

4D Mobile

Wakanda, publicado por 4D SAS, es una plataforma de desarrollo y de publicación de aplicaciones web totalmente basadas en tecnologías estándar tales como JavaScript y HTML5.

Puede utilizar la arquitectura "4D Mobile" para establecer un vínculo directo entre 4D y Wakanda. Con esta configuración, se combina la riqueza gráfica y funcional de las interfaces web de última generación de Wakanda con el poder de sus bases de datos 4D.

Si desea crear de inmediato su primer vínculo entre 4D y Wakanda, marque en el párrafo **Configuración** que tiene una configuración apropiada y a continuación, consulte la sección **Ejemplo paso a paso**.

-  [Arquitectura 4D Mobile](#)
-  [Ejemplo paso a paso](#)
-  [Configuración de la base de datos 4D](#)
-  [Configuración de la aplicación Wakanda](#)
-  [Llamar tablas y métodos 4D](#)
-  [Uso de relaciones](#)
-  [Gestión de sesiones 4D Mobile](#)
-  [Acerca de la seguridad de la aplicación 4D Mobile](#)

Arquitectura 4D Mobile

Configuración

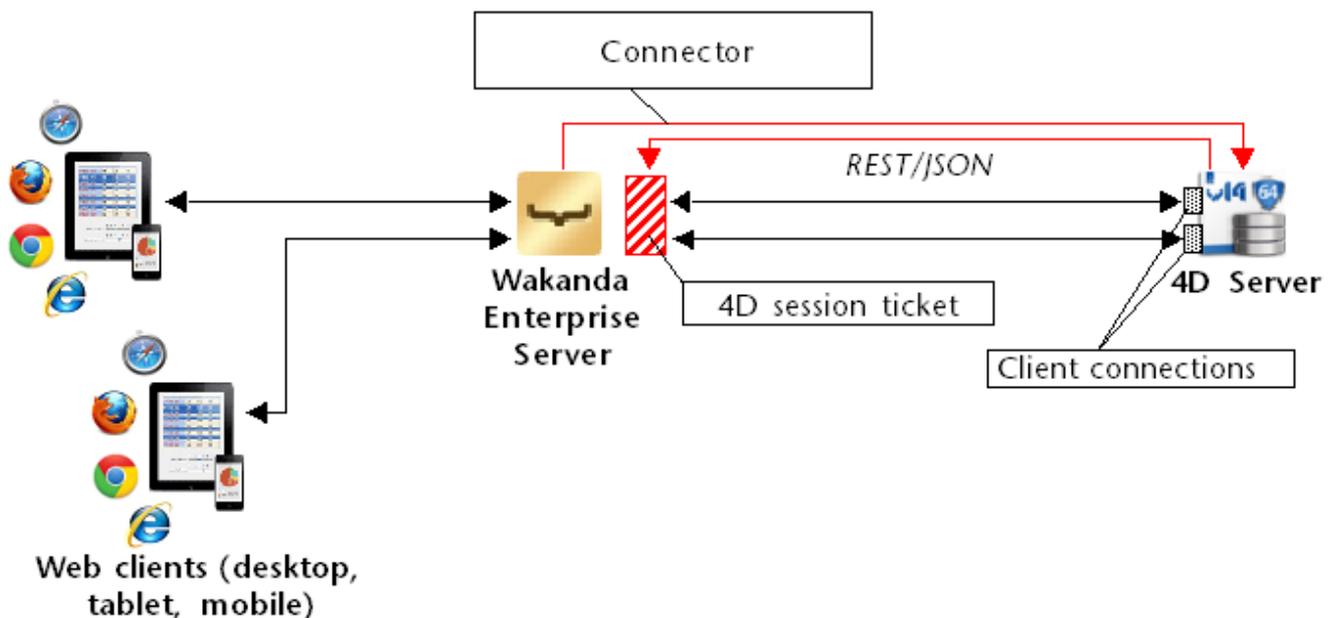
Para configurar una arquitectura utilizando el conector "Pro" 4D/Wakanda, necesita como mínimo:

- **4D** monopuesto (*versión Professional*) para desarrollar y probar su solución mediante el conector 4D mobile (tres conexiones 4D Mobile simultáneas se permiten en este contexto) o **4D Server** con un Expansion pack 4D Mobile (dos conexiones 4D Mobile autorizadas).
- **Wakanda Enterprise Server** así como también **Wakanda Enterprise Studio** para el desarrollo. Puede descargar ambas aplicaciones desde la [página de descarga Wakanda](#) (pestaña *Enterprise*).
- una base 4D y una aplicación Wakanda que deban comunicarse entre sí

Por el lado de 4D, debe configurar cada tabla, atributo y método a los que accederán las aplicaciones Wakanda (ver la sección **Configuración de la base de datos 4D**).

Descripción

La arquitectura de 4D Mobile se puede representar de la siguiente manera:



Cuando se inicia la solución Wakanda, Wakanda Enterprise server establece un enlace con 4D Server en función de los parámetros definidos en la caja de diálogo "Connect to Remote Datastore" o de los métodos JavaScript de conexión. Una vez la conexión es aceptada por 4D Server (ver el párrafo **Control de acceso REST**), un "tiquete" de sesión cliente 4D Mobile se entrega al servidor Wakanda. Este tiquete será utilizado por Wakanda para todas las peticiones cliente REST posteriores.

A través de esta conexión, el servidor Wakanda potencialmente puede acceder a dos tipos de recursos de la base 4D:

- las tablas y sus atributos (incluyendo sus datos)
- los métodos proyecto

Cuando estos recursos son autorizados son utilizables directamente del lado Wakanda, como si pertenecieran al catálogo local de la aplicación Wakanda (su acceso es transparente desde la aplicación Wakanda).

Cuando un cliente web envía al servidor Wakanda una solicitud requiere acceso a la base 4D, esta solicitud se envía al servidor 4D utilizando el tiquete actual y una conexión 4D Mobile se abre en el equipo 4D Server. Esta conexión permanece abierta mientras el usuario realiza las solicitudes y se cierra por defecto después de un tiempo de espera de 60 minutos de inactividad. Este tiempo de espera por defecto se puede cambiar durante la ejecución del método **mergeOutsideCatalog()** a través del parámetro *timeout*.

Si durante la sesión, se alcanza el número de licencias correspondientes al número de conexiones 4D Mobile autorizadas en 4D Server, se devuelve un mensaje de error al servidor Wakanda,

Nota: También pueden establecer vínculos temporales dinámicos entre Wakanda y las aplicaciones 4D mientras la aplicación Wakanda se está ejecutando utilizando los métodos **addRemoteStore()** y **openRemoteStore()**. Estos métodos se describen en **openRemoteStore()** y **RemoteStore()**.

Ejemplo paso a paso

Esta sección está pensada para ayudarle a familiarizarse con la funcionalidad conector Wakanda/4D mediante un ejemplo paso a paso. Vamos a:

- crear y configurar una base de datos 4D
- crear una aplicación Wakanda de una sola página
- mostrar datos de la base de datos 4D en la página Wakanda.

Para mantener el ejemplo sencillo, vamos a utilizar una aplicación 4D y una aplicación Wakanda que se ejecutan en la misma máquina. Por supuesto, también puede utilizar una arquitectura remota.

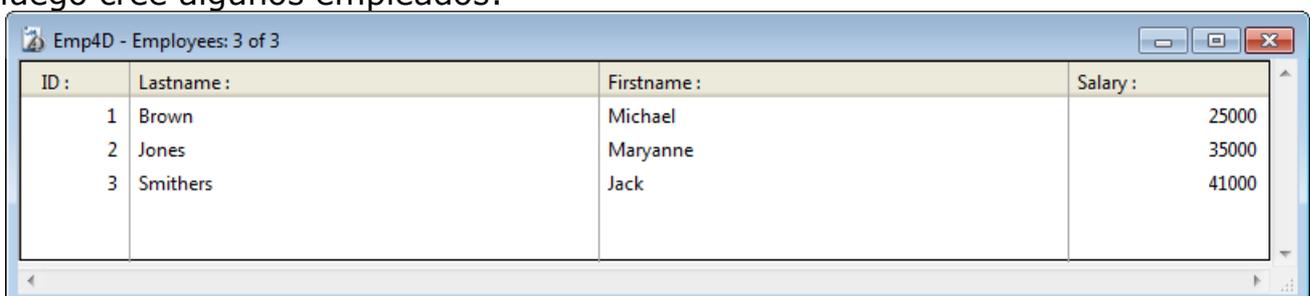
1- Creación y configuración de la base 4D

1. Inicie su aplicación 4D o 4D Server y cree una nueva base de datos. Puede llamarla "Emp4D", por ejemplo.
2. En el editor de estructura, cree una tabla [Empleados] y agregue los siguientes campos:
 - Apellido (Texto)
 - Nombre (Texto)
 - Salario (Entero largo)



El atributo "Exponer con el servidor 4D Mobile" está seleccionado por defecto para la tabla y para todos los campos; no cambie esta configuración.

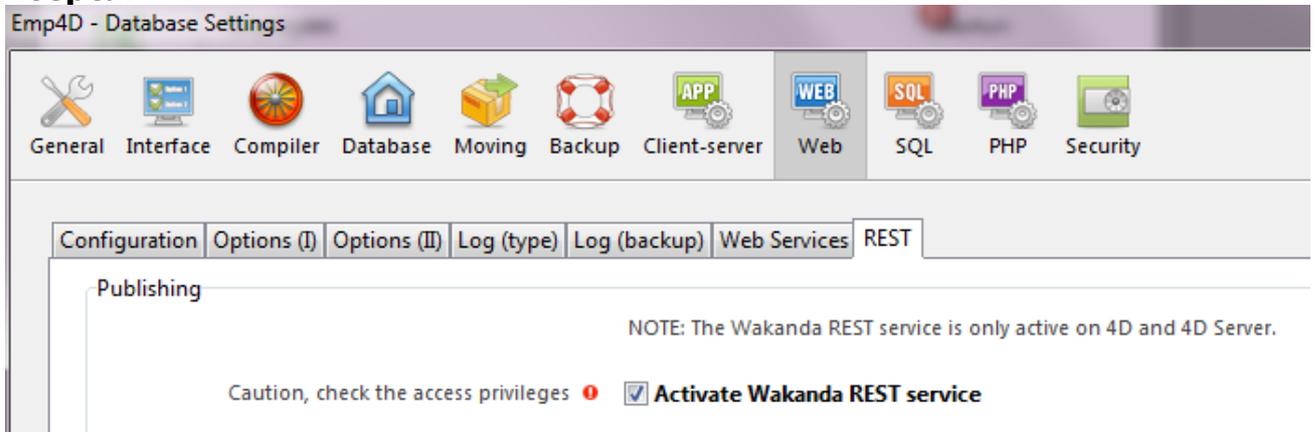
3. Haga clic en el botón **Tablas** y deje que 4D cree los formularios por defecto, luego cree algunos empleados:



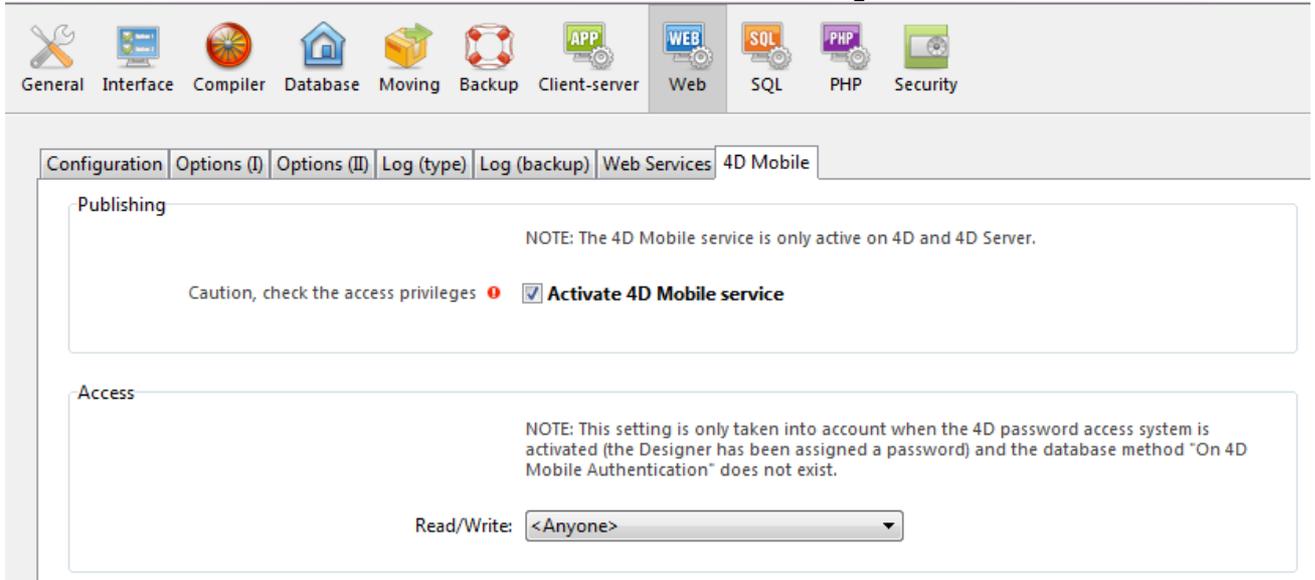
ID :	Lastname :	Firstname :	Salary :
1	Brown	Michael	25000
2	Jones	Maryanne	35000
3	Smithers	Jack	41000

4. Muestre la caja de diálogo de Propiedades de la base, página **Web** y haga clic en la pestaña **4D Mobile**.

5. Marque la opción "Activar los servicios 4D Mobile" y luego haga clic en **Aceptar**:



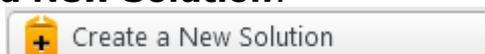
6. Seleccione **Arrancar el servidor web** en el menú **Ejecutar**:



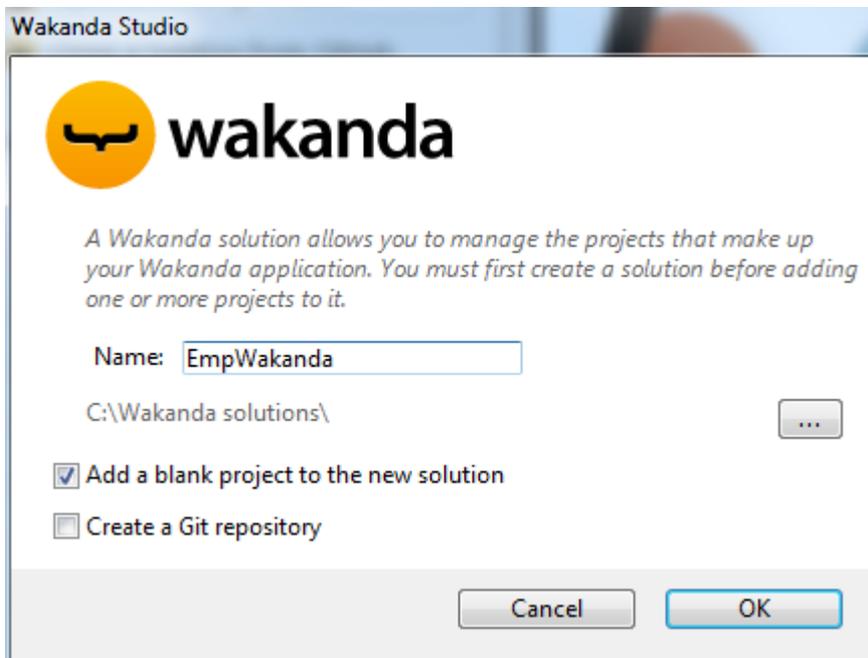
La base de datos 4D ya está lista para responder a las solicitudes 4D Mobile de Wakanda. Tenga en cuenta que para simplificar este ejemplo, no vamos a controlar los accesos 4D Mobile. Sin embargo, en un contexto de producción o de arquitectura abierta, es indispensable para garantizar los accesos 4D Mobile vía REST (ver la sección **Acerca de la seguridad de la aplicación 4D Mobile**).

