Nombre técnico: **APP Synchro**

1. **Introducción**

Este módulo permite sincronizar datos de tablas desde un sistema externo con Odoo. Eso nos da la posibilidad de tener la información deseada replicada en dos sistemas distintos, alcanzando una alta integridad. Esta sincronización consiste en el envío o recepción de los registros que fueron creados, modificados o eliminados en uno de los dos sistemas, sea Odoo o el sistema externo, y necesita replicarse en el otro. Este proceso se puede realizar en dos modos, por **Acción Planificada (Via Cron)** o **En** **Tiempo Real**. El modo por **Via Cron** realiza la sincronización en una fecha y hora determinada, permitiéndonos establecer la frecuencia con la cual deseamos que se ejecute. Este modo tiene la ventaja que el usuario no interviene en ningún momento en el proceso, el cual se activa y ejecuta en automático. El modo **En Tiempo Real** se activa en el momento en que el usuario realiza una creación, modificación o eliminación de un registro cuya tabla está definida en la configuración que requiere sincronización. Tiene la ventaja que los dos sistemas están actualizados en todo momento.

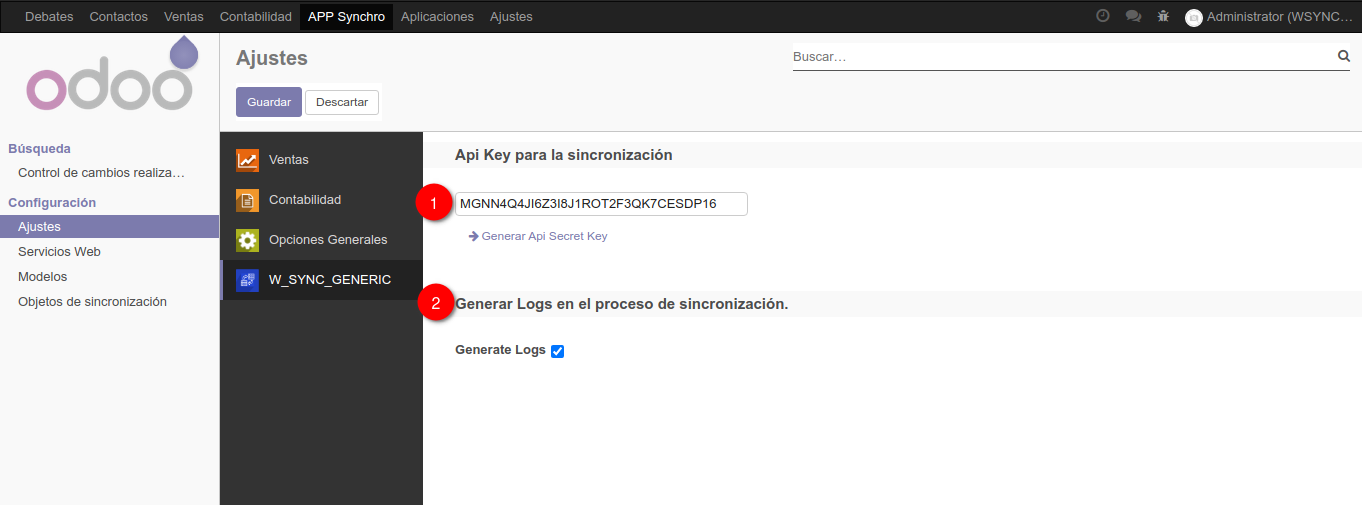
Además se cuenta con un **Control de cambios** que no es más que un log, donde podemos ver todos los detalles de los registros que se intentaron sincronizar, incluyendo hora de sincronización, datos que se están eviando o recibiendo, nombre y modelo del registro, si la sincronización fue exitosa o fallida; y más.

**Actualizaciones Parciales**

Cuando se está ejecutando una sincronización de registros por vía Cron, si la dirección configurada es Odoo-Sistema Externo o Bidireccional, cuando se va a enviar los registros desde Odoo hacia el sistema externo, si el registro nunca se ha sincronizado se envían todos los datos de los campos configurados en el objeto de sincronización. Si el registro ya fue previamente sincronizado y lo que se está haciendo es una actualización, entonces solo se enviarán los datos de los campos que fueron modificados. Estos son los llamadas actualizaciones parciales, porque solo se envía la información modificada.

