

4D Mobile

Wakanda ist eine von 4D herausgegebene Plattform, um Web Applikationen vollständig mit Standardtechnologien wie JavaScript und HTML5 zu entwickeln und zu veröffentlichen.

Über die "4D Mobile" Architektur können Sie 4D und Wakanda direkt miteinander verbinden und so die grafische und funktionale Vielfältigkeit der neuesten Generation von Web Interfaces in Wakanda mit der Leistungsstärke Ihrer 4D Applikationen kombinieren.

Wenn Sie Ihre erste Verbindung zwischen 4D und Wakanda gleich einrichten wollen, prüfen Sie über den Abschnitt **Konfiguration**, ob Sie die passende Konfiguration haben und gehen dann zum Abschnitt **Schritt-für-Schritt Beispiel**.

-  4D Mobile Architektur
-  Schritt-für-Schritt Beispiel
-  Die 4D Datenbank konfigurieren
-  Die Wakanda Applikation konfigurieren
-  4D Tabellen und Methoden aufrufen
-  Verknüpfungen verwenden
-  4D Mobile Sessions verwalten
-  Sicherheit der 4D Mobile Applikation

4D Mobile Architektur

Konfiguration

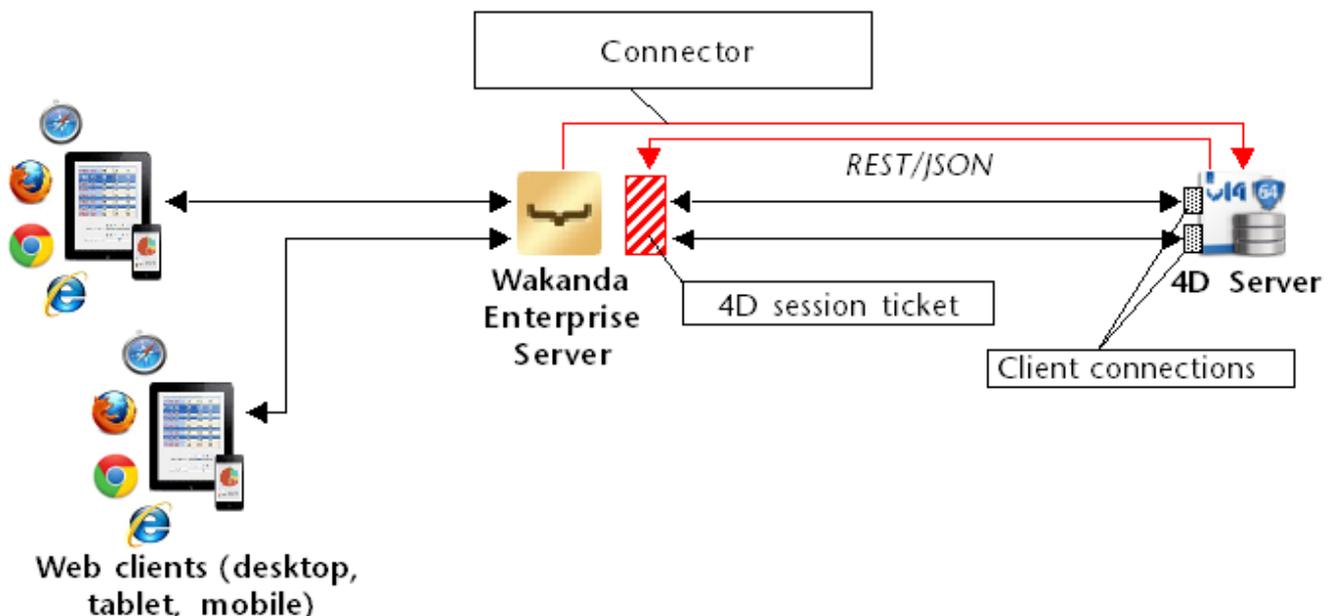
Zum Einrichten einer Verbindung zwischen 4D und Wakanda ist ab 4D v14 R3 folgende Mindestkonfiguration erforderlich:

- **4D Einzelplatz (Professional Version)**, um Ihre Lösung mit 4D Mobile zu entwickeln und zu testen - hier sind drei gleichzeitige 4D Mobile Verbindungen erlaubt.
ODER **4D Server** mit 4D Mobile Expansion Pack - hier sind zwei gleichzeitige 4D Mobile Verbindungen erlaubt.
- **Wakanda Enterprise Server** sowie **Wakanda Enterprise Studio** zur Entwicklung. Sie können beide Applikationen laden unter [Wakanda Download](#) (Registerkarte *Enterprise*).
- Eine 4D Anwendung und eine Wakanda Applikation, die miteinander kommunizieren sollen.

Auf der 4D Seite müssen Sie alle Tabellen, Attribute und Methoden, auf die die Wakanda Applikationen zugreifen sollen, entsprechend konfigurieren (siehe Abschnitt **Die 4D Datenbank konfigurieren**).

Beschreibung

Hier eine Darstellung der 4D Mobile Architektur:



Beim Starten der Wakanda Applikation stellt der Wakanda Enterprise Server eine Verbindung mit 4D Server her, gemäß den Einstellungen im Dialogfenster "Connect to Remote Datastore" oder über JavaScript Verbindungsmethoden. Hat 4D Server die Verbindung akzeptiert (siehe Abschnitt **REST Zugriffe steuern**), wird ein 4D Mobile Session "Ticket" an Wakanda Server gesendet. Wakanda Server verwendet dieses Ticket für alle nachfolgenden Anfragen.

Über diese Verbindung kann der Wakanda Server auf folgende Ressourcen in der 4D Datenbank zugreifen:

- Tabellen und ihre Attribute (inkl. Daten)
- Projektmethoden

Die autorisierten Ressourcen werden direkt auf der Wakanda Seite verwendet, so als ob sie zum lokalen Catalog der Wakanda Applikation gehören würden (Für die Wakanda Applikation ist der Zugriff transparent).

Sendet ein Web Client eine Anfrage an den Wakanda Server, der Zugriff auf die 4D Datenbank benötigt, wird diese Anfrage über das aktuelle Ticket an den 4D Server gesendet und auf dem Rechner mit 4D Server wird eine 4D Mobile Verbindung geöffnet. Sie bleibt geöffnet, solange der Benutzer Anfragen ausführt und schließt standardmäßig nach einem Timeout von 60 Minuten. Das standardmäßige Timeout lässt sich über die Verbindungsparameter verändern.

Wird während der Session die Anzahl der zugelassenen 4D Mobile Verbindungen auf dem 4D Server erreicht, erhält der Wakanda Server eine Fehlermeldung.

Schritt-für-Schritt Beispiel

Dieser Abschnitt erläutert in einem Beispiel Schritt für Schritt die Funktionsweise von Wakanda / 4D Connector. Wir führen folgendes aus:

- Eine 4D Anwendung erstellen und konfigurieren
- Eine einzelne Seite der Wakanda Applikation anlegen
- Daten aus der 4D Anwendung in der Wakanda Seite anzeigen

Um das Beispiel einfach zu halten, verwenden wir die 4D Anwendung und die Wakanda Applikation auf demselben Rechner. Sie können natürlich auch mit einer remote Architektur arbeiten.