2 - Crear la aplicación Wakanda

1. Inicie la aplicación "Wakanda Enterprise Studio" y haga clic en el botón **Create a New Solution**:

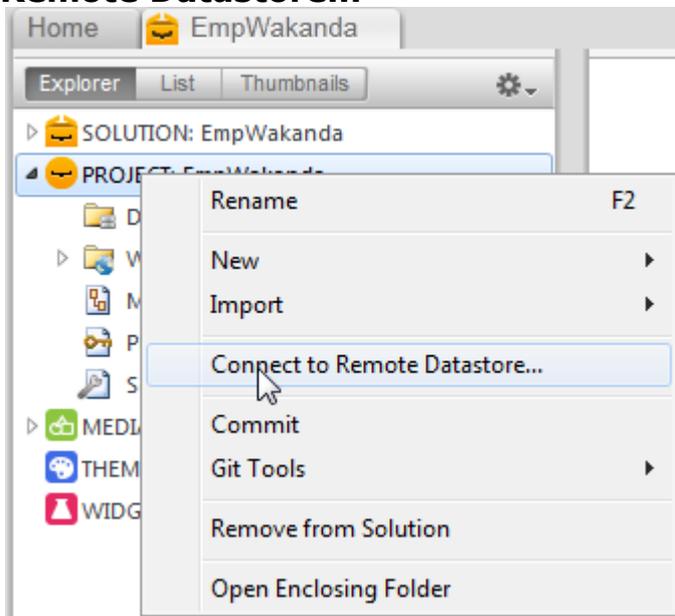


2. En la caja de diálogo de creación, escriba, por ejemplo, "EmpWakanda" y haga clic en **OK**:

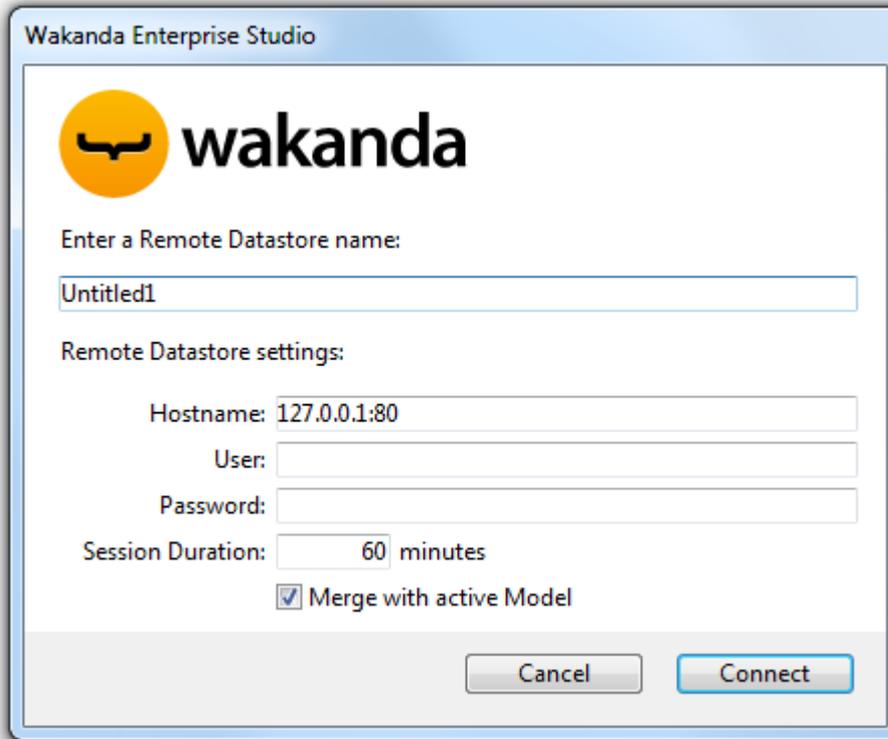


El proyecto de aplicación se crea y sus elementos por defecto se muestran en el Wakanda Studio Explorer, al lado izquierdo de la ventana.

3. Haga clic derecho en la línea PROJECT y seleccione el comando **Connect to Remote Datastore...**



Aparece la caja de diálogo de conexión:



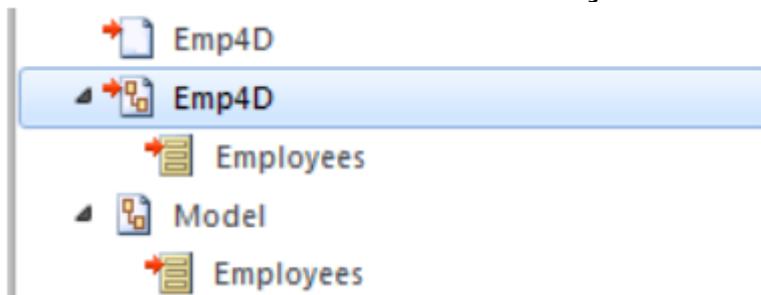
4. Introduzca un nombre para el enlace, por ejemplo "Emp4D":

Enter a Remote Datastore name:

Emp4D

Este es el nombre local del enlace, tal y como aparecerá en Wakanda Enterprise Studio. Puede introducir cualquier nombre, pero mantener las cosas simples usamos el nombre de la base 4D.

5. (opcional) Si su 4D Server está ubicado en una máquina diferente que la de Wakanda Enterprise Studio, introduzca su hostname o su dirección IP como parámetro Hostname. De lo contrario, puede conservar la dirección local "127.0.0.1:80" (o "localhost").
6. Conserve los otros parámetros por defecto y haga clic en el botón **Connect**. Después de algunos momentos, puede ver que el modelo externo "Emp4D" listado en los archivos de la aplicación Wakanda y la tabla [Empleados] de la aplicación 4D se lista en las clases de datos del modelo local. Los elementos externos se indican con una flecha roja:



Nota: el primer archivo Emp4D contiene los parámetros de conexión.

En caso de problemas...

Si la tabla no se muestra en la lista en este momento, compruebe que:

- ningún servicio o software de terceros (mensajería instantánea, por ejemplo) esté en conflicto con el puerto de publicación (80 por defecto) del servidor HTTP de 4D,

- del lado 4D, el Servidor web 4D se inició, los servicios 4D Mobile se activan y la tabla ha sido expuesta,
- la dirección pasada en el parámetro "Hostname" es válida.

Para comprobar si el servidor 4D en realidad responde a las peticiones REST, pueden probar las siguientes URLs en su navegador:

```
<address>/rest/$catalog/$all
```

(devuelve las tablas expuestas con el servicio 4D Mobile)

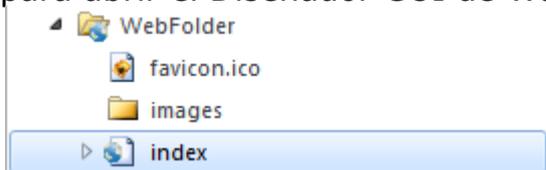
```
<address>/rest/my_table/my_method
```

(devuelve el resultado del método - si devuelve un resultado)

3 - Visualización de datos 4D utilizando un widget Wakanda

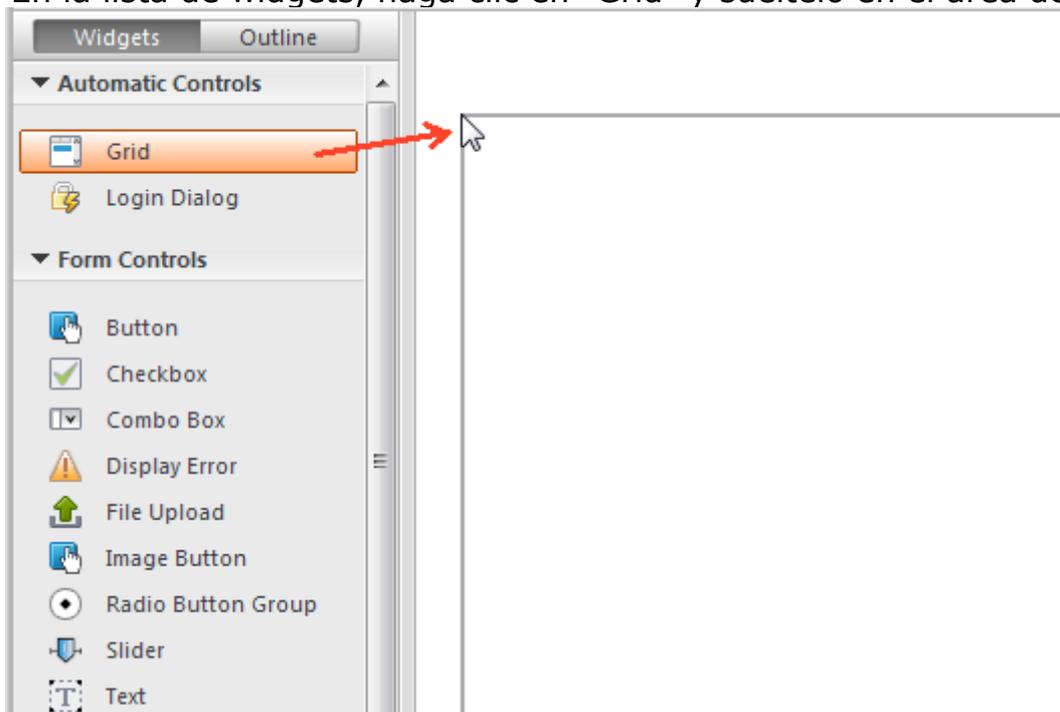
Ahora vamos a asociar una tabla 4D con un widget Wakanda por simple arrastrar y soltar, a continuación, luego lance el Wakanda Enterprise Server y visualice los datos.

1. Abra la carpeta "WebFolder" en el Explorer y haga doble clic en la página Index para abrir el Diseñador GUI de Wakanda.

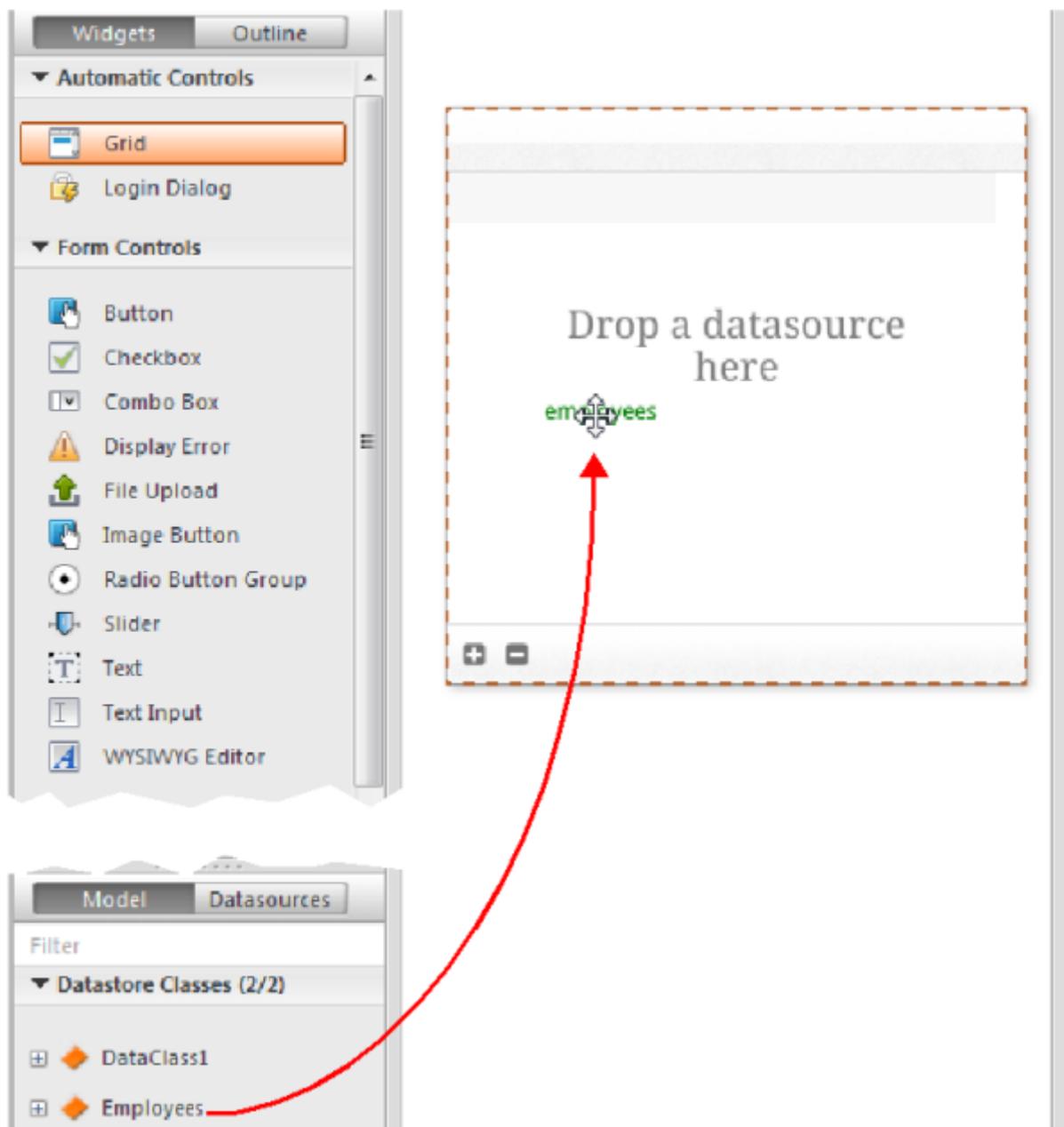


Nota: "WebFolder" contiene los elementos destinados para la publicación web de su proyecto. "Index" es la página por defecto del proyecto.

2. En la lista de widgets, haga clic en "Grid" y suéltelo en el área de trabajo:



3. En la lista de clases de datos para el modelo, haga clic en "Empleados" y suéltelo en la rejilla que acaba de crear:



En este punto, el editor crea automáticamente una fuente de datos para usted basado en la clase "Empleados", que gestionará el contenido del widget. Esta datasource es un objeto JavaScript gestionado por Wakanda, llamada "empleados" por defecto, es decir, el nombre de la clase con su primera letra en minúscula. El widget muestra una vista previa de su contenido. Puede ampliarlo para visualizar todos los campos de la fuente de datos:

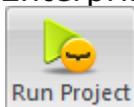
ID	Lastname	Firstname	Salary
Text	Text	Text	Text

La asociación entre el *datasource* y los widgets ahora está establecida.