1. **Configuración**
   1. Ajustes

Se establecen las configuraciones para generar el api key que caso de usarse en modo **En Tiempo Real** y si se desea llevar los Logs de cada sincronización.

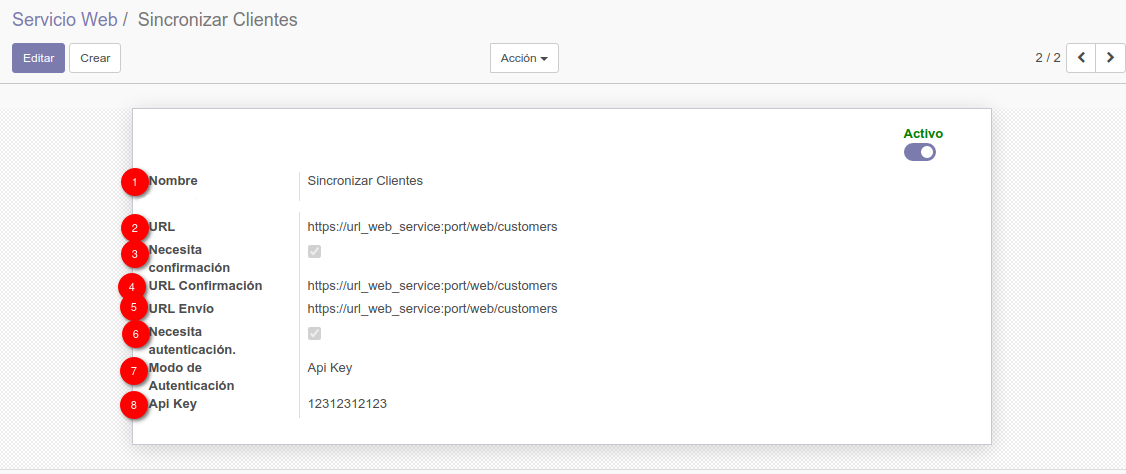


1. **Api Key para la sincronización**: Es la llave secreta que se utiliza para la sincronización en Tiempo Real desde el Sistema Externo hacia Odoo. Es necesario para consumir los servicios (se explica más adelante) enviar en los Headers esta api key que garantiza los permisos.
2. **Generar Logs**: Esta configuración es para si queremos que se cree un registro con los datos de cada sincronización y poder analizarlos.

* 1. Servicios Web

Se definen los Servicios Web a consumir y enviar la información.

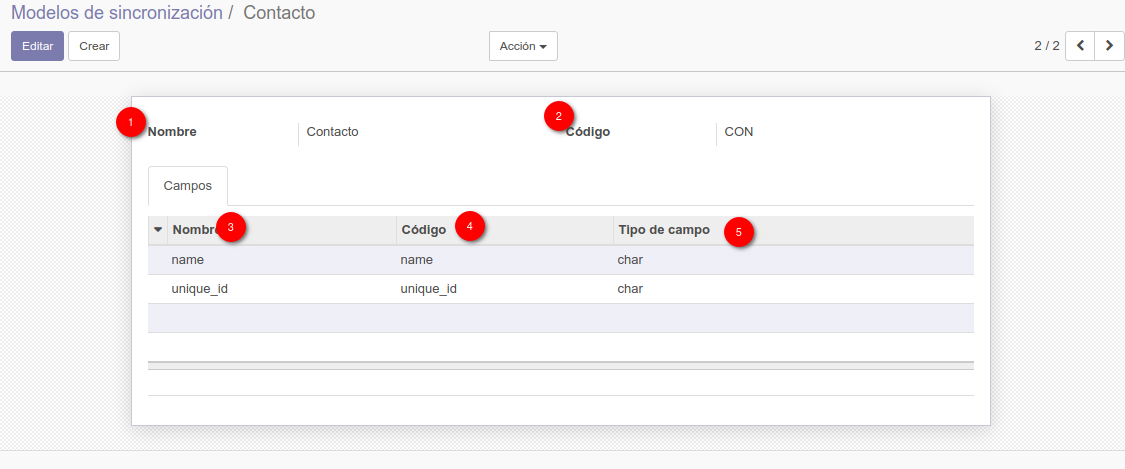
* + 1. **Nombre**: Nombre del Web Service.



