una forma más coloquial para que entiendas el uso que le podrías dar en tu empresa, comercio o estudio contable. Cada uno de los procesos de la nueva plataforma cuenta con una API. La API es una interface que permite que interactúen dos aplicaciones entre sí para lograr un intercambio de datos entre ellas. Si necesitás hacer un desarrollo externo a Tango informale a tu programador o asesor técnico que ahora contamos con APIs. No sólo será más fácil su trabajo, sino que a través de ellas podemos garantizar que se apliquen las mismas validaciones que utilizamos nosotros cuando vos trabajás con el sistema. De esta forma podemos asegurar que la información ingresada al sistema sea consistente. Todas las API de Tango están autodocumentadas y están detalladas en cada uno de los procesos. Para conocer más sobre API mirá este video.

[axoft_youtube url=»[### Características principales del API REST](https://youtu.be/8ICHmEmVEL4? class=»bordevideo»]</p></div><div data-bbox=)

Las operaciones que podés hacer son:

- **GET** para consultar y leer.
- **POST** para crear.
- **PUT** para editar.
- **DELETE** para eliminar.

Lo más importante a tener en cuenta al crear nuestro servicio o API REST no es el lenguaje en el que lo programes, sino que las respuestas a las peticiones se hagan en JSON, ya que es el lenguaje de intercambio de información más usado en la actualidad.

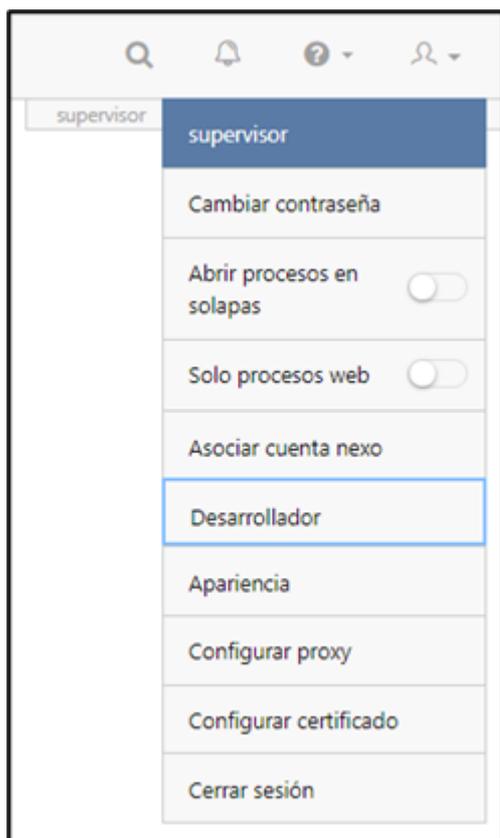
Si sos programador, podés interactuar con el API REST de Tango utilizando cualquier lenguaje de programación. Ejemplo: C#, JAVA, PHP, Swift, Kotlin, etc.

Obtener «Token de desarrollador»

Para utilizar el API del sistema, debes obtener primero una identificación de usuario, a la que llamamos TOKEN.

Ese TOKEN lo debés enviar en el encabezado (header) de cada transacción para garantizar la seguridad de la operación.

Para obtener el TOKEN, ingresá al Menú web del sistema > Menú de usuario y seleccioná la opción Desarrollador; luego de ingresar pulsá sobre el botón «Generar» y obtendrás el TOKEN que debé utilizar en el [header](#) para identificar tus transacciones.



Identificando la empresa

Tango es un sistema multiempresa, cuando utilizás el API REST, es necesario que identifiques la empresa sobre la que querés realizar la transacción. Esa identificación es un número que podés obtener en dos lugares:

- **Obervando la URL del sistema:** ingresá al menú del sistema y buscá en la dirección de internet de tu navegador el número que figura a continuación del texto «company/». Por ejemplo, en la url «<https://MiTango.axoft.com/company/117/menu/10023>» la identificación de tu empresa corresponde al número 117.
- **Consultando la base de datos «Diccionario»:** realizá una consulta a tabla «Empresa» y tomá

el valor del campo ID_Empresa para la empresa con la que querés operar.