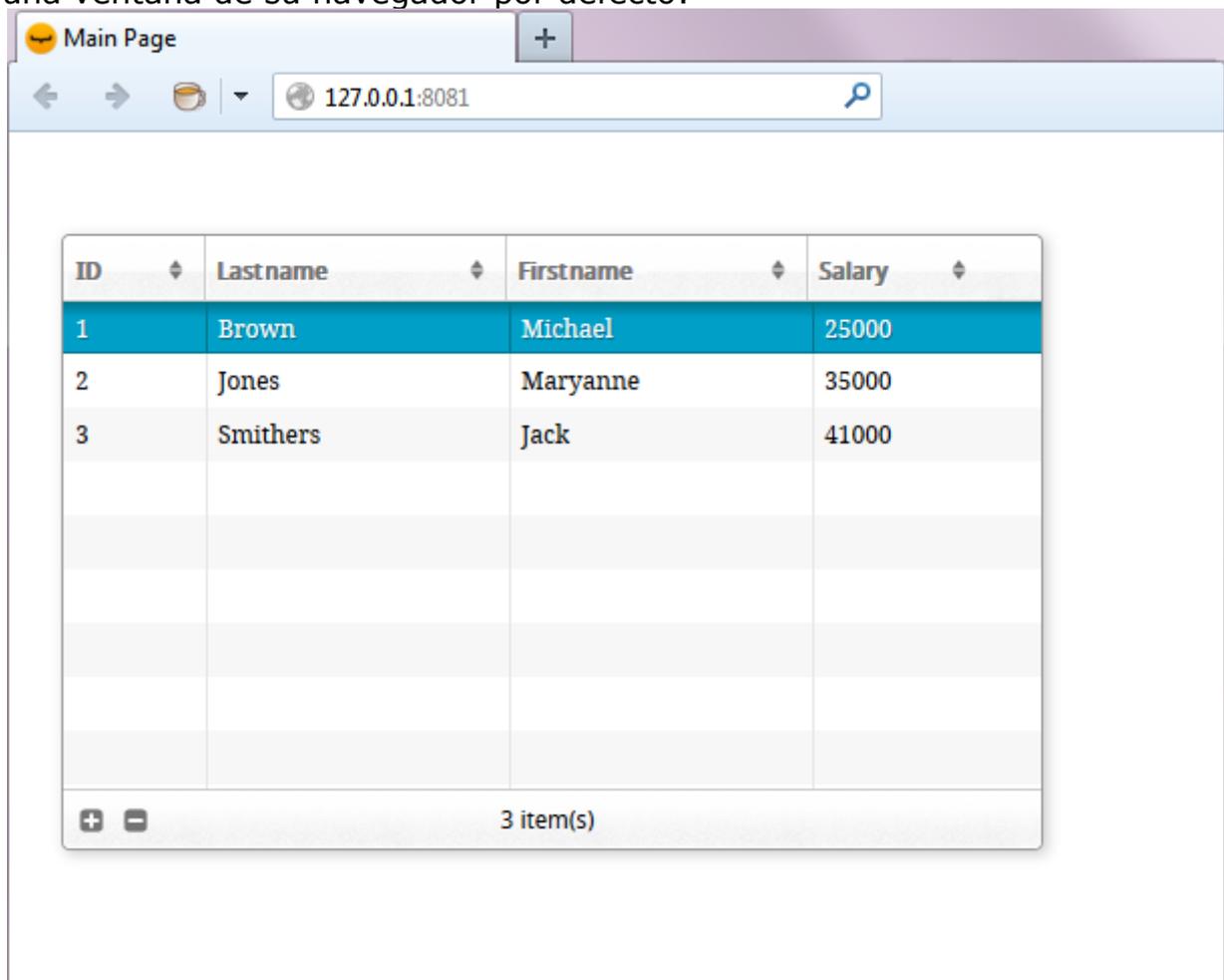
- Haga clic en el botón **Guardar**  en la barra de herramientas del editor.

Ahora vamos a mostrar los datos utilizando un navegador.

- Haga clic en el botón **Run project** en la barra de herramientas de Wakanda Enterprise Studio:



Esto inicia el Wakanda Enterprise Server y publica la aplicación "EmpWakanda". Gracias al enlace que hemos creado, los datos de la base 4D se muestran en una ventana de su navegador por defecto:



The screenshot shows a web browser window with the address bar set to 127.0.0.1:8081. The page displays a table with the following data:

ID	Lastname	Firstname	Salary
1	Brown	Michael	25000
2	Jones	Maryanne	35000
3	Smithers	Jack	41000

At the bottom of the table, it indicates "3 item(s)".

Puede probar las propiedades dinámicas del enlace mediante el cambio de

datos del lado Web. Por ejemplo, aquí cambiamos el apellido de Maryanne Jones por "Jackson" y este cambio se refleja inmediatamente en 4D:

ID	Lastname	First name	Salary
1	Brown	Michael	25000
2	Jackson	Maryanne	35000
3	Smithers	Jack	41000

ID :	Lastname :	Firstname :
1	Brown	Michael
2	Jones	Maryanne
3	Smithers	Jack

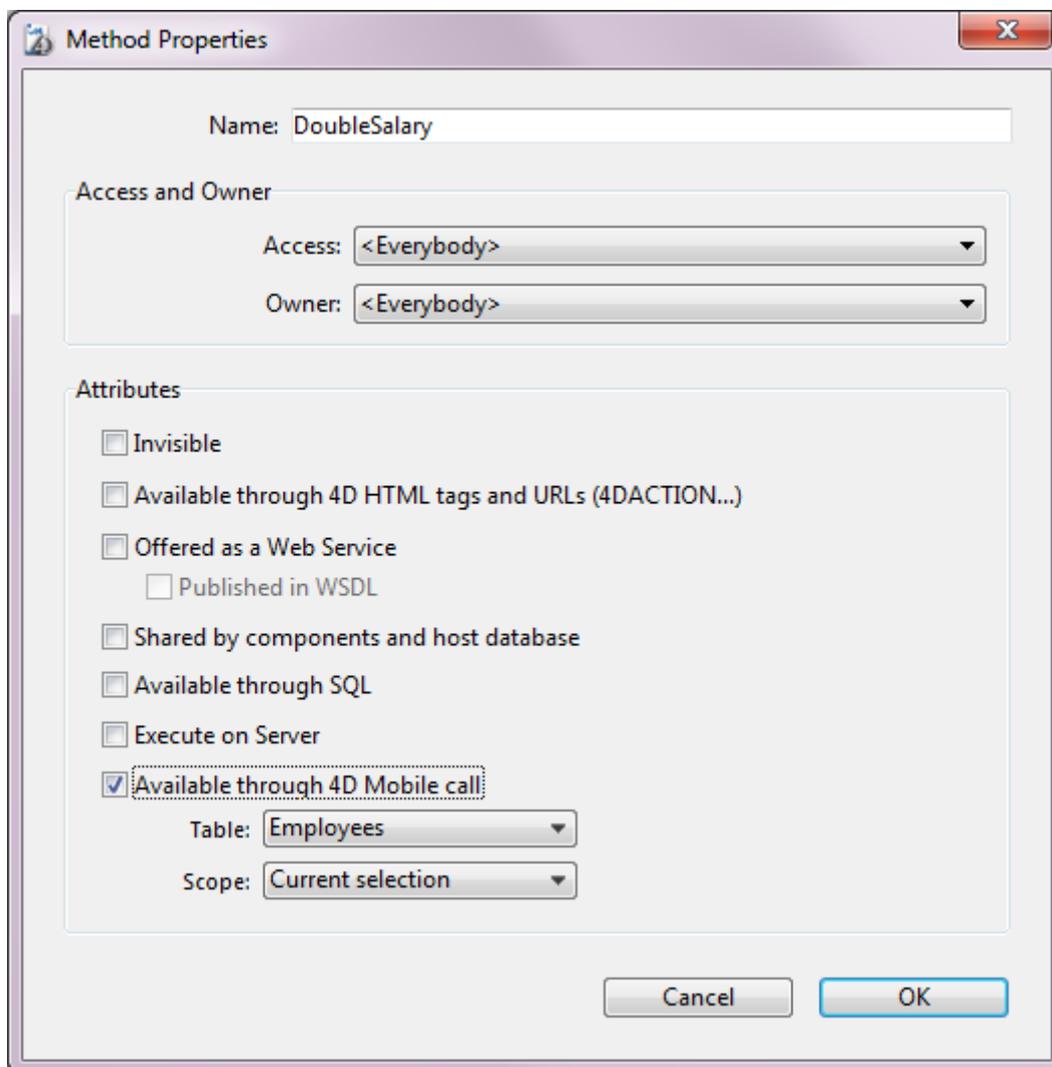
4 - Creación y llamada a un método 4D

Ahora vamos a crear un método de proyecto muy simple del lado 4D y a ejecutarlo desde nuestra página web. Este método doblará todos los salarios.

1. Del lado 4D, cree un método proyecto llamado **DoubleSalary** e introduzca el siguiente código:

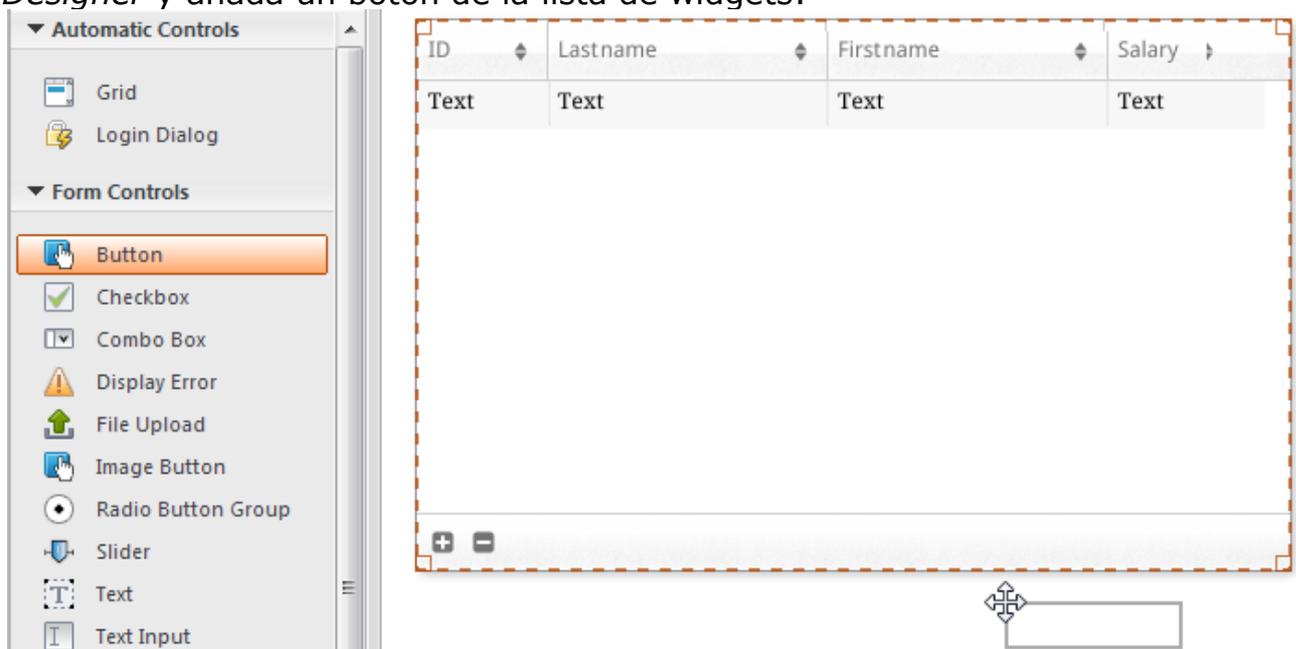
```
FIRST RECORD([Empleados])
While(Not(End selection([Empleados])))
    [Empleados]salary:=[Empleados]salary*2
    SAVE RECORD([Empleados])
    NEXT RECORD([Empleados])
End while
```

2. Defina las propiedades de las llamadas 4D Mobile del método y haga clic en **Aceptar**:



En Wakanda, los métodos clase se aplican en uno de los siguientes contextos: la entidad (registro), la colección de entidades (selección) o la clase de datos (todos los registros). Usted tendrá que especificar este contexto del lado 4D.

3. Del lado Wakanda Enterprise Studio, vuelva a la página **Index** en el *GUI Designer* y añada un botón de la lista de widgets:

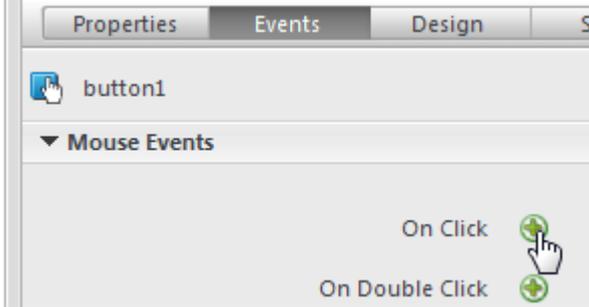


4. Haga doble clic en el botón y dele un nombre, por ejemplo, "Doblar salarios":

Double salaries

5. Asegúrese de que el botón "Doblar salarios" esté seleccionado y luego haga clic en el botón **Eventos** en el área derecha del *Diseñador GUI*.

6. Haga clic en el icono para agregar el evento "On click":



El editor de código aparece y puede introducir el código a ejecutar cuando se hace clic en el botón. Sólo vamos a llamar al método **DoubleSalary** de 4D y luego, en la función de retrollamada (*onSuccess*), todos los registros se vuelven a cargar.

7. Escriba el siguiente código:

```
sources.employees.DoubleSalary({           onSuccess:function(event){
sources.employees.allEntities();           }});
```

En el editor de código:

```
button1.click = function button1_click (event)
{
    sources.employees.DoubleSalary({
        onSuccess:function(event) {
            sources.employees.allEntities();
        });
};
```

Tenga en cuenta la minúscula "e" se utiliza para "empleados", estamos utilizando la datasource que fue creada automáticamente cuando la clase se asoció al widget.

8. Haga clic en el botón **Guardar**  en la barra de herramientas del editor.

Podemos probar la llamada al método 4D, pero de antemano que hay que volver a cargar el modelo en Wakanda Enterprise Server.

9. Haga clic en el botón **Reload Models** en la barra de Wakanda Enterprise Studio.

10. Refresque su página del navegador para que aparezca el botón **Doblar salarios** y luego haga clic en este botón:

ID	Lastname	Firstname	Salary
1	Brown	Michael	25000
2	Jackson	Maryanne	35000
3	Smithers	Jack	41000

+ - 3 item(s)

Double salaries

Se puede ver que los valores salariales se han doblado:

ID	Lastname	Firstname	Salary
1	Brown	Michael	50000
2	Jackson	Maryanne	70000
3	Smithers	Jack	82000

Tenga en cuenta que este ejemplo está pensado simplemente para mostrar cómo utilizar el conector Wakanda / 4D. Los métodos simplificados que se muestran aquí no pueden utilizarse en un contexto de producción.

Configuración de la base de datos 4D

Por razones de seguridad y rendimiento, el acceso a las tablas, datos y métodos de la base 4D a través de peticiones 4D Mobile (servidores Wakanda) debe estar activado y expresamente autorizado. Debe configurar tres niveles de acceso:

- inicio de los servicios 4D Mobile,
- control de accesos 4D Mobile (opcional pero recomendado)
- la exposición de cada objeto de la base (tabla, atributo o método de proyecto) a los servicios 4D Mobile debe ajustarse individualmente de acuerdo a sus necesidades. Por defecto:
 - Todas las tablas y todos los atributos son accesibles en 4D Mobile,
 - Los métodos proyecto no son accesibles en 4D Mobile.