* + 1. **URL**: Url donde obtener los registros para un tipo de tabla específico.
    2. **Necesita confirmación**: Se marca si hay que confirmarle al sistema externo que el registro fue insertado, modificado o eliminado correctamente.
    3. **URL Confirmación**: Url para enviar la confirmación. En la mayoría de los casos la url es la misma que la que se recibe el registro, pero puede darse el caso que se use otra para confirmar, por eso es este campo. Siempre hay poner la Url aunque sea la misma que para recibir.
    4. **URL Envío**: Es similar a lo anterior, en la mayoría de los casos es la misma que para obtener el registro, en caso que sea distinta se establece aquí. Si es la misma Url también hay que establecerla. El único caso que se deja en blanco es que la sincronización sea unidireccional desde el Sistema Externo a Odoo.
    5. **Necesita autenticación**: En caso que el consumo del Web Service requiera autenticación.
    6. **Modo de Autenticación**: Hay dos modos, Usuario y Contraseña; o por Api Key.
    7. Si el modo de autenticación es Api Key se habilita el campo Api Key para introducirla, en caso que sea Usuario y Contraseña se habilitan estos campos.
  1. Modelos de sincronización

Aquí se definen los modelos o tablas con los campos que se van a sincronizar.

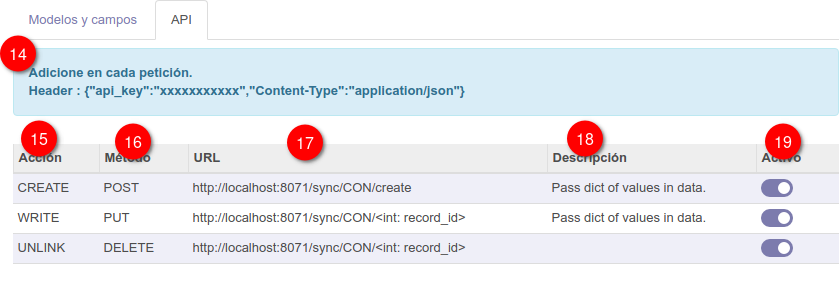
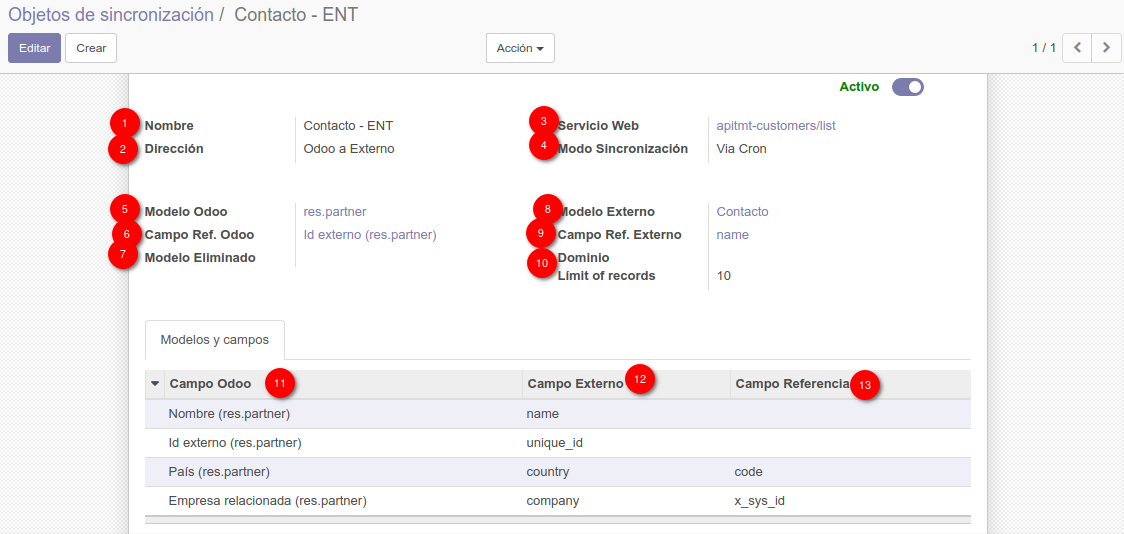
* + 1. **Nombre:** Nombre del modelo o tabla.