Header

El «Header» es una propiedad del API REST y acompaña a cada método que expone el sistema como un conjunto de clave valor. En esta propiedad identificamos el tipo de transporte que es JSON, el [token de acceso](#) y la [identificación de empresa](#).

```
"Accept", "application/json";  
"ApiAuthorization", "8c45ec224ab849a3ac3519414d5f476f"  
"Company", 1
```

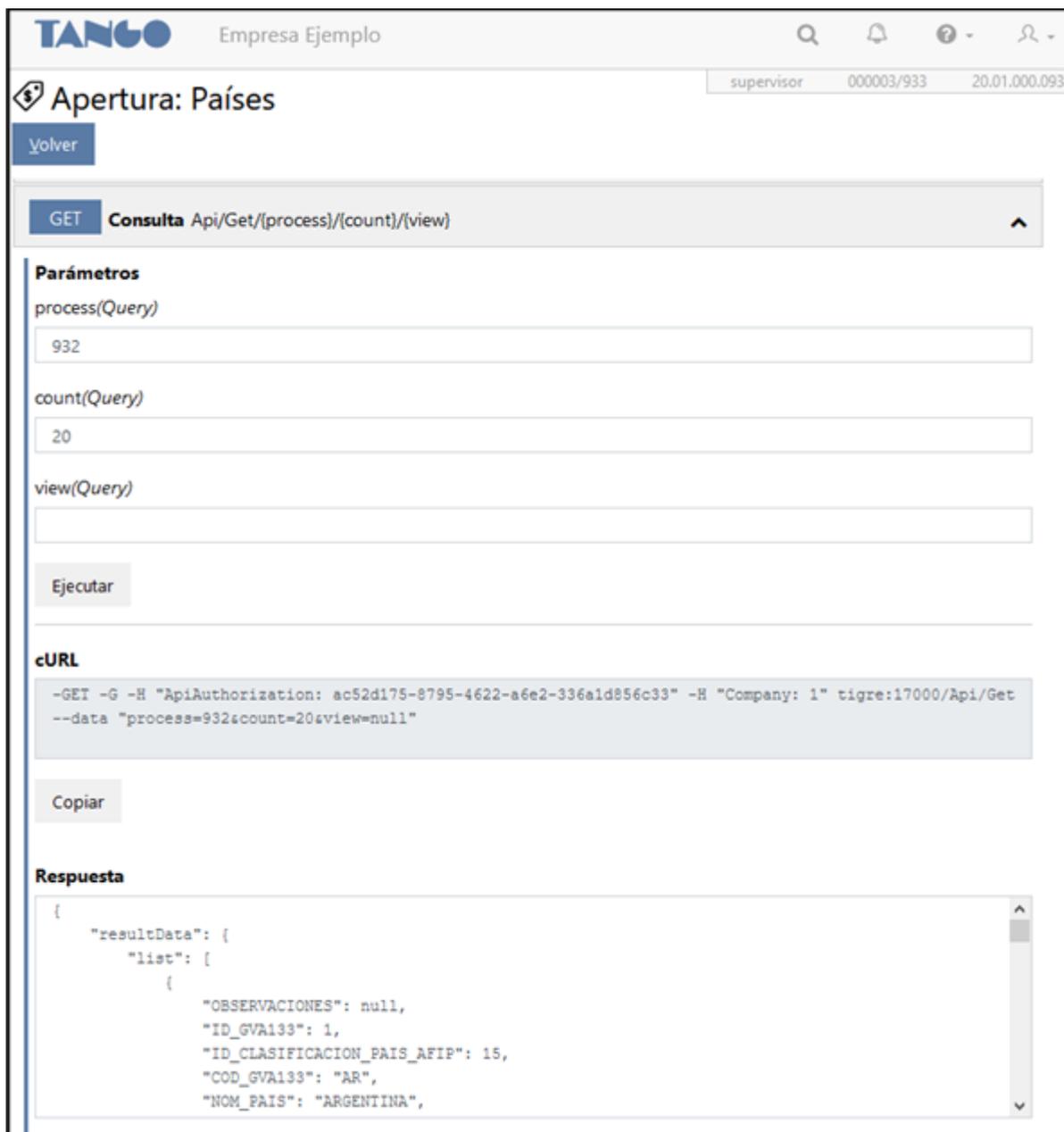
Métodos disponibles en API

Cada proceso cuenta con métodos para poder consumir del servicio API REST.

Estos métodos se encuentran autodocumentados y podés acceder a ellos ingresando a la opción Apertura > API dentro de la barra de herramientas de cada proceso.

Dentro de esa opción vas a encontrar información de cada método, información del modelo de transporte, un ejemplo en C# y otro en TypeScript a fin de que puedas copiar y pegar el código a tu programa.

También, podés probar el API desde esta misma sección; por ejemplo, si te posicionás en «Consulta», completás los parámetros y presionás el botón «Ejecutar» vas a obtener los registros correspondientes al proceso en esa empresa en formato JSON.



TANGO Empresa Ejemplo

Apertura: Países

Yolver

GET Consulta Api/Get/{process}/{count}/{view}

Parámetros

process(Query)
932

count(Query)
20

view(Query)

Ejecutar

cURL

```
-GET -G -H "ApiAuthorization: ac52d175-8795-4622-a6e2-336a1d856c33" -H "Company: 1" tigre:17000/Api/Get --data "process=932&count=20&view=null"
```

Copiar

Respuesta

```
{
  "resultData": {
    "list": [
      {
        "OBSERVACIONES": null,
        "ID_GVA133": 1,
        "ID_CLASIFICACION_PAIS_AFIP": 15,
        "COD_GVA133": "AR",
        "NOM_PAIS": "ARGENTINA",
      }
    ]
  }
}
```