1 - Die 4D Anwendung erstellen und konfigurieren

1. Starten Sie Ihr Programm 4D oder 4D Server und legen eine neue Anwendung an.
Nennen Sie Ihre Anwendung z.B. "Ang4D".
2. Erstellen Sie im Struktureditor die Tabelle [Angestellte] und fügen Sie folgende Felder hinzu:
 - Name (Text)
 - Vorname (Text)
 - Gehalt (Lange Ganzzahl)

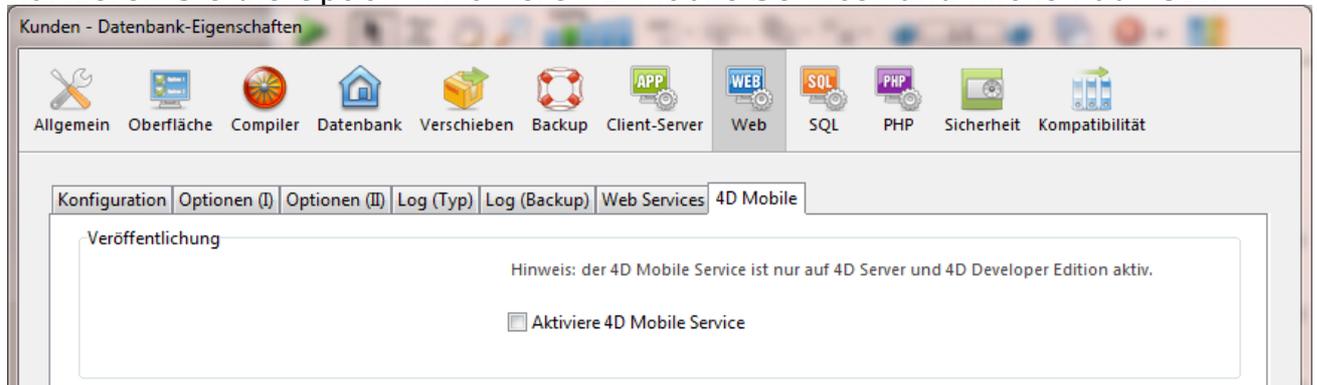


Das Attribut "Mit 4D Mobile Service veröffentlichen" ist standardmäßig für die Tabelle und jedes Feld markiert; verändern Sie diese Einstellung nicht.

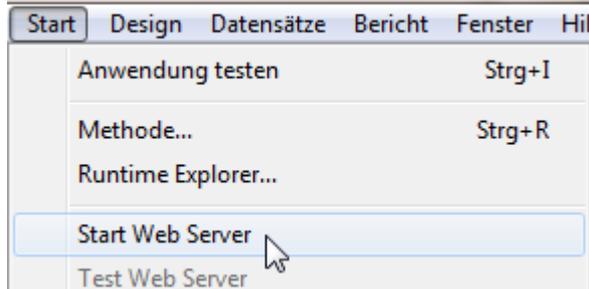
3. Wählen Sie im Menü **Datensätze > Aktuelle Tabelle anzeigen** und lassen 4D die Standardformulare anlegen. Dann geben Sie ein paar Angestellte ein:

ID :	Name :	Vorname :	Gehalt :
1	Schneider	Hans	1800
2	Müller	Anton	2100
3	Schulz	Maria	1950

- Öffnen Sie in den Datenbank-Eigenschaften die Seite **Web** und klicken auf die Registerkarte **4D Mobile**.
- Markieren Sie die Option "Aktiviere 4D Mobile Service" und klicken auf **OK**:



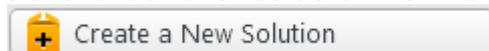
- Wählen Sie im Menü **Start** den Befehl **Start Web Server**:



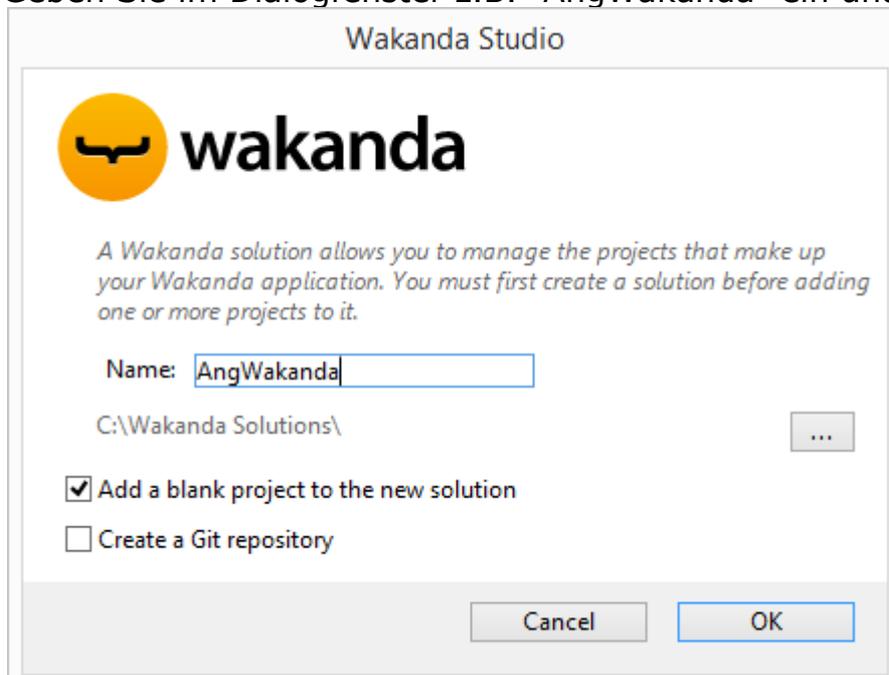
Die 4D Anwendung steht jetzt zum Empfangen von 4D Mobile Anfragen von Wakanda bereit. Beachten Sie, dass wir in diesem einfachen Beispiel die Zugriffe über 4D Mobile nicht überprüfen. In der Praxis oder in einer offenen Architektur sollten Sie 4D Mobile Zugriffe (via REST) unbedingt absichern (siehe Abschnitt **Sicherheit der 4D Mobile Applikation**).