* + 1. **Código**: Código para para definirle, puede ser cualquiera
    2. **Campos Nombre**: Nombre del campo en el sistema externo. Nombre técnico en la tabla.
    3. **Campos Código**: Mismo que el Nombre.
    4. **Campos Tipo de campo**: Es el tipo de datos del campo.
  1. Objetos de Sincronización

Aquí es donde se establece toda la configuración para sincronizar. Se define una por cada modelo o tabla. Se establece el WS relacionado, que ya se configuró anteriormento. También se establece el modo de sincronización, así como la dirección y prioridad. Además se emparejan los campos del Sistema Externo con su análogo en Odoo.

* + 1. **Nombre**: Nombre del Objeto, puede ser cualquiera, siempre haciendo referencia el modelo de Odoo con el externo para saber identidicarlo.



* + 1. **Dirección**: Existen tres opciones:

**Odoo a Externo**: Para enviar registros de este modelo desde Odoo hacia al Sistema Externo.

**Externo a Odoo**: Para recibir registros desde el Sistema Externo hacia Odoo.

**Bidireccional:** Se envía y se reciben desde ambos lados. Para este caso ver el campo Prioridad de sincronización el cual es solo visible cuando la dirección es bidireccional.

* + 1. **Servicio Web**: Se selecciona el Servicio Web que tenemos definido para este modelo.
    2. **Modo Sincronización**: Definimos si este modelo se sincronizará mediante un Via Cron o En tiempo Real. Es importante aclarar que cuando se selecciona En Tiempo Real se habilita una pestaña APIs con los Servicios Web expuestos en Odoo para consumir desde el Sistema Externo.

**Prioridad de Sincronización**: Este campo es solo visible cuando el modo de sincronización es Via Cron. Se define para establecer la prioridad en caso que haya un registro que ha sido modificado tanto en Odoo como en el Sistema Externo, para definir cual modificación prevalece.

* + 1. **Modelo Odoo**: Se establece el modelo a sincronizar en Odoo.
    2. **Campo Ref. Odoo**: Es el campo referencia en Odoo, siempre será el campo Id externo.
    3. **Modelo eliminado**: Este campo solo es visible cuando el modo de sincronización es Via Cron. Se debe establecer si queremos que los regitros eliminados en Odoo también se eliminen en el Sistema Externo. Al establecerlo se lista una única opción, es esa la que debemos seleccionar.
    4. **Modelo Externo**: Seleccionamos el Modelo Externo homólogo a contactos. Este ya lo configuramos en la sección de **Modelos de sincronización**.
    5. **Campo Ref. Externo**: Es el campo referencia en el Sistema Externo, está entre los campos que definimos en el Modelo de Sincronización para este modelo. Debe ser el Id del registro en el Sistema Externo.
    6. **Dominio y Límite**: Esto se usa si definir un dominio de los registro en Odoo que queremos sincronizar. Por ejemplo. En Odoo en el modelo res.partner están todos los contactos, clientes, empresas etc. Si solo quisieramos sincronizar los clientes sería [(‘customer’, ‘=’, True)]. El límite se usa para definir la cantidad de registros en cada sincronización si es modo Vía Cron.
    7. **Campo Odoo**: Corresponde a los campos en Odoo que queremos sincronizar y emparejarlo con su homólogo. Nunca debe faltar el campo Id. externo.
    8. **Campo Externo**: Corresponde a los campos en el Sistema Externo. Ya están definidos en el Modelo de Sincronización, solo falta emparejarlos.
    9. **Campo Referencia**: Esto es muy importante, se debe establecer el campo referencia por el cual encontrará el registro en Odoo. Solo se utilizará si el campo es de tipo One2many, Many2many o Many2one. Es el campo por el cual los registros están relacionados. En la imagen hay dos ejemplos de muestra que se explican a continuación.

**País**: En casi todos los sistemas que tengan el campo país, en esa tabla hay un campo que tiene el mismo valor para todos los sistemas. Ese campo contiene el código de País, por ejemplo para España es ES, México es MX etc. Bueno el campos referencia que se usó en este caso es code, porque en Odoo la tabla de país es en el campo code donde está el código único de cada país, y es el mismo en el sistema externo. Los registros de País están unidos con el Sistema Externo por el campo code en Odoo.