Cuando ejecutes los métodos de ALTA, MODIFICACION o ELIMINACION es importante que siempre revises la respuesta que devuelve el proceso ya que puede ocurrir que la operación no se pueda realizar debido a alguna validación o falla en el armado del JSON.

[axoft_service title=»Nota» icon=»icon: info-circle» icon_color=»#6f6f6f» size=»18? class=»ax-nota ax-nota-inner»]

En el método 'Consulta', en un proceso de tipo ABM, obtenemos las columnas y filtros de la Vista por defecto con que fue desarrollado el proceso. Si queremos obtener otras columnas o aplicar un filtro nuevo, se debe crear una [Vista nueva](#) y el nombre de la vista se debe pasar en el parámetro VIEW.

[/axoft_service]

Método Consulta de procesos Live

En los procesos de tipo Consulta Live, contamos con apertura en cada uno de los procesos. El método 'Consulta' devuelve la consulta Live ejecutándose con los parámetros por defecto.

GET **Consulta** Api/GetApiLiveQueryData/{process}/{fromDate}/{toDate}/{pageSize}/{pageIndex}

Parámetros

process(*Query*)

fromDate(*Query*)

toDate(*Query*)

pageSize(*Query*)

pageIndex(*Query*)

Ejecutar

Se pueden modificar los filtros de fecha en los parámetros fromDate y toDate.

Toda la información de la consulta se devuelve paginada. El desarrollador puede modificar el tamaño de la página en el parámetro pageSize y obtener una página específica utilizando el parámetro pageIndex.

En el Json que retorna con los datos especificados, siempre se devuelve información con respecto a la cantidad de registros y cantidad de páginas para que el desarrollador lo tenga en cuenta en su lógica de consulta.

Respuesta

```
{,
  "pageIndex": 0,
  "pageSize": 10,
  "totalCount": 36967,
  "totalPages": 3697,
  "hasPreviousPage": false,
  "hasNextPage": true
},
"-----": "11"
```