2 - Die Wakanda Applikation erstellen

- Starten Sie die Applikation "Wakanda Enterprise Studio" und klicken auf die Schaltfläche **Create a New Solution**:

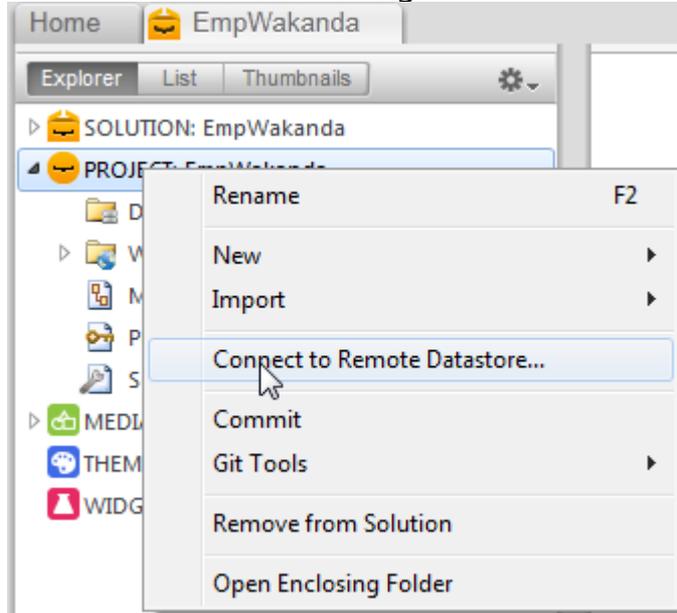


- Geben Sie im Dialogfenster z.B. "AngWakanda" ein und klicken auf **OK**:

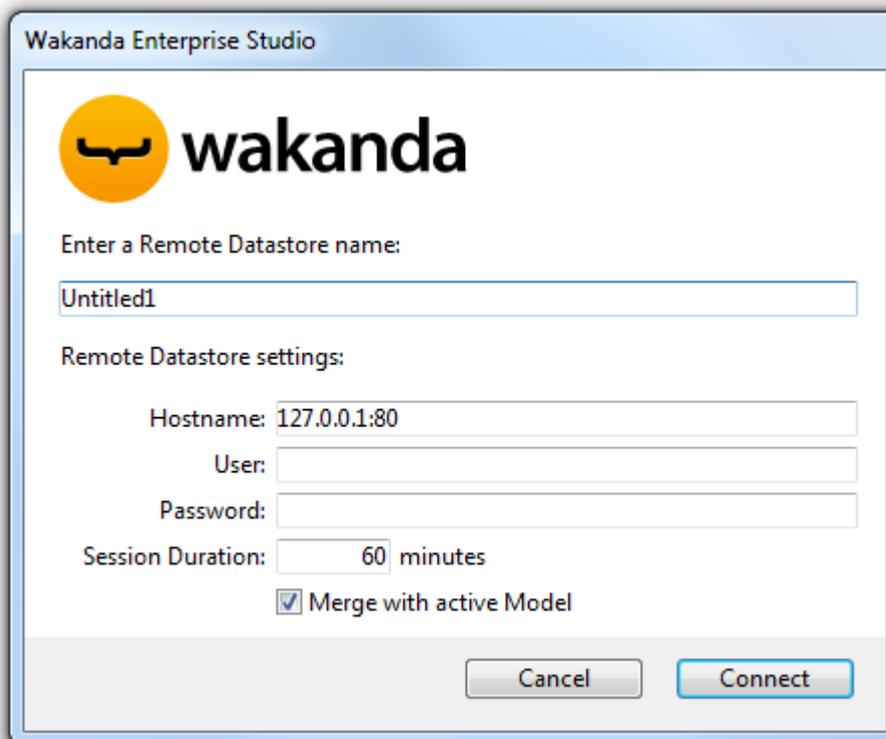


Das entsprechende Projekt wird erstellt und im Wakanda Studio Explorer erscheinen im linken Teil des Fensters die Standardelemente.

3. Klicken Sie mit rechter Maustaste auf die Zeile PROJECT und wählen im Kontextmenü den Eintrag **Connect to Remote Datastore**.

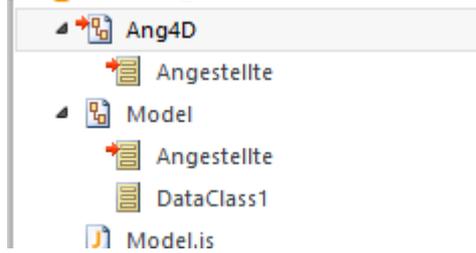


Der Verbindungsdialog erscheint:



4. Geben Sie unter **Enter a Remote Datastore name** einen Namen für den Link ein, z.B. "Ang4D":
Das ist der lokale Name des Links, wie er in Wakanda Enterprise Studio erscheint. Sie können einen beliebigen Namen eingeben. Zur Klarheit verwenden wir aber den Namen der 4D Datenbank
5. (optional) Liegt 4D Server auf einem anderen Rechner wie Wakanda Enterprise Studio, geben Sie unter **Hostname** seinen Hostnamen bzw. seine IP Adresse ein. Andernfalls können Sie die lokale Adresse "127.0.0.1:80" oder "localhost" beibehalten.
6. Behalten Sie die anderen Standardparameter bei und klicken auf die Schaltfläche **Connect**
Nach kurzer Zeit sehen Sie, dass das externe Modell "Ang4D" in den Dateien

der Wakanda Applikation erscheint, die Tabelle [Angestellte] der 4D Anwendung in den Datastore Klassen des lokalen Modells. Externe Elemente sind mit einem roten Pfeil gekennzeichnet:



Hinweis: Die erste Datei Ang4D enthält die Verbindungsparameter.

Bei Problemen...

Wird die Tabelle hier nicht angezeigt, prüfen Sie, ob:

- Kein third-party Service bzw. Software, wie z.B. instant messenger, mit dem Publishing Port des 4D HTTP Server (standardmäßig 80) in Konflikt ist,
- Auf 4D Seite der 4D Web Server gestartet ist, 4D Mobile Services aktiviert sind und die Tabelle veröffentlicht ist,
- Die im Parameter "Hostname" übergebene Adresse gültig ist.

Zum Testen, ob der 4D Server derzeit auf REST Anfragen antwortet, können Sie in Ihrem Browser folgende URLs aufrufen:

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

(gibt alle Tabellen zurück, die in 4D Mobile veröffentlicht sind)

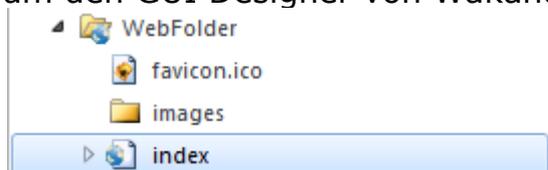
```
<address>/rest/my_table/my_method
```

(gibt das Ergebnis der Methode zurück - falls sie ein Ergebnis liefert)