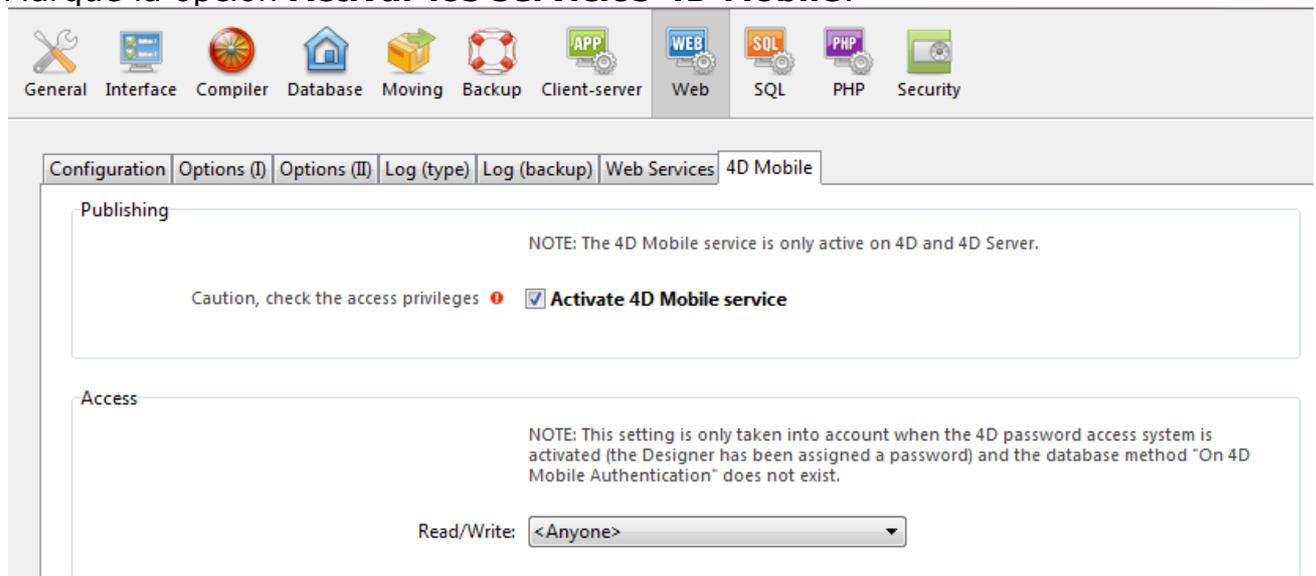
Activar los servicios 4D Mobile

Por defecto, 4D Server v14 no responde a las peticiones 4D Mobile. Debe activar los servicios 4D Mobile para que estas solicitudes sean procesadas y así se puede utilizar el conector Wakanda/4D

Nota: los servicios 4D Mobile utilizan el servidor HTTP de 4D. Por lo que debe asegurarse de que el servidor web 4D o 4D Server esté iniciado.

Para activar los servicios 4D Mobile:

1. En las propiedades de la base, vaya a la página Web/4D Mobile
2. Marque la opción **Activar los servicios 4D Mobile:**



The screenshot shows the configuration interface of 4D Designer. At the top, there is a navigation bar with icons for General, Interface, Compiler, Database, Moving, Backup, Client-server, Web, SQL, PHP, and Security. Below this, there is a sub-navigation bar with tabs for Configuration, Options (I), Options (II), Log (type), Log (backup), Web Services, and 4D Mobile. The '4D Mobile' tab is selected. The main content area is divided into two sections: 'Publishing' and 'Access'. In the 'Publishing' section, there is a note: 'NOTE: The 4D Mobile service is only active on 4D and 4D Server.' Below the note, there is a warning icon and the text 'Caution, check the access privileges'. A checkbox labeled 'Activate 4D Mobile service' is checked. In the 'Access' section, there is another note: 'NOTE: This setting is only taken into account when the 4D password access system is activated (the Designer has been assigned a password) and the database method "On 4D Mobile Authentication" does not exist.' Below the note, there is a label 'Read/Write:' followed by a dropdown menu currently set to '<Anyone>'. The 'Web Services' tab is also visible in the sub-navigation bar.

Se muestra el mensaje de advertencia "Atención, revise los privilegios de acceso" para llamar su atención sobre el hecho de que cuando los servicios 4D Mobile estén activos, por defecto el acceso a los objetos de la base es libre,

mientras no se haya definido control sobre el acceso 4D Mobile (vía REST) (ver abajo).

Controles de acceso 4D Mobile

El control de accesos 4D Mobile le permite autorizar o no la apertura de una sesión del lado 4D tras una petición Wakanda.

Como parte de un acceso 4D Mobile, los identificadores controlados son el nombre y la contraseña enviados durante la petición de conexión efectuada por:

- la caja de diálogo "Connect to Remote Datastore" de Wakanda Enterprise Studio
- los métodos SSJS [mergeOutsideCatalog\(\)](#), [openRemoteStore\(\)](#) o [addRemoteStore\(\)](#).

A nivel global, hay dos maneras de controlar los accesos 4D Mobile:

- ya sea automáticamente , por medio de contraseñas 4D,
- mediante programación a través del **Método base On 4D Mobile Authentication**.

Estos dos modos de control son excluyentes: si se define **Método base On 4D Mobile Authentication**, el control de accesos automáticos por contraseñas de 4D se desactiva.

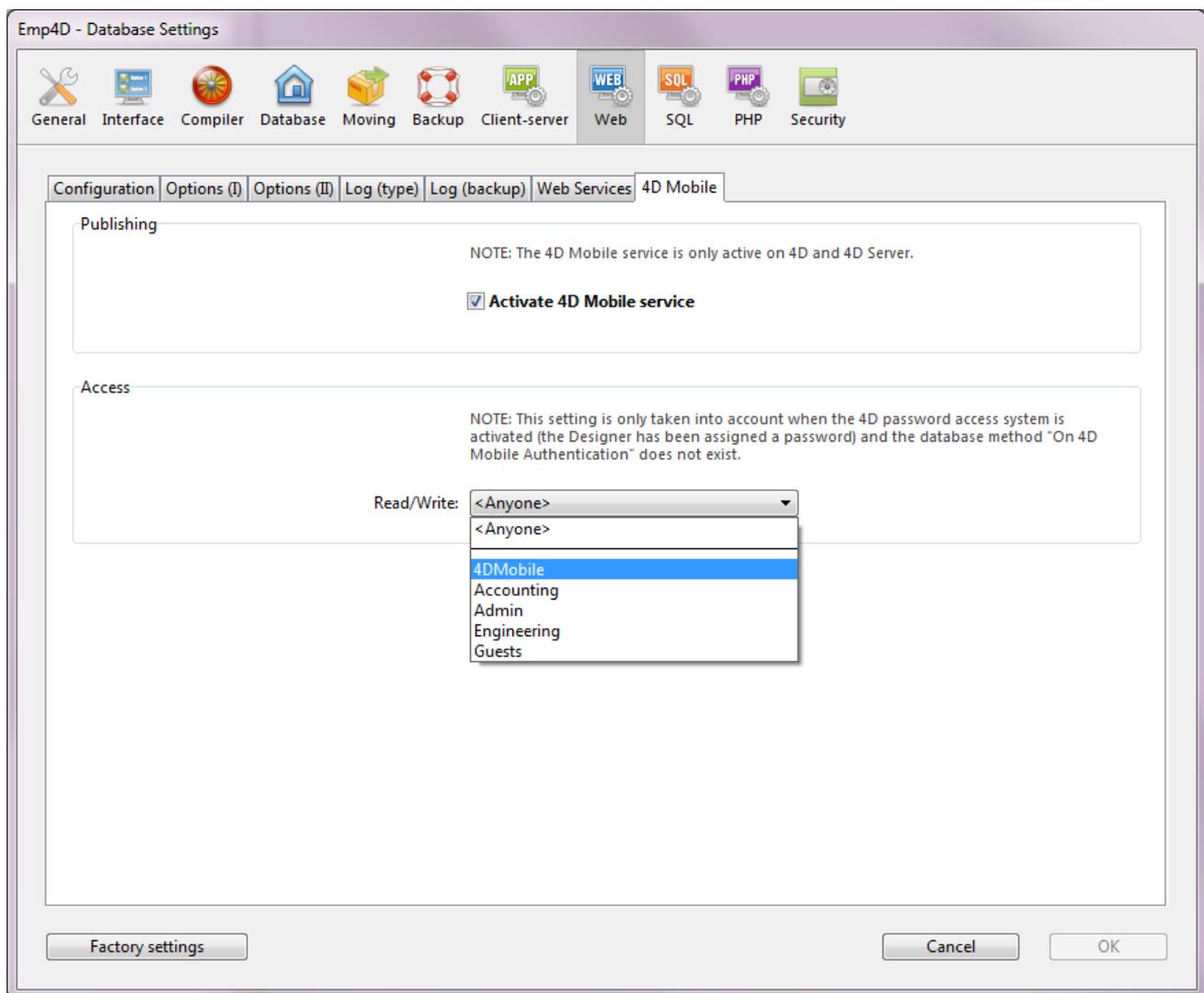
Advertencia: si ninguno de estos dos modos de control están activos, los accesos a la base de datos a través de 4D Mobile siempre se aceptan (no recomendado).

Controles automáticos por contraseñas 4D

En 4D, puede especificar el grupo de usuarios 4D que está autorizado para establecer el enlace al servidor 4D de la aplicación Wakanda

Para designar la cuenta de inicio de sesión:

1. En las Propiedades de la base, muestre la página Web/4D Mobile.
2. Seleccione el grupo a utilizar en el menú "Lectura/Escritura" del área de acceso:



Por defecto, el menú muestra **<Cualquiera>**, lo que significa que los accesos 4D Mobile está abiertos a todos los usuarios.

Una vez que haya especificado un grupo, sólo una cuenta de usuario 4D que pertenezca a este grupo se puede utilizar para acceder a 4D por medio de una petición Wakanda, en particular, para abrir una sesión en el servidor 4D utilizando el método **mergeOustideCatalog()** por ejemplo. Si se utiliza una cuenta que no pertenece a este grupo, 4D devuelve un error de autenticación al remitente de la solicitud.

Tenga en cuenta que para que esta configuración surta efecto:

- el sistema de contraseñas de 4D debe estar activo (se debe haber asignado una contraseña al Diseñador),
- **Método base On 4D Mobile Authentication** no debe haber sido definido. Si existe, 4D no tiene en cuenta los valores de acceso definidos en las propiedades de la base.

Utilizar el método base On 4D Mobile Authentication

El **Método base On 4D Mobile Authentication** permite controlar de forma personalizada la apertura de las sesiones 4D Mobile en 4D. Cuando está definido, es llamado automáticamente por 4D o 4D Server cuando una petición 4D Mobile es recibida por el servidor HTTP.

Cuando la solicitud de apertura de sesión 4D Mobile proviene de Wakanda Server (caso general), los identificadores de conexión están en el encabezado de la solicitud. El método base On 4D Mobile Authentication se llama para que pueda

evaluar estos identificadores. Puede utilizar la lista de usuarios de la base 4D o puede utilizar su propia tabla de usuarios.

Para mayor información, consulte la descripción **Método base On 4D Mobile Authentication** en el manual de *Lenguaje 4D*.

Definir los objetos 4D expuestos en 4D Mobile

Una vez que los servicios 4D Mobile están habilitados en la base 4D, por defecto una sesión 4D Mobile puede acceder a todas las tablas y campos de la base de datos, y por lo tanto utilizar sus datos. Por ejemplo, si su base contiene una tabla [Empleados], es posible escribir, del lado Wakanda Server:

```
var emp=ds.Employee.query("name == 'Martin'"); //Devuelve todos los empleados cuyo nombre sea 'Martin'
```

Nota: las tablas y/o los campos 4D que tienen el atributo "Invisible" también son expuestos en 4D Mobile por defecto.

El servidor Wakanda también puede acceder a los métodos proyecto de la base 4D. Sin embargo, por defecto, este acceso está deshabilitada por razones de seguridad.

Si desea personalizar la lista de objetos de la base accesibles en 4D Mobile, debe:

- desactivar la exposición de cada tabla y/o campo que desee ocultar,
- permitir la exposición de cada método proyecto al que quiera dar acceso.

Cuando una petición 4D Mobile intenta acceder a un recurso no autorizado (tabla o método de proyecto), 4D devuelve un error.

Exposición de tablas

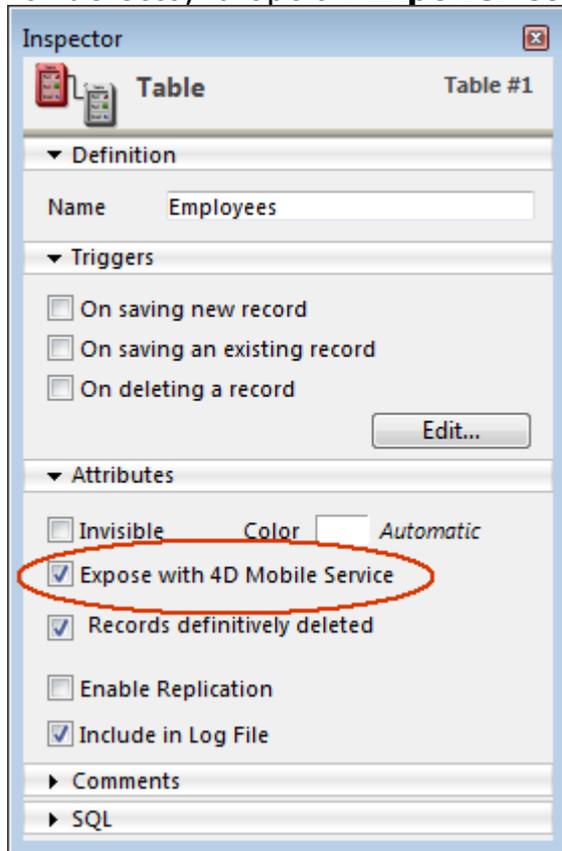
Por defecto, todas las tablas se exponen en 4D Mobile.

Por razones de seguridad, es posible que desee solamente exponer ciertas tablas de su base a llamadas 4D Mobile. Por ejemplo, si creó una tabla [Usuarios] que guarda nombres de usuario y contraseñas, sería mejor no exponerla.

Para modificar la exposición 4D Mobile de una tabla:

1. Visualice el Inspector de la tabla en el editor de estructura y seleccione la tabla a modificar

Por defecto, la opción **Exponer con el servicio 4D Mobile** está seleccionada:



2. Desactive la opción **Exponer con el servicio 4D Mobile**.

O

Marque la opción para exponer una tabla.

Haga esto para cada tabla cuya exposición necesite modificar.

Exposición de campos

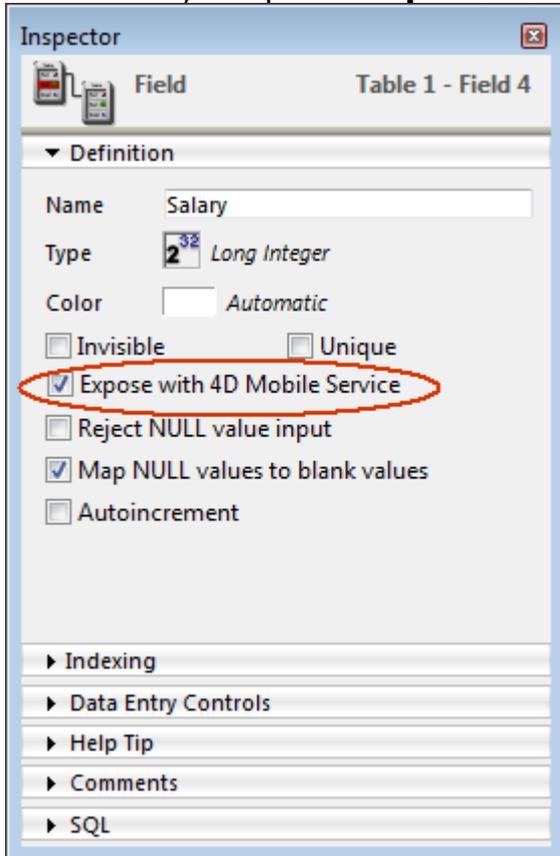
Por defecto, todos los campos están expuestos en 4D Mobile.

Puede que no quiera exponer ciertos campos de las tablas a través de 4D Mobile. Por ejemplo, puede que no desee exponer el campo [Empleados]Salario.