**Empresa relacionada**: En este caso el campo referencia es x\_sys\_id porque es el campo que guarda el Id. del sistema externo.

**Mejora realizada: Si no se define nada en este campo, el sistema asume que el valor que se está enviando es el Id en Odoo de ese campo relacional. Esta mejora solo aplica para los campos Many2one.**

* + 1. Es una nota donde se explica lo que se debe enviar en el Header de cada petición. El api key previamente generado y el Content Type debe ser application/json. La variable de la api\_key fue modificada por cuestiones técnicas, ahora es **Apikey**.
    2. Listado de Web Services construidos especificamente para este modelo. **Acción** : Es la acción que se desea realizar en Odoo, ‘Crear’, ‘Modificar’ o ‘Eliminar’ un registro.
    3. **Método**: Es el método a usar en la petición, Ejemplo:

POST: requests.request("POST", url, headers=headers, data=payload)

PUT: requests.request("PUT", url, headers=headers, data=payload)

DELETE: requests.request("DELETE", url, headers=headers, data=payload)

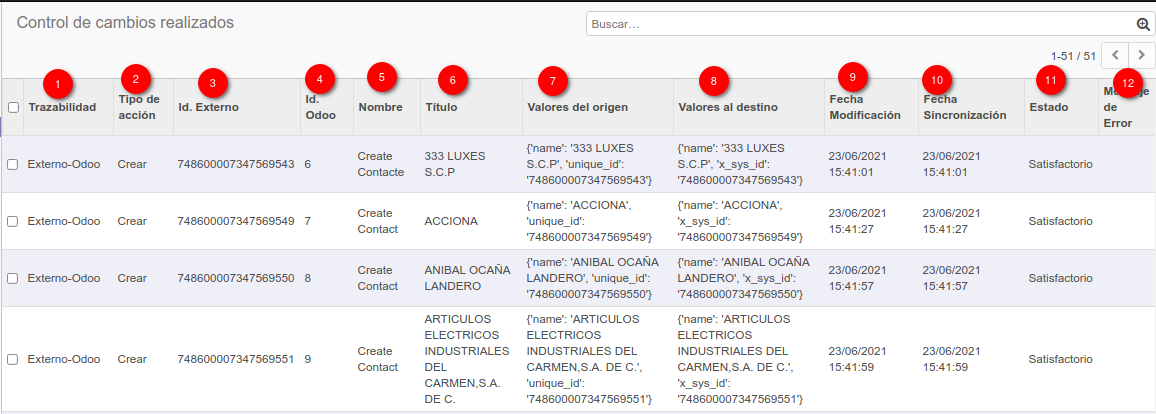
**Nota aclaratorio: Para el DELETE no se debe poner Content type en el header.**.

* + 1. **URL**: Dirección a realizar la petición. En la versión actual se modificó para recibir string en vez de enteros. Por tanto la url termina ahora en **<string: record\_id>**.
    2. **Descripción**: Descripción de si hay que pasar pasar parámetros o no.
    3. **Activo**: Activar o desactivar el Web Service.

1. **Control de Cambios**

Aquí se muestra un histórico de todas las sincronizaciones que se ha realizado, mostrando la trazabilidad de la misma.

* 1. **Trazabilidad**: Muestra la dirección de la sincronización.



* 1. **Tipo de acción**: Muestra el tipo de acción, Crear, Modificar o Eliminar.
  2. **Id. Externo**: Es el Id. Externo en el registro en cuestión.
  3. **Id. Odoo**: Id. Odoo del registro en cuestión.
  4. **Nombre**: Nombre de la acción.
  5. **Título**: Nombre del registro en cuestión.
  6. **Valores del origen**: Es un diccionario con los valores que recibe desde el origen de esa sincronización.
  7. **Valores al destino**: Es un diccionario con los valores que se envían al destino de esa sincronización.
  8. **Fecha Modificación**: Fecha en que se modificó el registro.
  9. **Fecha Sincronización**: Fecha en que se realizó la sincronización.
  10. **Estado**: Estado de la sincronización, Satisfactorio o Fallido.
  11. **Mensaje de Error**: Es el mensaje que se arroja si el estado fue Fallido.

1. **Estructura**

A continuación se define la estructura de las llamadas que deben cumplir los Servicios Web para recibir, confirmar y enviar los datos y se realice una correcta sincronización.