3 - 4D Daten über ein Wakanda Widget anzeigen

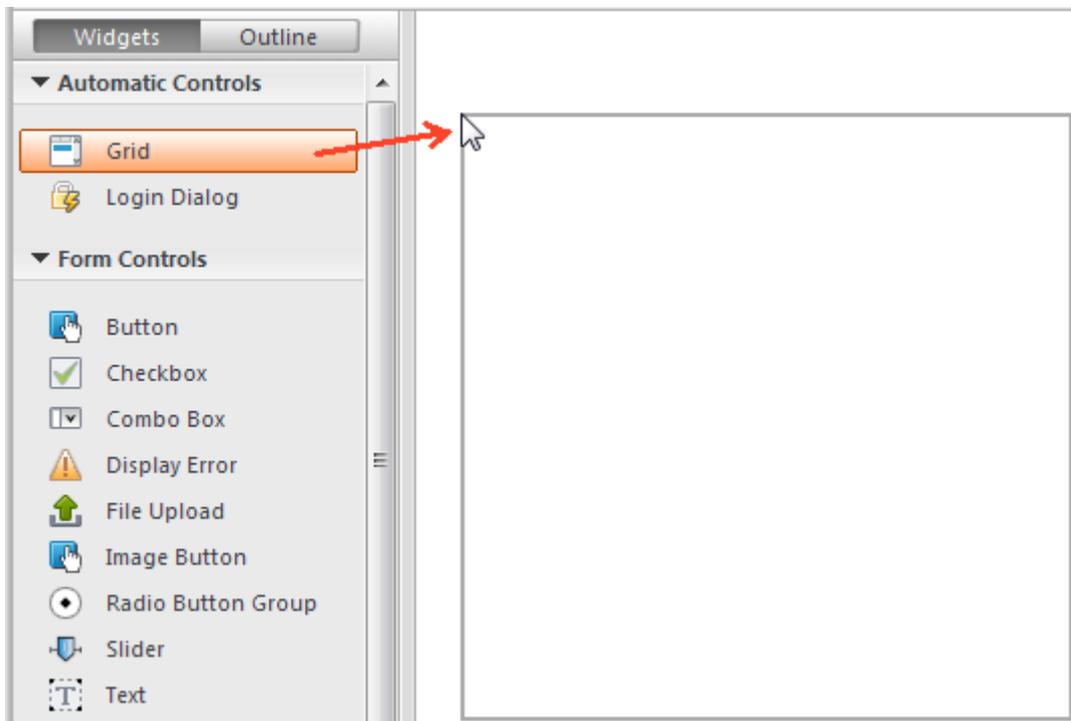
Als nächstes weisen wir im Wakanda Widget einfach per Drag-and-Drop eine 4D Tabelle zu, starten Wakanda Enterprise Server und sehen die Daten an.

1. Öffnen Sie den "WebFolder" im Explorer und doppelklicken auf die Seite Index, um den GUI Designer von Wakanda zu öffnen.

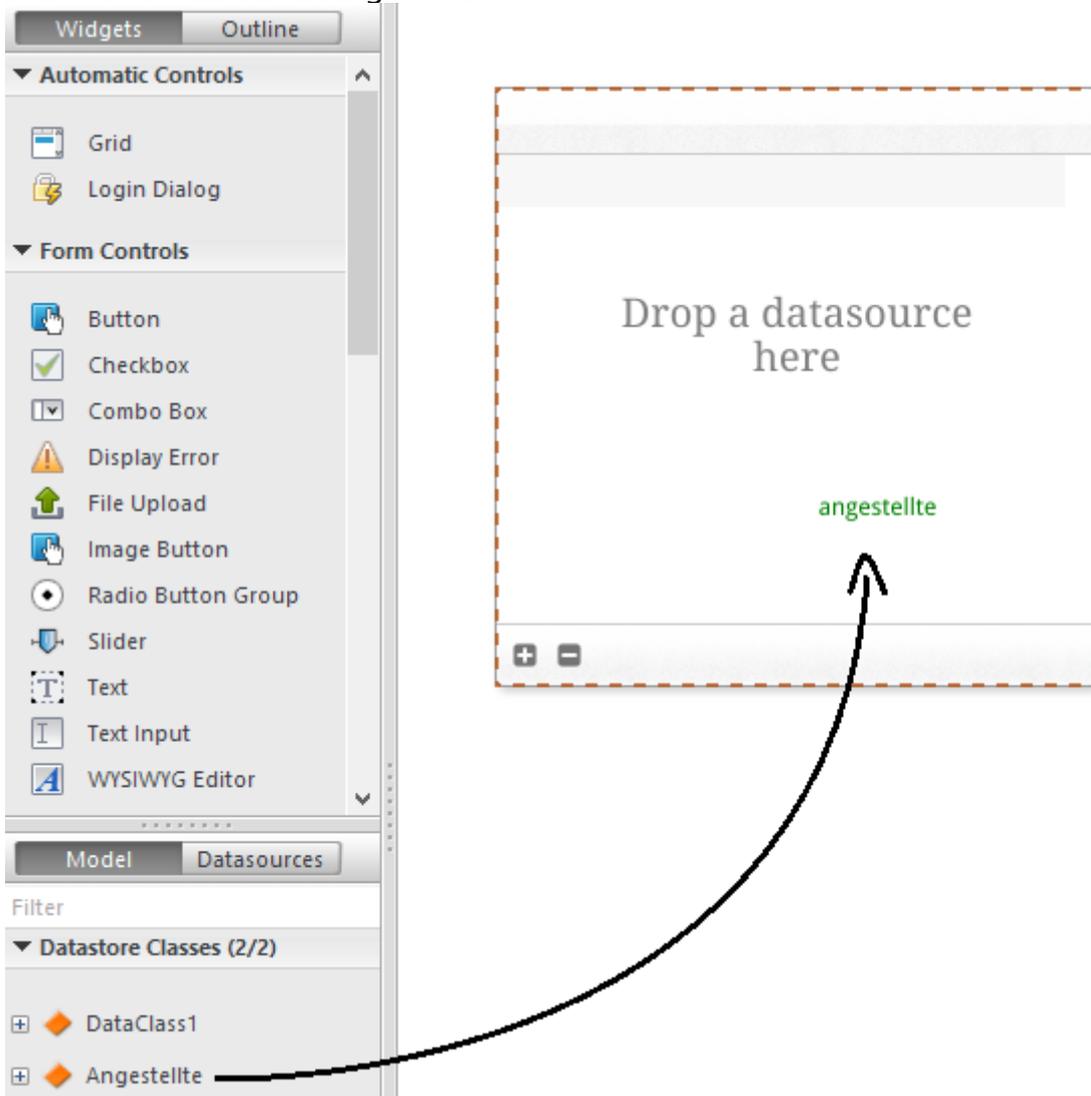


Hinweis: Der "WebFolder" enthält Elemente zur Web Publikation Ihres Projekts. "Index" ist die Standardseite des Projekts.

2. In der Liste der Widgets klicken Sie auf "Grid" und ziehen es in den Arbeitsbereich:



3. In der Liste der Datastore Klassen für das Modell klicken Sie auf "Angestellte" und ziehen es auf das gerade erstellte Grid:



Der Editor legt automatisch eine Datasource an, die auf der Klasse "Angestellte" basiert und die zur Verwaltung des Inhalts im Widget dient. Diese Datasource ist ein JavaScript Objekt mit dem Standard-namen "angestellte", das ist z.B. der Klassenname mit kleingeschriebenem ersten Buchstaben. Das Widget zeigt eine Vorschau des Inhalts. Sie können sie vergrößern, um alle

Felder der Datasource anzuzeigen:

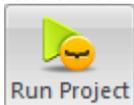
ID	Name	Vorname	Gehalt
Text	Text	Text	Text

Die Verknüpfung zwischen Datasource und Widget ist jetzt hergestellt.