Para modificar la exposición 4D Mobile para un campo:

1. Visualice el Inspector del campo en el editor de estructura y seleccione el campo a modificar.

Por defecto, la opción **Exponer con el servicio 4D Mobile** está marcada:



2. Desmarque la opción **Exponer con el servicio 4D Mobile** para el campo.

O

Marque la opción para exponer un campo que no estaba seleccionado.

Repita esto para cada campo cuya exposición necesite ser modificada.

Tenga en cuenta que para que un campo sea accesible por 4D Mobile, su tabla también debe serlo. Si la tabla no está expuesta, ninguno de sus campos lo estará, sin importar su estado. Debido a esto, usted puede temporalmente activar/desactivar la exposición 4D Mobile de una tabla, mientras que los valores individuales de la exposición de cada campo se mantienen sin cambios.

Exposición de métodos proyecto

Por defecto, no hay métodos proyecto expuestos en 4D Mobile.

Es posible que desee volver ciertos métodos proyecto de su base 4D accesibles mediante 4D Mobile. Para hacerlo, debe seleccionar la opción correspondiente y definir el contexto de ejecución Wakanda del método.

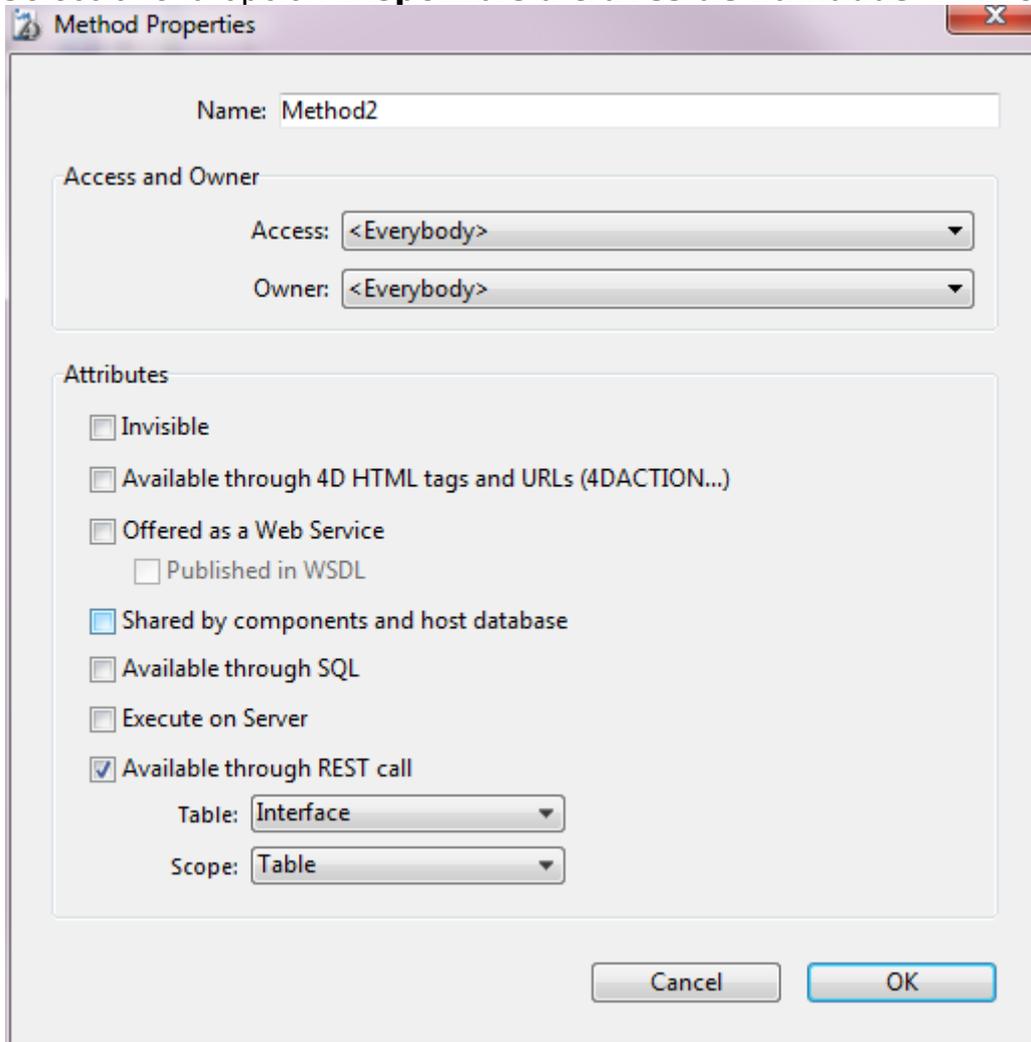
Nota: si un grupo de acceso se asocia al método 4D, debe asegurarse de que el grupo 4D Mobile es parte de este grupo.

Para definir la exposición 4D Mobile para un método proyecto:

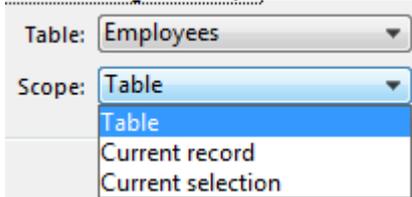
1. Abra la caja de diálogo "Propiedades del método".

Nota: puede acceder a la caja de diálogo de propiedades del método utilizando el menú contextual en la página "Métodos" del Explorador, o desde el menú Método/Propiedades del método... en el editor de métodos.

2. Seleccione la opción **Disponible a través de llamadas 4D Mobile:**



3. Defina el contexto de ejecución Wakanda del método proyecto por medio de los menús **Tabla** y **Alcance**.



Estos ajustes son necesarios con el fin de respetar la lógica de Wakanda. Para obtener más información acerca de este punto, consulte la siguiente sección.

4. Haga clic en **Aceptar** para validar los cambios.

Los métodos proyecto disponibles a través de 4D Mobile se listan en la sección "Métodos 4D Mobile" del Explorador 4D (ver el párrafo "Explorador" más adelante).

Tabla padres y alcance de los métodos proyecto

Cuando declara un método proyecto disponible a través de peticiones 4D Mobile, debe declarar explícitamente su contexto de llamada utilizando los parámetros **Tabla** y **Alcance**:

- **Tabla:** tabla a adjuntar al método proyecto. Este parámetro no está relacionado directamente a la utilización de los datos de la tabla, pero le permite designar el objeto *clase de datos* a través del cual desea acceder al método que utiliza el código JavaScript. El menú muestra la lista de tablas de la base expuestas en 4D Mobile. Si el método utiliza específicamente los datos de una tabla, puede seleccionar la tabla. Si el método no está relacionado a una sola tabla, puede utilizar

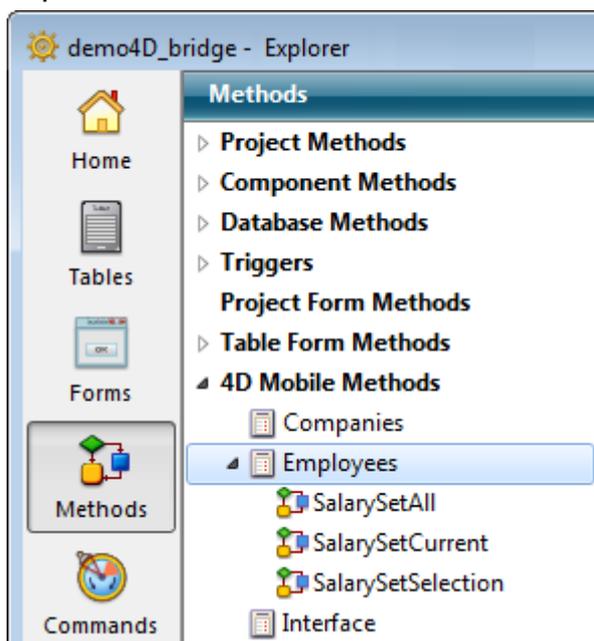
cualquier tabla expuesta. O, una vez más, si sólo desea exponer los métodos correspondientes con la lógica empresarial de su aplicación 4D, puede crear y exponer una tabla dedicada, por ejemplo [4D MobileInterface] y asociar todos los métodos proyecto expuestos en 4D Mobile.

- **Alcance:** rango de registros en los que el método se aplica. Esta declaración es necesaria porque, del lado Wakanda, los métodos son propiedades de objetos JavaScript y sólo se pueden llamar a través de estos objetos. Cada método 4D que está expuesto debe estar asociado explícitamente al contexto de la base donde se va a llamar: **Tabla**, **Registro actual** y **Selección actual**.
 - **Tabla:** esta opción indica que el método 4D se ejecutará utilizando todos los registros de la tabla designada. Del lado de Wakanda, el método será llamado en un objeto del tipo *clase de datos*, por ejemplo *ds.MiTabla.MiMétodo*.
 - **Registro actual:** esta opción indica que el método 4D será ejecutado utilizando el registro actual de la tabla designada. Del lado Wakanda, el método será llamado en un objeto de tipo *Entidad*, por ejemplo *ds.MiTabla.all().MiMétodo*.
 - **Selección actual:** esta opción indica que el método 4D será ejecutado utilizando la selección actual de registros de la tabla designada. Del lado de Wakanda, el método será llamado en un objeto del tipo *Colección de entidades*, por ejemplo *ds.MiTabla.all().MiMétodo*.

Atención: cuando cambia la exposición o el alcance de un método proyecto del lado 4D, hay que volver a cargar el modelo remoto del lado Wakanda para que estos cambios se tengan en cuenta.

Explorador

Cuando los servicios 4D Mobile están habilitados, las tablas expuestas en 4D Mobile y los métodos proyecto adjuntos a ellas se muestran en la página "Métodos" del Explorador 4D en la sección Métodos 4D Mobile:



Configuración de la aplicación Wakanda

Del lado Wakanda Enterprise, puede conectarse a una base 4D:

- utilizando la caja de diálogo "Connect to Remote Datastore" (disponible en Wakanda Enterprise Studio),
- o utilizando el método JavaScript (`mergeOutsideCatalog()`, `openRemoteStore()` o `addRemoteStore()`).

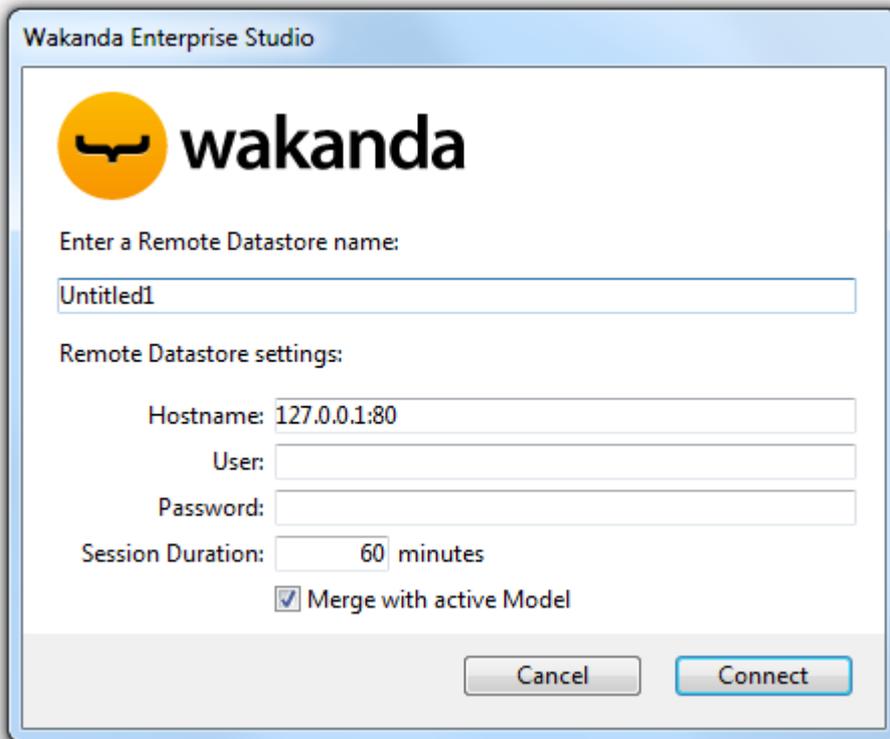
Una vez se establece la conexión entre Wakanda y 4D, la aplicación Wakanda puede utilizar las tablas, atributos y métodos proyecto expuestos de la aplicación 4D como objetos locales.

También puede utilizar este método para ejecutar código JavaScript adicional, por ejemplo, puede modificar localmente propiedades de atributos remotos o extender las clases, o añadir atributos calculados.

Uso de la caja de diálogo Connect to Remote Datastore

En Wakanda Enterprise Studio, el comando **Connect to Remote Datastore...** (que se encuentra en el menú **File** y en el menú contextual del proyecto) abre un enlace con un *datastore* remoto. Este *datastore* remoto puede ser una base 4D u otra aplicación Wakanda. En ambos casos, el servidor HTTP del *datastore* remoto debe iniciarse para que Wakanda Enterprise Studio pueda acceder al modelo remoto. Una vez definido el enlace, se restablece automáticamente cada vez que la aplicación se abra con los parámetros de conexión guardados en el archivo ".waRemoteConfig" (ver abajo).

Cuando seleccione el comando **Connect to Remote Datastore...**, aparece la caja de diálogo de conexión:

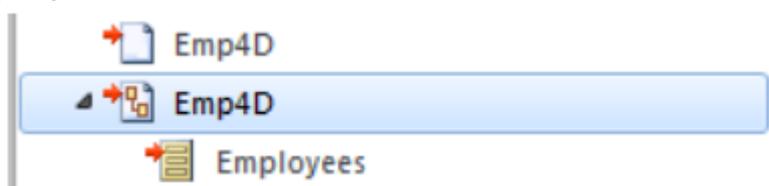


Contiene los parámetros de conexión siguientes:

- **Remote datastore name:** nombre local del catálogo remoto, que se muestra en el Explorador de soluciones. Si no selecciona la opción **Merge with active Model**, este nombre se utiliza como id del datastore en lugar de "ds" (ver abajo). En este caso, asegúrese de utilizar caracteres compatibles (ver [Programming and Writing Conventions](#) en la documentación Wakanda).
- **Hostname:** dirección del servidor de datos remoto (utilice HTTPS para mayor seguridad)
- **User y Password:** nombre de usuario y contraseña para abrir la sesión 4D Mobile en la base 4D
- **Session duration:** número de minutos (60 por defecto) para mantener la sesión conectada a la base 4D remota. Este parámetro no se tiene en cuenta si la conexión está abierta con un usuario y una contraseñas no vacías (la protección de acceso 4D Mobile del lado 4D es muy recomendada).
- **Merge with active Model** (opción seleccionada por defecto): fusionar el *datastore* remoto con el modelo activo del proyecto (objeto **ds**), de modo que las *datastore* clases remotas se incluyan en la lista de clases del GUI Designer de Wakanda. Para obtener más información, consulte **Integrar al modelo activo o utilizar un modelo dedicado**.

Archivos de parámetros

Cuando se establece una conexión entre Wakanda y 4D Server a través de la caja de diálogo "Connect to a Remote Datastore", Wakanda Enterprise Studio crea automáticamente dos archivos (iconos con una flecha roja) en la carpeta del proyecto:



- el primer archivo (extensión ".waRemoteConfig") guarda los parámetros de conexión definidos en la caja de diálogo,
- el segundo archivo (extensión ".waRemoteModel") contiene la representación local del modelo del *datastore* remoto. Su contenido se puede mostrar (pero no modificar) en la ventana del editor de modelos de Wakanda.

Nota: puede ver las extensiones de los archivos en un mensaje de ayuda que aparece cuando se seleccionan los archivos en el Explorador de Wakanda Studio.