* **Estructura en peticiones en modo Via Cron.**

Todas las peticiones se realizan por POST.

* 1. Petición para recibir:
* Se envía

*{*  
'limit': 15  
*}*

* Se espera recibir

*{*

# this is the status of response

'status': "success, error or failed",  
 'error\_message': "Description of error if status is error",

# content is a list of the values for each record  
 'content': *[*  
 *{*  
'action': "insert, update or delete",

# values not necesary if action is delete  
 'values': *{*  
'external\_id': 1,  
 'name': "Contact 1"  
 *}*   
 *}*,  
 *{*  
'action': "insert, update or delete",   
 'values': *{*  
'external\_id': 2,  
 'name': "Contact 2"  
 *}*   
 *}*  
 *]*  
*}*

* 1. Petición de confirmación:
* Se envía:

*{*  
# this is the status of response  
 'status': "success, error or failed",  
 'error\_message': "Description of error if status is error",  
 'content': *[*  
 *{*  
# this is the status of record  
 'status': 'failed',  
 'error\_message': 'Failed reason',  
 'external\_id': 1  
 *}*,  
 *{*  
'status': 'success',  
 'error\_message': False,  
 'external\_id': 2  
 *}*  
 *]*  
*}*

* Se espera:

*{*  
'status': 'success, failed',  
 'error\_message': "Description of error if status is error else False"  
*}*

* 1. Petición para enviar:
     + - * Se envía:

*{*  
'action': 'create, update, delete',  
 # values not necesary if action is delete  
 'values': *{*  
'name': 'Contact 3'  
 *}*,  
 'external\_id': 2  
*}*

* + - * + Se espera:

*{*  
'status': 'success, failed',  
 'error\_message': "Description of error if status is error else False"  
*}*

* **Estructura en peticiones en modo Tiempo Real.**
  1. Petición para crear desde Externo a Odoo:
* Se envía desde Externo a Odoo:



import requests  
url = "http://localhost:8071/sync/CON/create"  
payload={ "name": "Name", "unique\_id": 1231}  
headers = *{*  
'api\_key': 'MGNN4Q4JI6Z3I8J1ROT2F3QK7CESDP16', # api key generado previamente  
 'Content-Type': 'application/json'  
*}*  
response = requests.request*(*"POST", url, headers=headers, data=payload*)*

* Se retorna:

*{*  
"jsonrpc": "2.0",  
 "id": null,  
 "result": *{*  
"status": "success", # may be failed  
 "error\_message": "" # description of error if status is failed  
 }  
*}*

* 1. Petición para modificar desde Externo a Odoo:



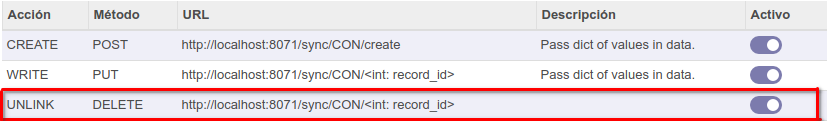
* + - * Se envía desde Externo a Odoo:

import requests  
url = "http://localhost:8071/sync/CON/1231"  
payload=*{*"name": "Name of record modified"*}*  
headers = *{*  
'api\_key': 'MGNN4Q4JI6Z3I8J1ROT2F3QK7CESDP16',  
 'Content-Type': 'application/json'  
*}*  
response = requests.request*(*"PUT", url, headers=headers, data=payload*)*

* Se retorna:

*{*  
"jsonrpc": "2.0",  
 "id": null,  
 "result": *{*  
"status": "success", # may be failed  
 "error\_message": "" # description of error if status is failed  
 }  
*}*

* 1. Petición para eliminar desde Externo a Odoo:
     + - Se envía desde Externo:



import requests  
url = "http://localhost:8071/sync/CON/1354"  
payload=*{}*  
headers = *{*  
'api\_key': 'MGNN4Q4JI6Z3I8J1ROT2F3QK7CESDP16'  
*}*  
  
response = requests.request*(*"DELETE", url, headers=headers, data=payload*)*

* + - * Se retorna:

*{*  
"jsonrpc": "2.0",  
 "id": null,  
 "result": *{*  
"status": "success", # may be failed  
 "error\_message": "" # description of error if status is failed  
 }  
*}*