4. Klicken Sie in der Toolbar des Editors auf das Icon **Save** .

Jetzt zeigen wir die Daten über einen Browser an.

5. Klicken Sie in der Toolbar von Wakanda Enterprise Studio auf das Icon:



Das startet Wakanda Enterprise Server und veröffentlicht die Anwendung "AngWakanda".

Da wir einen Link eingerichtet haben, erscheinen die Daten der 4D Anwendung in einem Fenster Ihres Standardbrowsers:



ID	Name	Vorname	Gehalt
1	Schneider	Hans	25000
2	Müller	Anton	35000
3	Weber	Maria	41000

3 item(s)

Sie können die dynamischen Eigenschaften der Verbindung testen. Machen Sie

eine Änderung auf der Web Site

ID	Name	Vorname	Gehalt
1	Schneider	Hans	25000
2	Müller	Anton	35000
3	Schulz	Maria	41000

ID :	Name :	Vorname :	Gehalt :
1	Schneider	Hans	1800
2	Müller	Anton	2100
3	Schulz	Maria	1950

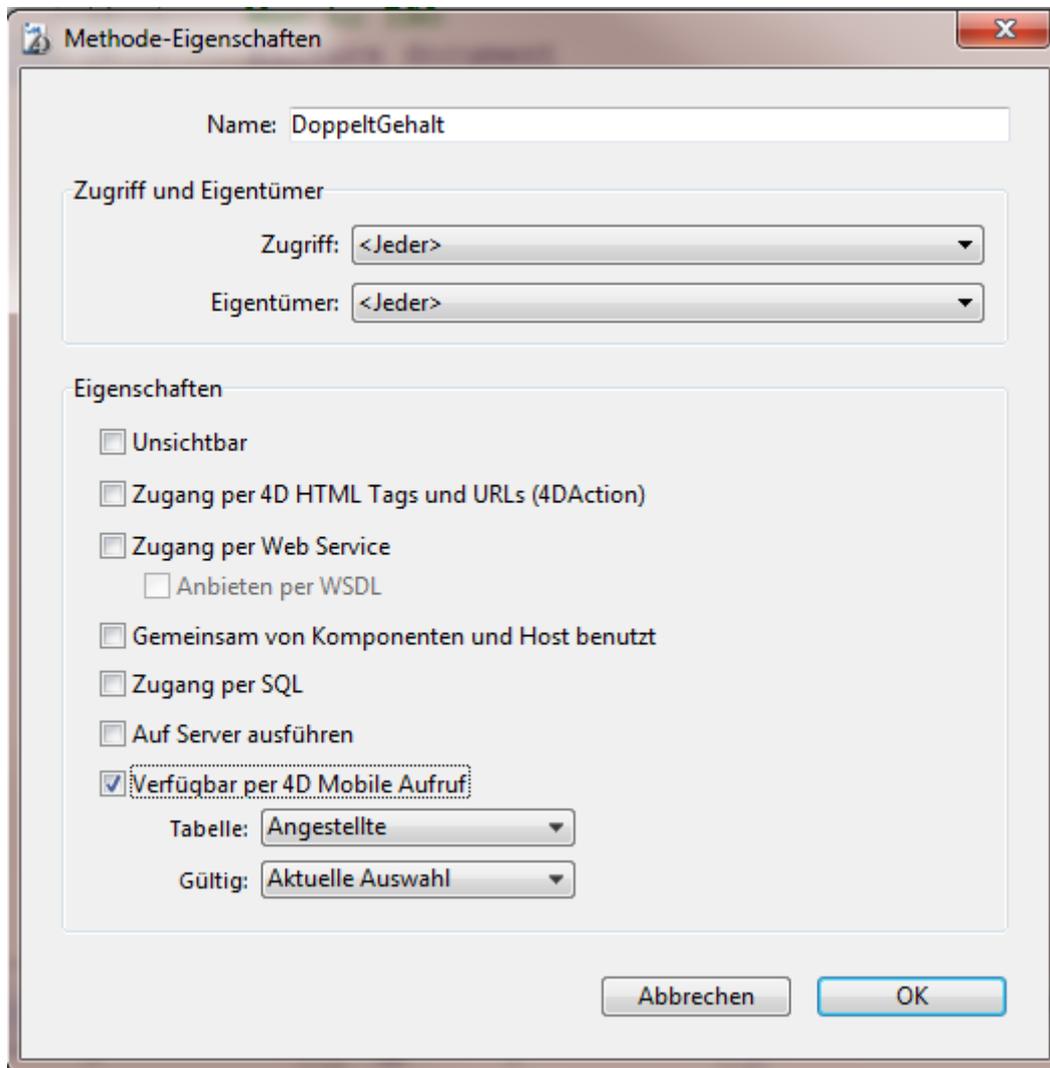
4 - Eine 4D Methode erstellen und aufrufen

Wir erstellen eine sehr einfache Projektmethode in 4D und führen sie über unsere Web Seite aus. Die Methode verdoppelt alle Gehälter.