Uso de un método JavaScript

Wakanda Enterprise Server le permite establecer un enlace con una base 4D mediante la ejecución de un método JavaScript. El método de conexión por lo general debe ubicarse en el código que se ejecuta cuando se abre la aplicación (*bootstrap.js*) o cuando el modelo se abre (*model.js*) para que el enlace esté disponible durante cada sesión.

Hay tres métodos que puede utilizar para establecer un enlace 4D Mobile:

- `model.mergeOutsideCatalog()`
- `addRemoteStore()`
- `openRemoteStore()`

La principal diferencia entre estos métodos es el modo de integración de los objetos procedentes del *datastore* remoto en la aplicación Wakanda:

`model.mergeOutsideCatalog()` fusiona el catálogo remoto con el modelo activo, mientras que **`addRemoteStore()`** y **`openRemoteStore()`** generan modelos dedicados. Para obtener más información sobre este punto, consulte [Integrar al modelo activo o utilizar un modelo dedicado](#) a continuación.

Ejecución del método `mergeOutsideCatalog()`

El método JavaScript **`mergeOutsideCatalog()`** designa un catálogo de datos remoto y lo utiliza dentro de su modelo Wakanda actual. Debe llamar a este método en un archivo `.js` asociado al modelo actual y ejecutado por Wakanda Server.

Hay dos posibles sintaxis:

- Sintaxis directa:

```
model.mergeOutsideCatalog(nomLocal, dirección, usuario, contraseña);
```

- Sintaxis utilizando un objeto:

```
model.mergeOutsideCatalog(nomLocal, { hostname: dirección, user: usuario, password: contraseña, jsFile: rutaArchivoJS, timeout: minutos });
```

La ventaja de usar la sintaxis con objeto es que se puede añadir un archivo `.js` que se ejecuta después de la conexión a la base 4D. Este archivo permite modificar localmente el catálogo referenciado de la base remota.

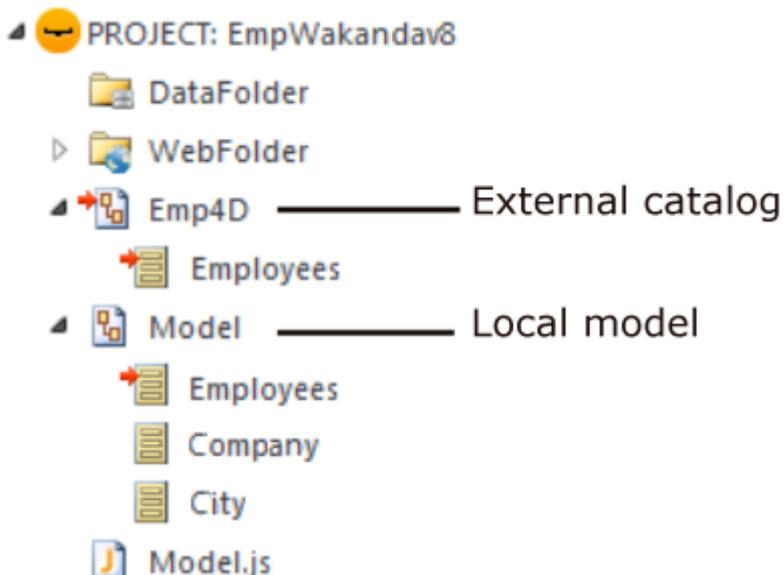
Parámetro	Tipo	Descripción
nomLocal	Cadena	Nombre local del catálogo remoto
dirección	Cadena	Dirección IP del servidor de datos remoto (utilice HTTPS para mayor seguridad)
usuario	Cadena	Nombre de usuario para la apertura de la sesión
contraseña	Cadena	Contraseña para la apertura de la sesión
jsFile	Cadena	(opcional) Ruta relativa de acceso del archivo JavaScript ubicado en la misma carpeta que el modelo (ver [#title id="3392" anchor="1284321"/])
timeout	Num	(Opcional) Tiempo de espera para conexión cliente a la base 4D en minutos (por defecto 60). Tenga en cuenta que este parámetro sólo se tiene en cuenta si la conexión está abierta con un usuario y una contraseña que no están en blanco (se recomienda especialmente para proteger los accesos 4D Mobile del lado del servidor 4D)

Para una descripción detallada, consulte la documentación del método [mergeOutsideCatalog\(\)](#) en el manual [Wakanda Server-Side API](#).

model

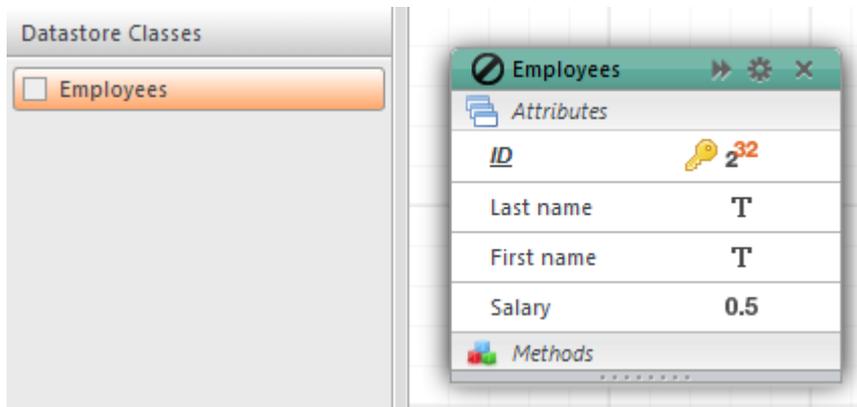
El objeto *model* indica el "modelo" actual de la aplicación Wakanda, en otras palabras, el conjunto de sus "clases de datos" (tablas) y métodos. En el contexto de una arquitectura 4D Mobile, el modelo Wakanda puede estar vacío. Si la aplicación Wakanda ya contiene los objetos, las clases y métodos referenciados de la aplicación 4D remota se fusionan al modelo local cuando utilice el método **mergeOutsideCatalog()**.

Cuando se establece la conexión correctamente, las tablas 4D "expuestas" se añaden a las clases del modelo del lado Wakanda. En Wakanda Enterprise Studio, aparecen las tablas remotas entre la lista de clases del modelo local. Los elementos externos se indican con una flecha roja. El catálogo externo también está representado en Wakanda Studio por un catálogo específico (llamado *localName.waRemoteCatalog*) que también se indica por una flecha roja:



Nota: las extensiones de archivo se pueden ocultar en Wakanda Studio.

Puede hacer doble clic en este archivo para ver el catálogo externo en el editor de modelos de Wakanda Enterprise Studio:



Ejemplo

- Ejemplo de conexión directa:

```
model.mergeOutsideCatalog("base4D","localhost:80", "admin", "123456");
```

- Ejemplo de conexión utilizando un objeto:

```
model.mergeOutsideCatalog("base4D", {      hostname: "http://localhost:8050",    user: "wak",
password: "123456",    jsFile: "Model2.js"    timeout: 15 });
```

openRemoteStore() y RemoteStore()

Los métodos **openRemoteStore()** y **addRemoteStore()** son formas alternativas de establecer vínculos dinámicos entre una aplicación Wakanda y una aplicación 4D. Como **mergeOutsideCatalog()**, estos métodos ofrecen acceso dinámico a los datos de las bases 4D pero funcionan de una manera diferente:

- pueden hacer referencia a un modelo remoto en cualquier momento durante la sesión Wakanda y no sólo cuando se carga la solución.
- tablas, atributos y métodos del modelo externo se pueden acceder por medio de un datastore separado, no se fusionaron con el modelo local de la aplicación Wakanda (accedida por medio del objeto **ds**).

`openRemoteStore()` sólo devuelve una referencia válida en el contexto JavaScript actual, mientras que `addRemoteStore()` mantiene la referencia durante toda la sesión.

Para obtener más información, consulte la descripción de los métodos [openRemoteStore\(\)](#) y [addRemoteStore\(\)](#) en la documentación Wakanda.

Integrar al modelo activo o utilizar un modelo dedicado

Cualquiera que sea el modo de conexión con la *datastore* 4D remota (usando la caja de diálogo "Connect to Remote Datastore" de Wakanda Studio o ejecutando un método JavaScript), debe elegir si las clases (tablas) remotas deben fusionarse con el modelo activo, o ubicarse en un modelo dedicado.

Esta elección se resume en la siguiente tabla:

Para...	fusionar al modelo activo	utilizar un modelo dedicado
Diálogo "Connect to Remote Datastore"	Seleccionar <i>Merge with active Model</i>	Deseleccionar <i>Merge with active Model</i>
Método JavaScript	<code>mergeOutsideCatalog()</code>	<code>openRemoteStore()</code> o <code>addRemoteStore()</code>

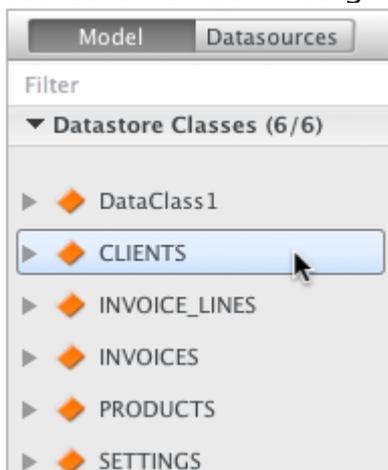
Fusionar el modelo activo

Cuando fusiona las tablas 4D remotas con el modelo activo, se integran al modelo por defecto de la aplicación (cuyo datastore es el objeto **ds**), como las clases locales. Los principios de acceso a los datos son los siguientes:

- del lado del servidor, acceda a las tablas y métodos 4D remotos por medio del objeto **ds** (ver **Llamar tablas y métodos 4D**). Ejemplo:

```
var invoiceList = ds.INVOICES.all(); //acceso a la tabla INVOICES del catálogo por defecto
```

- del lado cliente, dispone de las funcionalidades automáticas de la librería Wakanda Ajax Framework (WAF): las tablas 4D remotas están disponibles por medio de los objetos de alto nivel *datasource*, o utilizando el API *dataprovider*, que ofrece acceso de más bajo nivel.
- en Wakanda Enterprise Studio, las tablas de la base 4D se listan con las clases locales en el GUI Designer de Wakanda:



Estos principios facilitan el desarrollo de las aplicaciones 4D Mobile pero pueden dar lugar a conflictos de nombres entre las tablas, en particular cuando la aplicación Web llama a varios *datastores*. En este caso, puede ser útil para ubicar los elementos remotos en un modelo dedicado.

Uso de un modelo dedicado

Cuando las tablas 4D remotas no se fusionan con el modelo activo, utilizan un modelo "dedicado". Las clases remotas entonces utilizan un espacio de nombres que es específico para el *datastore* al que la aplicación está conectada y no se pueden acceder en el objeto **ds**. De esta manera es posible utilizar varias tablas con el mismo nombre en varias *datastores* diferentes:

- del lado del servidor, puede acceder a las tablas y métodos 4D remotos por medio de un catálogo personalizado cuyo nombre es el que usted pasa en el parámetro de conexión **Remote datastore name** (diálogo), o en *localName* (métodos JavaScript). Por ejemplo, si ha creado un enlace llamado "my4Dstore", en el código de la aplicación puede escribir:

```
var invoiceList = my4Dstore.INVOICES.all(); //access to the INVOICES table of the my4Dstore
datastore
```

Sin embargo, este principio tiene ciertas limitaciones en la versión actual de Wakanda Enterprise:

- No es posible para las aplicaciones cliente acceder directamente a las clases remotas por medio de la librería WAF o usando REST,
- las clases remotas no se listan en el *GUI Designer* de Wakanda Enterprise Studio.

Así que por lo general se recomienda elegir el modo fusionado para las *datastores* remotas, si su aplicación cliente debe acceder directamente a los datos de las tablas 4D remotas.

Modificar el archivo externo

Wakanda Enterprise le permite modificar ciertas características de la versión local del modelo externo, por personalización, optimización o seguridad.

Para ello, sólo tiene que añadir el código JavaScript en un archivo .js que tiene el mismo nombre local del catálogo más el sufijo .js y poner este archivo en la misma carpeta que el modelo. Por ejemplo, si el nombre del catálogo local es *Emp4D.waRemoteModel*, debe utilizar un archivo llamado *Emp4D.js* ubicado en la carpeta del modelo.

Notas:

- A partir de la versión 11, por defecto este archivo es creado automáticamente por Wakanda Studio.
- Cuando se establece la conexión utilizando un método JavaScript, es posible utilizar otro nombre utilizando el parámetro *jsFile*.

Wakanda ejecuta este archivo cuando inicializa el catálogo externo. Utilizando este archivo, puede:

- modificar las propiedades de los atributos de clase de datos, tales como los eventos o el alcance. Ejemplo:

```
model.className.attributeName.scope = "publicOnServer"
```

- añadir los atributos calculados a las clases de datos. Ejemplo:

```
model.className.attributeName.onGet = function() model.className.attributeName.onSet = function()
```

- añadir los atributos alias a las clase de datos. Ejemplo:

```
model.className.newAlias = new Attribute("alias", "number", "Link_15.cinteger");
```

- crear clases de datos locales derivadas de las tablas del catálogo externo, con el fin de controlar totalmente los datos enviados a los clientes. Una clase de datos derivada le permite presentar una vista personalizada de una tabla externa, mientras conserva el acceso global a la clase de datos extendida (padre) en el servidor Wakanda. Ejemplo:

```
model.DerivedClass = new DataClass("Emps", "public", "My4DTable")
```

- eliminar atributos de las clases de datos locales derivadas, por seguridad o para optimizar el tráfico de red. Ejemplo:

```
model.DerivedClass = new DataClass("Emps", "public", "My4DTable")
model.DerivedClass.removeAttribute("salary"); model.DerivedClass.removeAttribute("comments");
model.DerivedClass.removeAttribute("...");
```

Con este ejemplo, se crea una clase derivada denominada "DerivedClass", basada en la clase "My4DTable", que envía vía la red sólo los atributos que desee utilizar.

Para más información sobre el código JavaScript para trabajar con modelos, consulte la sección [Model API](#) en la documentación de Wakanda.

Definir permisos

Puede definir permisos específicos para Wakanda Server para el modelo remoto y/o en forma individual para cada clase. Para obtener más información sobre este punto, consulte la sección [Asignación de permisos de grupos](#) de la documentación Wakanda.

Llamar tablas y métodos 4D

Llamada a tablas 4D

El modo de acceso a las tablas 4D a las que se hace referencia en Wakanda depende de cómo está integrado el catálogo externo, el cual se definió en Wakanda durante su conexión a la aplicación remota (ver **Integrar al modelo activo o utilizar un modelo dedicado**):

- fusionado con el modelo activo (opción predeterminada): en este caso, las tablas remotas se usan exactamente igual a las clases locales a través del objeto **ds**.
- utilizando un modelo específico: en este caso, las tablas remotas son propiedades del objeto de modelo exclusivo.

Tablas fusionadas al modelo activo

Cuando se fusiona al modelo activo, las tablas 4D referenciadas en la aplicación Wakanda se pueden utilizar directamente en el código JavaScript del lado del servidor (SSJS) como propiedades del objeto **ds**, al igual que las datastore clases locales.

Nota: el objeto DS contiene el *datastore* actual de la aplicación Wakanda.

Por ejemplo, para realizar una consulta en los registros de la tabla 4D [Employees], puede escribir:

```
var emp = ds.Employees.query("age > :1",30);          //recuperar una colección de registros de la tabla
Employees      //donde la edad es mayor a 30 en la variable emp
```

Del lado del cliente, también puede aprovechar los mecanismos automáticos de las datasources basadas en las clases *de datos* y asociadas a los widgets. Por ejemplo, si asocia la datasource 'employees' a un widget de tipo 'Grid', puede ver la lista de empleados de forma automática:

The image shows a development environment with two main windows. The top window is a design tool showing a table widget with four columns: ID, Last name, First name, and Salary. Each column has a data type of 'Text'. A red arrow points from the 'employees' class in the 'Datasources' panel to the table widget. The bottom window is a browser showing the rendered table with three rows of data:

ID	Lastname	Firstname	Salary
1	Brown	Michael	25000
2	Jones	Maryanne	35000
3	Smithers	Jack	41000

At the bottom of the browser window, it says '3 item(s)'.