1. In 4D erstellen Sie eine Projektmethode mit Namen **DoppeltGehalt** und schreiben folgenden Code:

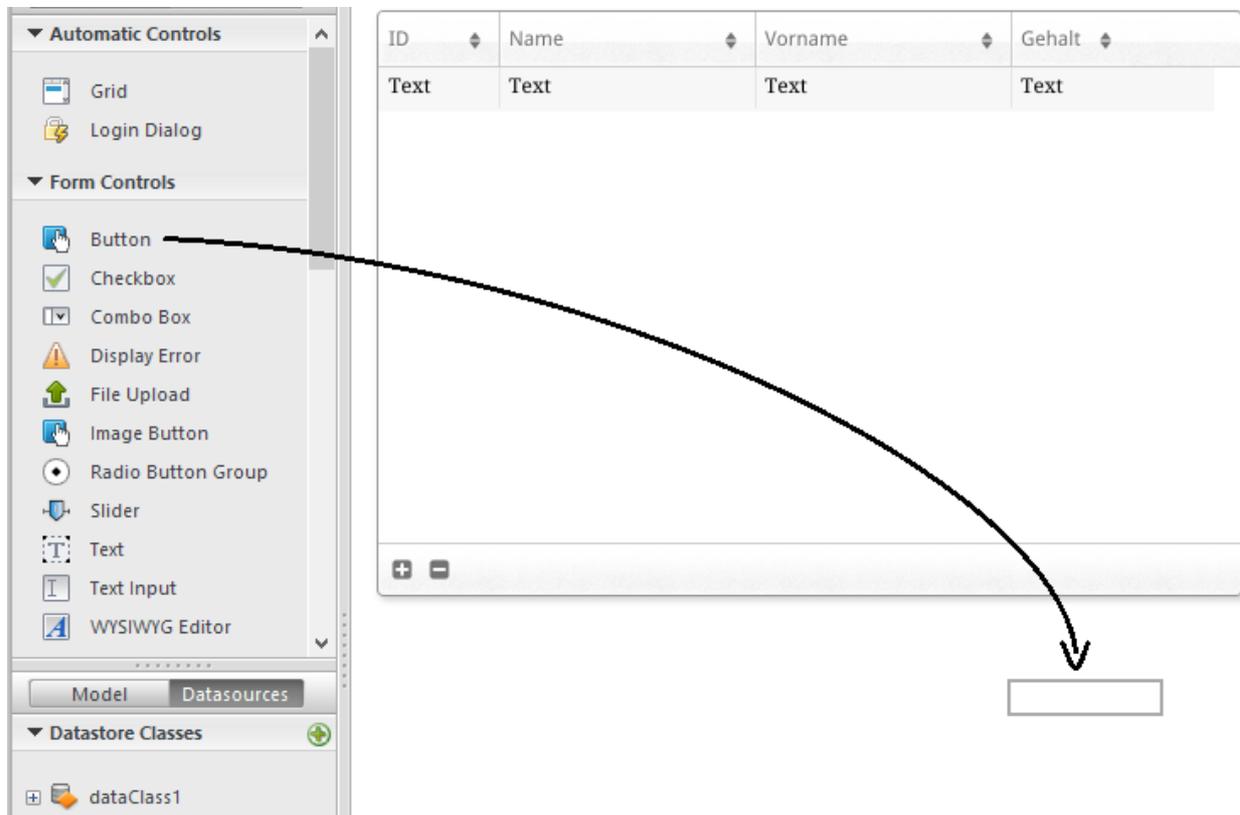
```
FIRST RECORD([Angestellte])
While(Not(End selection([Angestellte])))
    [Employees]salary:=[Angestellte]Gehalt*2
    SAVE RECORD([Angestellte])
    NEXT RECORD([Angestellte])
End while
```

2. Markieren Sie die Methodeigenschaft "Verfügbar per 4D Mobile Aufruf" und klicken auf die Schaltfläche **OK**:



In Wakanda gelten Methoden für Klassen in folgenden Kontexten: Entity (Datensatz), Entity Collection (Auswahl) oder Datastore Klasse (alle Datensätze). Das müssen Sie auf 4D Seite entsprechend definieren.

3. In Wakanda Enterprise Studio gehen Sie im GUI Designer zurück auf die Seite **Index.waPage** und fügen aus der Liste der Widgets einen Button hinzu:

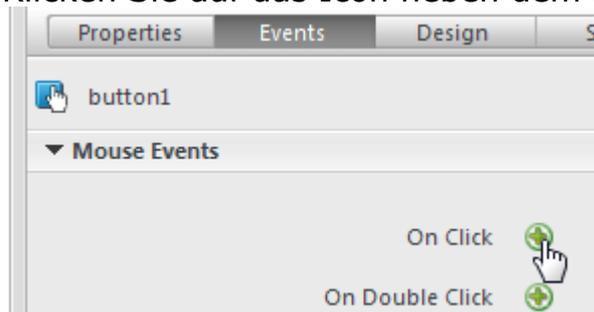


4. Doppelklicken Sie auf den Button und nennen ihn z.B. "DoppeltGehalt":

DoppeltGehalt

5. Stellen Sie sicher, dass dieser Button ausgewählt ist und klicken dann im rechten Bereich des GUI Designer auf den Button **Events**.

6. Klicken Sie auf das Icon neben dem Ereignis "On Click", um es hinzuzufügen:



Der Code-Editor erscheint. Hier geben Sie den Code ein, der bei Klick auf den Button ausgeführt wird. Wir rufen die Methode **DoppeltGehalt** von 4D auf und veranlassen dann in der Callback Funktion (onSuccess), dass alle Datensätze geladen werden.

7. Geben Sie folgenden Code ein:

```
sources.angestellte.DoppeltGehalt({           onSuccess:function(event){
sources.angestellte.allEntities();           });
```

Ihr Code Editor sieht jetzt wie folgt aus:

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

Beachten Sie das kleingeschriebene "a" für "Angestellte"; wir verwenden die

Die 4D Datenbank konfigurieren

Zur Wahrung der Sicherheit und Performance muss der Zugriff auf Tabellen, Daten und Methoden der 4D Anwendung über 4D Mobile Anfragen (von Wakanda Servern genutztes Protokoll) aktiviert und explizit autorisiert werden. Sie müssen drei Zugriffsebenen konfigurieren:

- Startup der 4D Mobile Services
- Steuerung der 4D Mobile Zugriffe (optional, wird aber empfohlen)
- Die Veröffentlichung der einzelnen Datenbankobjekte (Tabelle, Attribute oder Projektmethoden) in 4D Mobile Services gemäß Ihren Anforderungen individuell festlegen. Standardmäßig gilt:
 - alle Tabellen und alle Attribute sind für 4D Mobile zugänglich
 - Projektmethoden sind für 4D Mobile nicht zugänglich

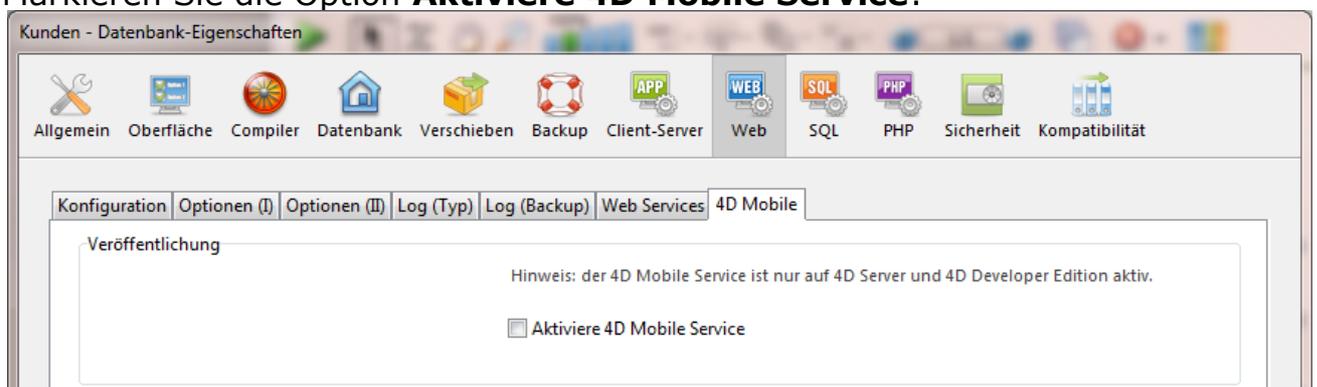
4D Mobile Services starten

Standardmäßig reagiert 4D Server v14 nicht auf 4D Mobile Anfragen. Sie müssen 4D Mobile Services aktivieren, damit solche Anfragen bearbeitet werden und Wakanda / 4D Connector eingerichtet werden kann.