Cuando la tabla está asociada a una *datasource*, también puede acceder a sus datos a través de esta *datasource*. Por ejemplo, para ordenar la colección de registros de la *datasource* "employees", puede escribir:

```
sources.employees.orderBy("age"); //ordenar la colección de empleados por edad
```

Para obtener más información sobre cómo trabajar con las clases *datastore*, consulte la [documentación de Wakanda](#).

Tablas ubicadas en un modelo dedicado

Las tablas 4D referenciadas se utilizan en el código JavaScript del lado del servidor como propiedades del catálogo donde se ubicaron cuando se creó el enlace. El nombre de este catálogo es el que pasó en el parámetros de conexión **Remote datastore name** (diálogo de conexión de Wakanda Studio) o en *nomLocal* (métodos JavaScript).

Por ejemplo, si ha creado un enlace llamado "my4Dstore" y desea realizar una consulta entre los registros de la tabla [Employees], puede escribir:

```
var emp2 = my4Dstore.Employees.query("age > :1", 30);    // Buscar en los registros de la tabla Employees
// En el enlace "my4Dstore"
```

Nota de implementación: del lado del cliente, utilizando un modelo dedicado en la versión actual de 4D Mobile no permite el acceso a las clases remotas en el momento.

Llamada a métodos 4D

Alcance y objetos

Los métodos 4D referenciados en la aplicación Wakanda se pueden utilizar directamente en el código JavaScript como propiedades de los objetos **clase de datos**, **colección de entidades** o **entidad** en función de su alcance definido del lado 4D (ver el párrafo **Tabla padres y alcance de los métodos proyecto**). Aquí está la correspondencia entre los objetos Wakanda y el alcance de los métodos proyecto:

Alcance 4D	Objeto Wakanda
tabla	clase de datos
selección actual	colección de entidades
registro actual	entidad

Nota: los métodos 4D también se pueden llamar del lado del cliente por las fuentes de datos (ver más adelante), en este caso todos los métodos están disponibles, y la fuente de datos los aplica automáticamente a la colección actual o la entidad actual dependiendo del contexto.

Por ejemplo, si realiza una búsqueda con el método query (ver sección anterior), Wakanda devuelve una colección de entidades. Puede ejecutar cualquier método de proyecto 4D cuyo alcance declarado sea "selección actual".

Servidor o Cliente API

Los métodos 4D pueden ser llamados por código JavaScript de tres formas:

- En el código JavaScript ejecutado en el servidor (SSJS), utilizando el [SSJS Datastore API](#). En este caso, los métodos de 4D se llaman como las propiedades de los objetos **clase de datos**, **colección de entidades** o **entidad**, tal como se describió anteriormente.

Ejemplos:

```
var vTot = ds.Emp.raiseSalary(param)    //raiseSalary es una propiedad de datastore class    //el
catálogo está fusionado al modelo activo var vTot2 = my4DStore.Company.first().capital(param)
//capital es una propiedad de entidad ya que first() devuelve una entidad    //utilización del modelo
dedicado my4DStore
```

- Desde el código JavaScript ejecutado en el cliente (es decir, el navegador) utilizando el Wakanda Ajax Framework (WAF). Hay dos posibilidades:

Nota de implementación: *en la versión actual de Wakanda Enterprise, el acceso cliente a los métodos de la base 4D está disponible únicamente cuando la base remota se conecta y fusiona con el modelo activo.*

 - utilizar el [WAF Datasource API](#): este API de alto nivel ofrece muchas funciones automáticas para la gestión de datos. Con esta API, los métodos 4D se llaman como las **propiedades de las fuentes de datos asociadas a la clase de datos**, la entidad colección actual, o la entidad actual en función del contexto. Puede manejar valores de retorno de los métodos o los posibles errores utilizando la sintaxis asíncronica (requerido para el código ejecutado en el cliente). Ejemplo:

```
sources.employee.raiseSalary(param, {onSuccess: function(event) { ... //código a ejecutar cuando el método ha terminado} })
```

No es obligatorio utilizar una función de retrollamada (callback) porque los objetos tipo datasource tienen funciones automáticas que admiten, por ejemplo, la actualización de los datos que aparecen en la colección actual después de una búsqueda

- utilizar el [WAF Dataprovider API](#): este API cliente de bajo nivel le permite manipular directamente los objetos. Al igual que con el API SSJS Datastore, los métodos 4D se llaman como propiedades de **clase de datos, colección de entidades o entidad**. Sin embargo, debe gestionar los valores de retorno de los métodos o cualquier error utilizando la sintaxis asíncronica (necesaria para código ejecutado en el cliente). Ejemplo:

```
ds.Employee.raiseSalary(param, // a sintaxis se parece a una llamada SSJS {onSuccess: function(event) // pero es el código del lado del cliente por lo que debe gestionar el método de retrollamada de la llamada asíncronica { ... //código a ejecutar cuando el método 4D termina} })
```

La elección del lugar (servidor o cliente) y del API depende de las necesidades de la aplicación y se describe en la documentación de Wakanda.

Parámetros

Al igual que con los métodos estándar, usted puede pasar parámetros durante la llamada, que se reciben en el orden en los parámetros \$1,\$2, etc. Del mismo modo, el método puede devolver un resultado en la variable \$0.

Ejemplo: Quiere darle un aumento del 5% a los empleados cuyo salario es menor que 1500.

- Del lado 4D, el método **IncreaseSalary** se expuso vía 4D Mobile y su alcance es la "Selección actual". Su código es el siguiente:

```
C_REAL($1)
READ WRITE([Empleados])
FIRST RECORD([Empleados])
While(Not(End selection([Empleados])))
  [Employees]salary:=[Empleados]salary*$1
SAVE RECORD([Empleados])
```

```
NEXT RECORD([Empleados])
End while
UNLOAD RECORD([Empleados])
```

- Del lado de Wakanda, ejecute el siguiente código en el servidor:

```
var emp = ds.Employees.query("salary < :1",1500); // emp contiene la colección de empleados
cuyo salario es <1500 emp.IncreaseSalary(1.05); //ejecuta el método IncreaseSalary en la
colección //También se podría escribir: //ds.Employees.query("salary <
:1",1500).IncreaseSalary(1.05);
```

También puede devolver una selección 4D directamente como una colección Wakanda utilizando el comando **MOBILE Return selection**. Por ejemplo:

```
//método proyecto FindCountries
//FindCountries( string ) -> object

C_TEXT($1)
C_OBJECT($0)
QUERY([Countries];[Countries]ShortName=$1+"@")
$0:=MOBILE Return selection([Countries])
```

Actualización del contexto 4D

Al llamar a un método 4D por medio del enlace de Wakanda:

- Si el método se aplica a una selección (*colección de entidades*), ésta se convierte en la selección actual y 4D se ubica en el primer registro de esta selección sin cargar o activar las relaciones. Si la selección está vacía, el comando **Selected record number** devuelve 0 en lugar de 1.
- Si el método se aplica a un registro (*entidad*), éste se convierte en el registro actual. La selección actual se reduce solo a este registro y el comando **Selected record number** devuelve 1.
Nota: por razones de optimización y para evitar bloqueos innecesarios, el registro se carga en modo de sólo lectura. Sin embargo, la tabla está en modo lectura-escritura, por lo que simplemente puede llamar al comando **LOAD RECORD** para forzar la carga del registro en modo lectura-escritura siempre que sea necesario.
- Si el método se aplica a una tabla (*clase de datos*), ni la selección actual ni el registro actual se ven afectados.

Tenga en cuenta que después de ejecutar un método vía 4D Mobile, el contexto 4D se reinicializa:

- las selecciones se reducen a 0,
- los registros se desapilan y descargan,
- las selecciones locales y los conjuntos para el proceso se destruyen,
- las transacciones abiertas durante la ejecución del método se cancelan,
- la configuración de las relaciones automáticas por campos, los destinos de búsqueda o consultas en el servidor se restablecen,

- se cancelan los trabajos de impresión,
- las ventanas se cierran,
- toda conexión SQL, PHP o HTTP se cierra.

Error de alcance

Debe asegurarse de que el alcance del método 4D corresponde al tipo de objeto Wakanda que lo está llamando, de lo contrario el error *"TypeError: 'undefined' is not a function"* es devuelto por Wakanda.

Por ejemplo, dado el método 4D "getcursel" que contiene el siguiente código:

```
$O:=Records in selection([Table_1])
```

Dado el método run del lado Wakanda

```
var tt = ds.Table_1.query("Field_2 = 'a*']").getcursel();
```

El método **query()** devuelve una colección. Si el alcance del método getcursel fue definido como "Registro actual", Wakanda devuelve el siguiente error:

TypeError: 'undefined' is not a function (evaluating 'ds.Table_1.query("Field_2 = 'a']").getcursel()')*.

Uso de relaciones

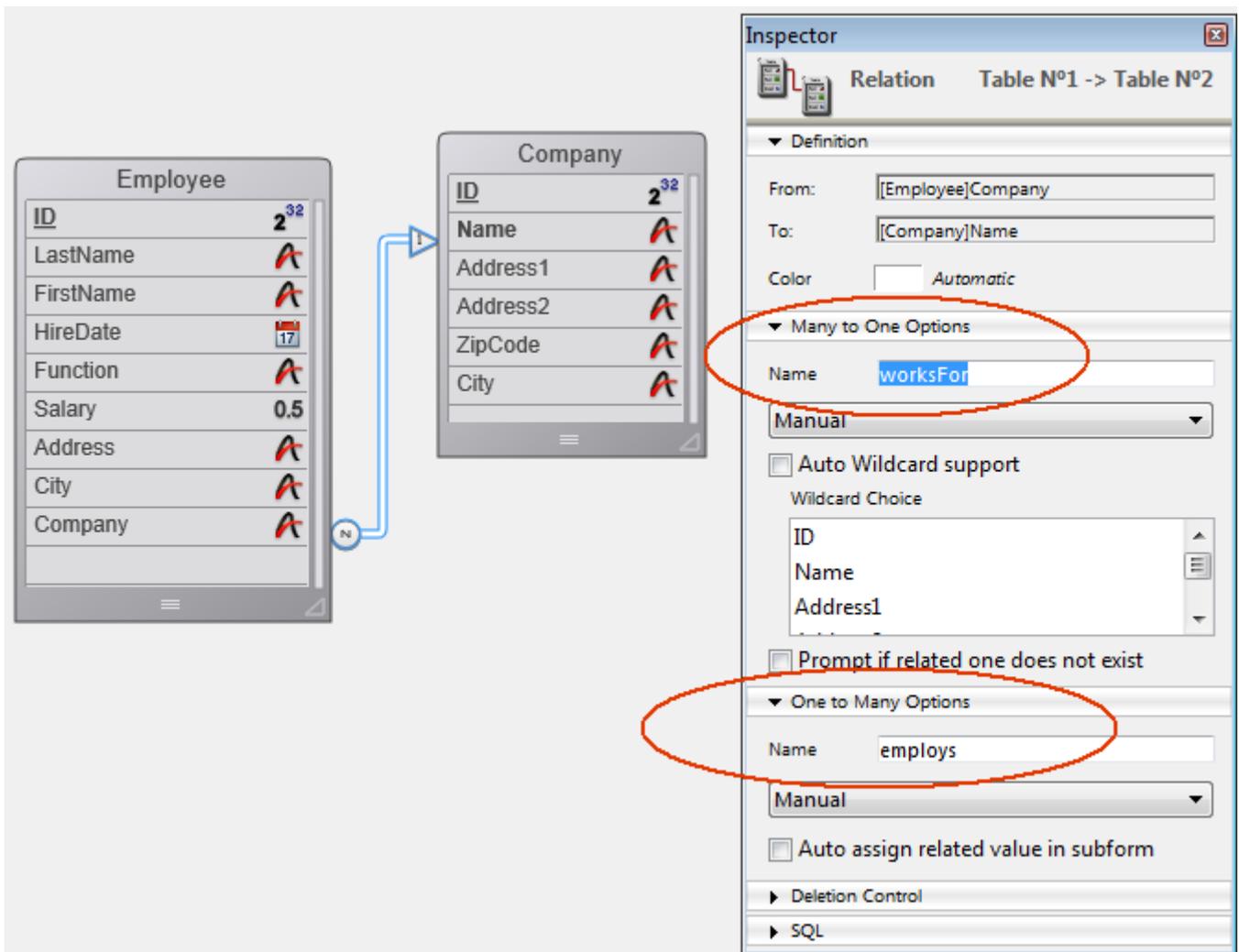
Las relaciones establecidas entre tablas 4D se utilizan de manera transparente en el contexto de un enlace 4D Mobile. Sin embargo, la forma en que estas relaciones se representan difiere en Wakanda a nivel del modelo. En el editor de modelos, las relaciones están vinculadas con atributos específicos, llamados atributos relacionales. Estos atributos se pueden utilizar para mostrar los datos vinculados directa o para llevar a cabo las consultas. Para obtener más información al respecto, consulte la sección "[Atributos](#)" en la documentación de Wakanda.

Para cada relación que se establece del lado 4D, se agregan dos atributos relacionales en la representación del modelo en la parte Wakanda:

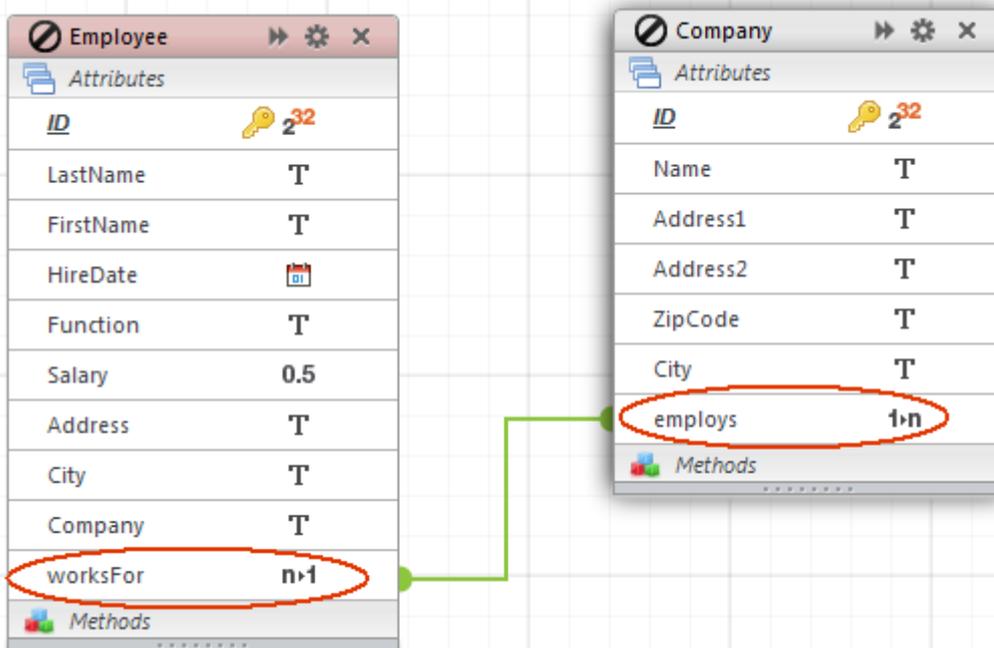
- un atributo n->1 en la tabla fuente (clase) de la relación
- un atributo 1->n en la tabla de destino (clase) de la relación.

A ambos atributos se les da el nombre de la relación, como se definió, respectivamente, para la relación muchos-a-uno y uno a muchos en el Inspector del lado 4D.

Por ejemplo, imaginemos que en el contexto de una estructura "Empleado/Empresa", crea una relación de la tabla [Empleados] a la tabla [Empresa]. Usted puede caracterizar esta relación por medio del nombre, por ejemplo, usted podría nombrar la relación muchos-a-uno "*trabajaPara*" y la relación uno-a-muchos podrían ser "*contratante*":



Del lado Wakanda, por medio de un enlace a través del conector, estas relaciones se materializan automáticamente por dos atributos relacionales adicionales, que se puede ver en el editor del modelo:

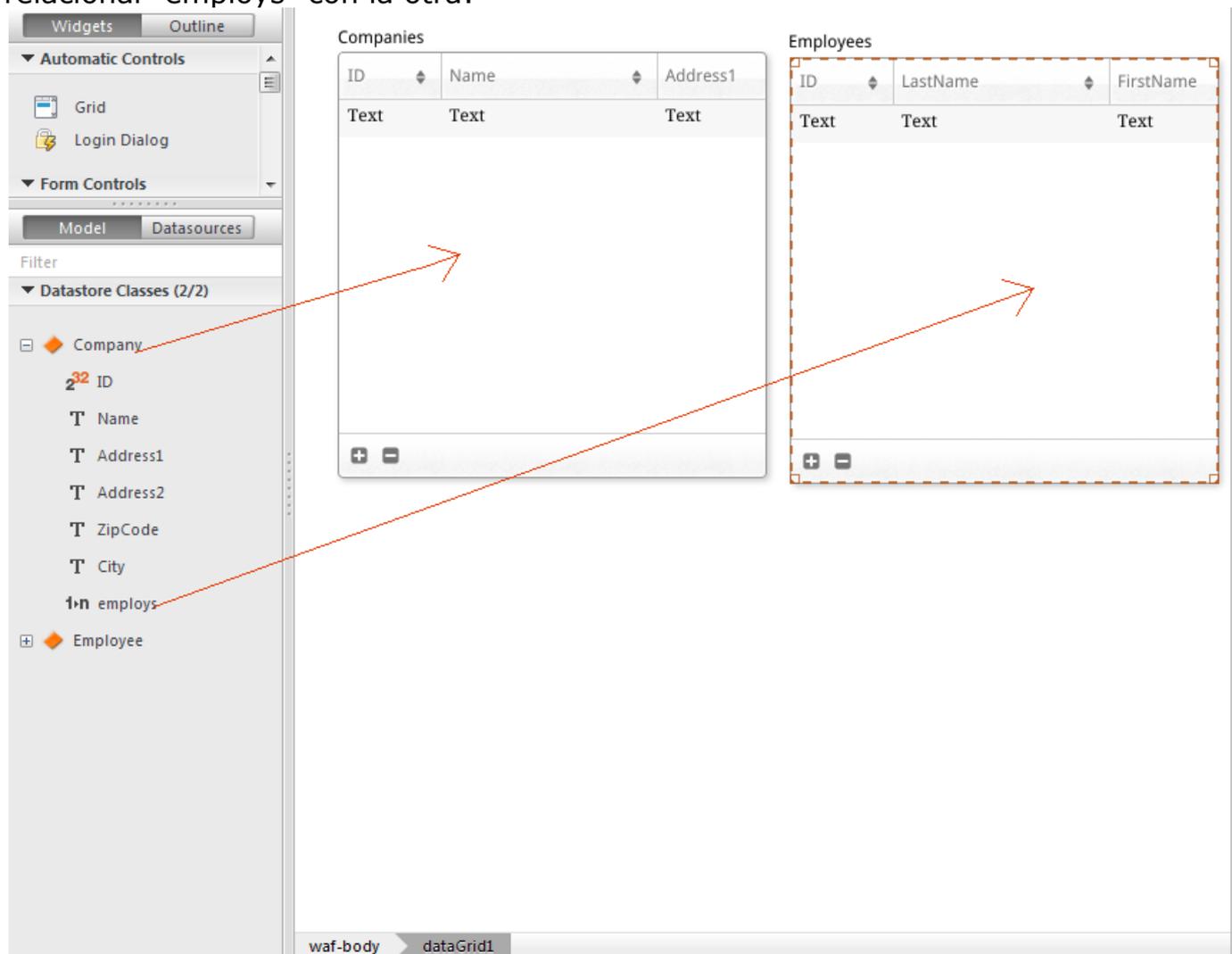


Puede darle a estas relaciones (y por tanto a sus atributos relacionales correspondientes) el nombre que desee, en base a la lógica de su aplicación.

La ventaja de esto es que es muy simple usar estos atributos del lado Wakanda para trabajar con los datos relacionados. Particularmente, puede crear widgets

relacionados con las *fuentes de datos* que se basan en atributos relacionales. Estos widgets luego se gestionan y actualizan automáticamente en función de las acciones del usuario.

Por ejemplo, es muy simple de crear una página que contiene una rejilla con la lista de empresas, y otra lista de los empleados de la empresa seleccionada. Para ello, basta relacionar las clases de datos "Company", con una rejilla, y el atributo relacional "employs" con la otra:



Las fuentes de datos correspondientes se crean automáticamente y durante la ejecución, ambas rejillas se sincronizan automáticamente:

Companies

ID	Name	Address1
1	Gizmo Computers	12332 Madis
2	Pepperson Pipes	2293 Park St
3	Solstice Systems	2332 Market
4	Carmelito Cosmetics	2369 Rodeo
5	Seaside Candies	93 Amsterda

5 item(s)

Employees

ID	LastName	FirstName
1	Parker	John
2	Jameson	Henry
3	Johnson	Susan
4	Clarkson	Claire
5	Marker	Carl

5 item(s)

📄 Gestión de sesiones 4D Mobile

Presentación

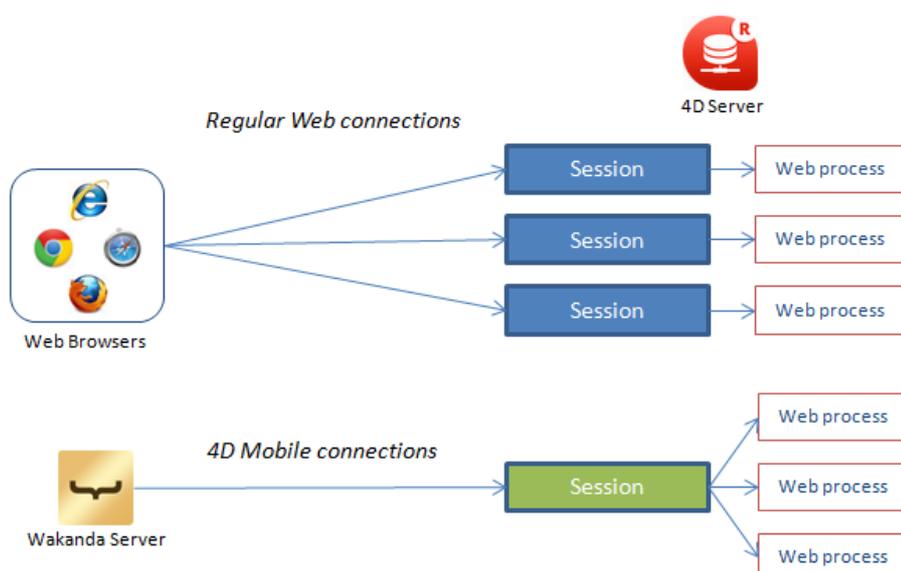
A partir de 4D v15 R4, es posible acceder por programación al ID de una sesión 4D Mobile en 4D Server. Esta funcionalidad permite a los desarrolladores obtener o definir la información relacionada con la sesión localmente (ver el ejemplo siguiente).

Las sesiones 4D Mobile pueden ser manejadas utilizando comandos estándar de sesión Web de 4D. Varios comandos web 4D, así como también el comando **WEB Get session process count** y el **Método base On Web Close Process** soportan sesiones 4D Mobile.

Sesiones 4D Mobile vs sesiones Web

Las sesiones 4D Mobile y las sesiones Web son dos tipos diferentes de sesiones. Aunque comparten algunos conceptos (y comandos), no tienen las mismas propiedades. Una diferencia significativa es la relación entre una sesión, un proceso y el contexto del proceso:

- Una sesión Web está relacionada con un sólo proceso web; gracias a la funcionalidad de Gestión automática de sesiones, el contexto del proceso (instancias de variables, selecciones, etc.) de la sesión puede ser reutilizado.
- Una sesión 4D Mobile puede estar asociada a varios procesos Web; cada contexto de proceso se reinicia automáticamente al final de la ejecución del método del proceso.



Como resultado, el intercambio de información relacionada con la sesión entre procesos Web 4D Mobile requiere de implementaciones específicas sobre 4D Server.

Comandos soportados sin modificación

Los siguientes comandos de gestión de sesiones Web existentes soportan sesiones 4D Mobile.

WEB CLOSE SESSION(sessionID)

El comando **WEB CLOSE SESSION** cierra la sesión 4D Mobile cuya identificación se pasa en *sessionID*. Como una sesión 4D Mobile puede manejar varios procesos, este comando solicita a todos los procesos Web relacionados terminar su ejecución.

WEB Get Current Session ID -> sessionID

El comando **WEB Get Current Session ID** ahora devuelve el UUID asociado con la sesión actual de 4D Mobile.

WEB GET SESSION EXPIRATION (sessionID ; expDate ; expTime)

El comando **WEB GET SESSION EXPIRATION** devuelve la información de caducidad de la cookie de una sesión 4D Mobile.

La misma cookie se utiliza para todos los procesos vinculados a una sesión 4D Mobile.

Nuevo comando WEB Get session process count

El comando **WEB Get session process count** le permite averiguar el número de procesos existentes relacionados con una sesión determinada.

- Para las sesiones Web regulares, el comando devuelve siempre 1 (una sesión Web = un proceso),
- Para las sesiones 4D Mobile, el comando devuelve todos los procesos web relacionados. El comando es útil en este contexto, por ejemplo, para ejecutar un bucle en todos los procesos de una sesión 4D Mobile.

Método base On Web Close Process (anteriormente On Web Session Suspend)

El **Método base On Web Close Process** es llamado por 4D cada vez que un proceso web está a punto de terminar su ejecución. Soporta totalmente los procesos de las sesiones 4D Mobile: en este contexto, se llama para cada proceso Web cerrado, lo que permite guardar todos los datos (variables, selección, etc.) generados por el proceso de sesión 4D Mobile.

Nota: para las sesiones web regulares, el método base **Método base On Web Close Process** se llama cada vez que la sesión Web, es decir, un proceso único de sesión Web, se cierra.

Ejemplo

Si desea compartir o reutilizar la información entre varios procesos de una sola sesión 4D Mobile, puede utilizar el UUID de la sesión 4D Mobile para identificar los datos relacionados con la sesión. Por ejemplo, después de una búsqueda de registros, usted desea mantener una selección temporal en 4D Server para que cualquier solicitud REST posterior en la misma sesión pueda acceder directamente a esta selección directa. Puede escribir, después de la instrucción de consulta:

```
//crear una selección interprocesos incluyendo el UUID de sesión  
COPY NAMED SELECTION([Emp];"<>EmpSel"+WEB Get Current Session ID)
```

//Más adelante, puede volver a utilizar esta selección desde la misma sesión
USE NAMED SELECTION([Emp];"<>EmpSel"+**WEB Get Current Session ID**)

Acerca de la seguridad de la aplicación 4D Mobile

Una vez que los datos de las tablas de las bases 4D expuestos en 4D Mobile y se integran al catálogo de la aplicación Wakanda, debe restringir el acceso a determinados recursos "sensibles".

A diferencia de las aplicaciones 4D, con las aplicaciones web no puede utilizar la interfaz para controlar los datos expuestos, por ejemplo, (sólo porque un campo no se muestre en el formulario no significa que sea inaccesible para el usuario) simplemente no mostrar un campo en un formulario no lo hace/no se puede sólo No mostrar un campo en un formulario para que sea inaccesible para el de usuario. Las peticiones HTTP y el uso de JavaScript pueden permitir que usuarios maliciosos puedan obtener potencialmente cualquier información de una protección insuficiente del servidor web.

El propósito de esta sección no es enumerar todas las medidas de seguridad a tomar con las aplicaciones 4D móviles sino ofrecerle pistas para ayudarle a asegurar los datos expuestos de una manera mínima.

- **Protección de accesos vía 4D Mobile a la base 4D:** debe controlar las solicitudes de conexión 4D Mobile. Puede utilizar:
 - Contraseñas 4D (ver [Controles automáticos por contraseñas 4D](#)),
 - El [Método base On 4D Mobile Authentication](#).
- **Control de la exposición 4D Mobile del lado 4D:** cada tabla, atributos y método pueden estar expuestos (o no) a través de 4D Mobile. Sólo exponga los datos y métodos que sean estrictamente necesarios, por ejemplo, no hay necesidad de exponer los campos no utilizados.
- **Protección de los datos expuestos:** debe utilizar los sistemas de seguridad ofrecidos por Wakanda para controlar los contenidos que son accesibles por medios de navegadores. Hay varias maneras de hacerlo (no exclusivo):
 - **Ajuste del alcance** de los atributos y métodos de la base 4D en Wakanda a nivel del modelo (consulte la propiedad alcance [para atributos](#) o [para métodos](#) en la documentación Wakanda). En particular, puede definir su alcance como **Público en el servidor**, lo que significa que pueden ser de acceso libre para el código de ejecución en el servidor, pero no van a ser accesibles en los clientes web. Este ajuste se realiza en el archivo .js del modelo externo (ver [Modificar el archivo externo](#)).
 - **Utilizar los atributos calculados:** los atributos calculados trabajan como atributos estándar, pero sus valores son devueltos por funciones específicas (**onGet, onSet...**) que se ejecutan cuando acceden a los campos. Esto significa que puede exponer sólo los atributos calculados necesarios sin exponer los campos de la base 4D. El acceso a los campos 4D se realiza desde el servidor Wakanda de una

manera segura. Puede añadir campos calculados en el archivo adicional .js que es ejecutado por el método **mergeOutsideCatalog()**. Para obtener más información, consulte la página [Attributes](#) de la documentación Wakanda.

- **Combinar clases de datos extendidas y búsquedas restrictivas**: este poderoso concepto le permite controlar no sólo los atributos expuestos, sino también los datos que pueden mostrar. Extender una *clase de datos* significa crear una copia (la clase derivada) que puede modificar mediante la adición de atributos calculados o mediante la supresión de los atributos existentes. También puede asociar una *búsqueda restrictiva* a ella: en este caso, todos los accesos a los datos de la clase derivada desencadenan automáticamente esta búsqueda, que devuelve sólo los registros que coinciden con los criterios. Este principio permite relacionar los datos al usuario conectado al servidor Wakanda. Por ejemplo, en el contexto de una base comercial, la búsqueda devuelve todos los clientes relacionados con el vendedor actual. Por supuesto, la clase derivada sólo puede ser accedida por los clientes web.

Se pueden crear *clases de datos* extendidas y añadir la *búsquedas restrictivas* en el archivo .js de configuración del modelo externo (ver **Modificar el archivo externo**). Para obtener más información, consulte la página [Programación de búsquedas restrictivas](#) en la documentación Wakanda.

Nota: se requiere la siguiente configuración mínima para soportar la restricción de consultas en 4D Mobile:

- 4D y 4D Server **v14.1**
- Wakanda Enterprise Server **v8**