Hinweis: 4D Mobile Services verwenden den 4D HTTP Server. Sie müssen also sicherstellen, dass der 4D Web Server bzw. 4D Server gestartet ist.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie in den Datenbank-Eigenschaften die Seite Web > 4D Mobile.
2. Markieren Sie die Option **Aktiviere 4D Mobile Service**:



Die Meldung "Achtung, prüfen Sie die Zugriffsberechtigungen" erscheint, um Sie darauf aufmerksam zu machen, dass durch Aktivieren von 4D Mobile Services die Datenbankobjekte standardmäßig frei zugänglich sind, solange die 4D Mobile Zugriffe (via REST) nicht gesteuert werden (siehe unten).

4D Mobile Zugriffe steuern

Durch Steuerung der 4D Mobile Zugriffe können Sie das Öffnen einer Session auf 4D Seite über eine Wakanda Anfrage erlauben oder verweigern.

Für den 4D Mobile Zugriff werden als Identifier Name und Kennwort geprüft, die während der Verbindungsanfrage gesendet werden über

- das Dialogfenster "Connect to Remote Datastore" des Wakanda Enterprise Studio
- die SSJS Methoden [mergeOutsideCatalog\(\)](#), [openRemoteStore\(\)](#) oder **addRemoteStore()**

Es gibt auf globaler Ebene zwei Wege zum Steuern von 4D Mobile Zugriffen:

- entweder automatisch über 4D Kennwörter
- oder per Programmierung über die **Datenbankmethode On 4D Mobile Authentication**

Beide Vorgehensweisen schließen sich gegenseitig aus: Ist die **Datenbankmethode 4D Mobile Authentication** definiert, ist der automatische Zugriff über 4D Kennwörter deaktiviert.

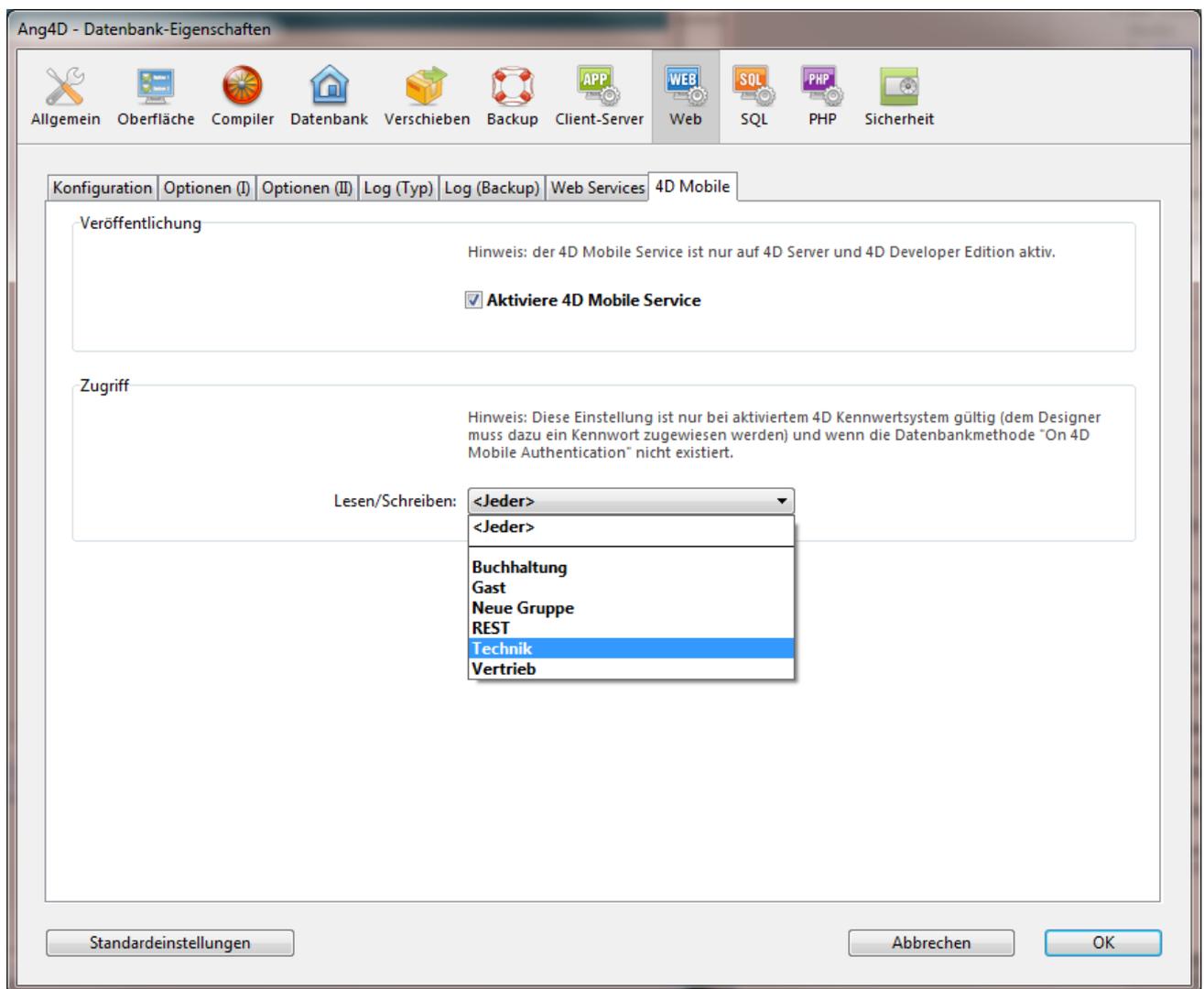
Achten Sie jedoch darauf, dass einer der beiden Wege aktiviert ist, denn sonst werden Zugriffe auf die Datenbank über 4D Mobile immer akzeptiert - was wir natürlich nicht empfehlen!

Automatische Steuerung über 4D Kennwörter

In 4D können Sie eine Gruppe 4D Benutzer definieren, die berechtigt ist, über die Wakanda Applikation die Verbindung zu 4D Server herzustellen.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie in den Datenbank-Eigenschaften die Seite Web > 4D Mobile.
2. Wählen Sie im Bereich **Zugriff** im DropDown-Menü Lesen/Schreiben die entsprechende Gruppe aus



Dieses Menü zeigt standardmäßig <Jeder> an, d.h. 4D Mobile Zugriffe sind für alle Benutzer offen.

Haben Sie eine Gruppe definiert, können nur 4D Benutzer, die zu dieser Gruppe gehören, über eine Wakanda Anfrage auf 4D zugreifen -- insbesondere, um über die Methode **mergeOutsideCatalog()** eine Session auf 4D Server zu öffnen. Wird ein Account verwendet, das nicht zu dieser Gruppe gehört, gibt 4D an den Absender der Anfrage einen Authentifizierungsfehler zurück.

Wollen Sie das 4D Kennwortsystem (der Designer muss ein Kennwort haben) verwenden, darf keine **Datenbankmethode On 4D Mobile Authentication** definiert sein. Da diese Methode vorrangig ist, berücksichtigt 4D dann nicht die in den Datenbank-Eigenschaften festgelegten Zugriffsrechte.

Die Datenbankmethode On 4D Mobile Authentication verwenden

Sie können die **Datenbankmethode On 4D Mobile Authentication** verwenden, um Zugriffsrechte für an die Web Server Engine gesendete 4D Mobile Anfragen zu steuern. Ist sie definiert, wird sie automatisch von 4D oder 4D Server aufgerufen, wenn der Server eine 4D Mobile Anfrage empfängt.

Im allgemeinen läuft die Anfrage zum Öffnen einer 4D Mobile Session über den Wakanda Server. In diesem Fall erscheinen die Identifier der Verbindung im Header der Anfrage. Die **Datenbankmethode On 4D Mobile Authentication** wird aufgerufen, so dass Sie diese Identifier bewerten können. Sie können die Liste der Benutzer für die 4D Datenbank oder Ihre eigene Tabelle mit Identifiern verwenden.

Weitere Informationen dazu finden Sie im Abschnitt **Datenbankmethode On 4D Mobile Authentication** des Handbuchs *4D Programmiersprache*.

In 4D Mobile veröffentlichte 4D Objekte setzen

Sind 4D Mobile Services in der 4D Anwendung aktiviert, kann eine 4D Mobile Session standardmäßig auf alle Tabellen und Felder der Datenbank zugreifen und so die Daten verwenden. Bei einer Datenbank mit der Tabelle [Angestellte] können Sie auf Wakanda Server Seite schreiben:

```
var emp=ds.Angestellte.query("name == 'Martin'"); //Alle Angestellten zurückgeben, deren Feld Name 'Martin' entspricht
```

Hinweis: Auch 4D Tabellen bzw. Felder mit dem Attribut "Unsichtbar" werden standardmäßig in 4D Mobile veröffentlicht.

Der Wakanda Server kann auch auf die Projektmethoden der 4D Datenbank zugreifen. Dieser Zugriff ist jedoch aus Sicherheitsgründen standardmäßig deaktiviert.

Um die Liste der in 4D Mobile zugänglichen Datenbankobjekte individuell anzupassen, gehen Sie folgendermaßen vor:

- Deaktivieren Sie die Veröffentlichung der entsprechenden Tabelle bzw. des Feldes
- Aktivieren Sie die Projektmethoden, auf die Sie Zugriff gewähren wollen

Versucht eine 4D Mobile Anfrage, auf eine nicht-zugelassene Ressource (Tabelle oder Projektmethode) zuzugreifen, gibt 4D einen Fehler zurück.

Tabellen veröffentlichen

Standardmäßig sind alle Tabellen in 4D Mobile veröffentlicht.

Aus Sicherheitsgründen wollen Sie aber nur bestimmte Tabellen Ihrer Datenbank für 4D Mobile Aufrufe veröffentlichen. Sie wollen z.B. die Tabelle [Benutzer], die Namen und Kennwörter speichert, nicht veröffentlichen.

Gehen Sie folgendermaßen vor:

1. Zeigen Sie im Struktureditor den Inspektor für Tabellen an und wählen die gewünschte Tabelle aus.
Standardmäßig ist die Option **Mit 4D Mobile Service veröffentlichen**

markiert



2. Deaktivieren Sie diese Option
ODER

Markieren Sie die Option, um eine Tabelle wieder zu veröffentlichen.

Führen Sie dies für jede Tabelle aus, deren Veröffentlichungsstatus geändert werden soll.

Felder veröffentlichen

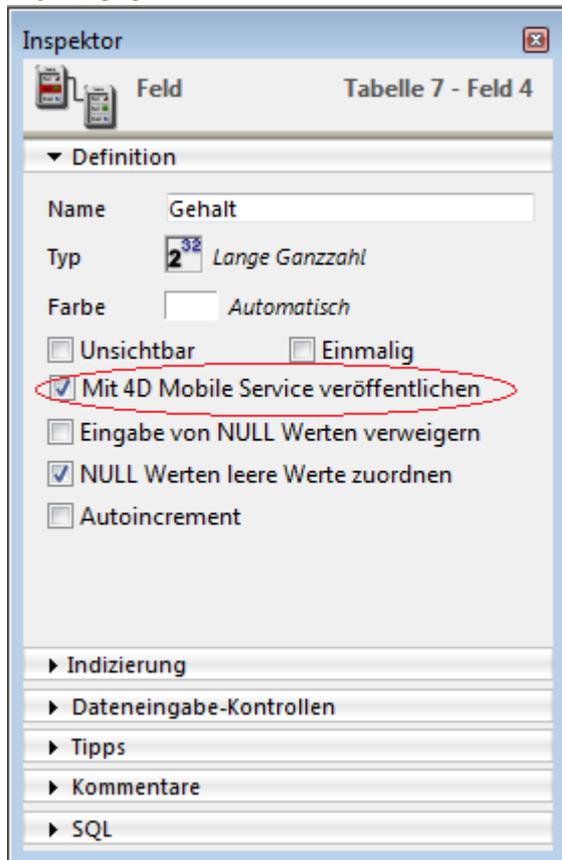
Standardmäßig sind alle Felder in 4D Mobile veröffentlicht.

Sie wollen u.U. bestimmte Felder in Ihren Tabellen nicht über 4D Mobile veröffentlichen, wie z.B. das Feld [Angestellte]Gehalt.

Gehen Sie folgendermaßen vor:

1. Zeigen Sie im Structureditor den Inspektor für Felder an und wählen das gewünschte Feld aus.
Standardmäßig ist die Option **Mit 4D Mobile Service veröffentlichen**

markiert



2. Deaktivieren Sie diese Option
ODER

Markieren Sie die Option, um ein Feld wieder zu veröffentlichen.

Führen Sie dies für jedes Feld aus, deren Veröffentlichungsstatus geändert werden soll.

Beachten Sie, dass ein Feld über 4D Mobile nur zugänglich ist, wenn das auch für die übergeordnete Tabelle gilt. Ist sie nicht veröffentlicht, gilt das auch für alle dazugehörigen Felder, unabhängig welchen Status sie haben. So können Sie die 4D Mobile Veröffentlichung für eine Tabelle je nach Bedarf aktivieren/deaktivieren und die Einstellung der einzelnen Felder unverändert lassen.

Projektmethoden veröffentlichen

Standardmäßig sind keine Projektmethoden in 4D Mobile veröffentlicht.

Wollen Sie, dass bestimmte Projektmethoden Ihrer 4D Datenbank über 4D Mobile verfügbar sind, müssen Sie die entsprechende Option markieren und den Kontext der Ausführung in Wakanda definieren.

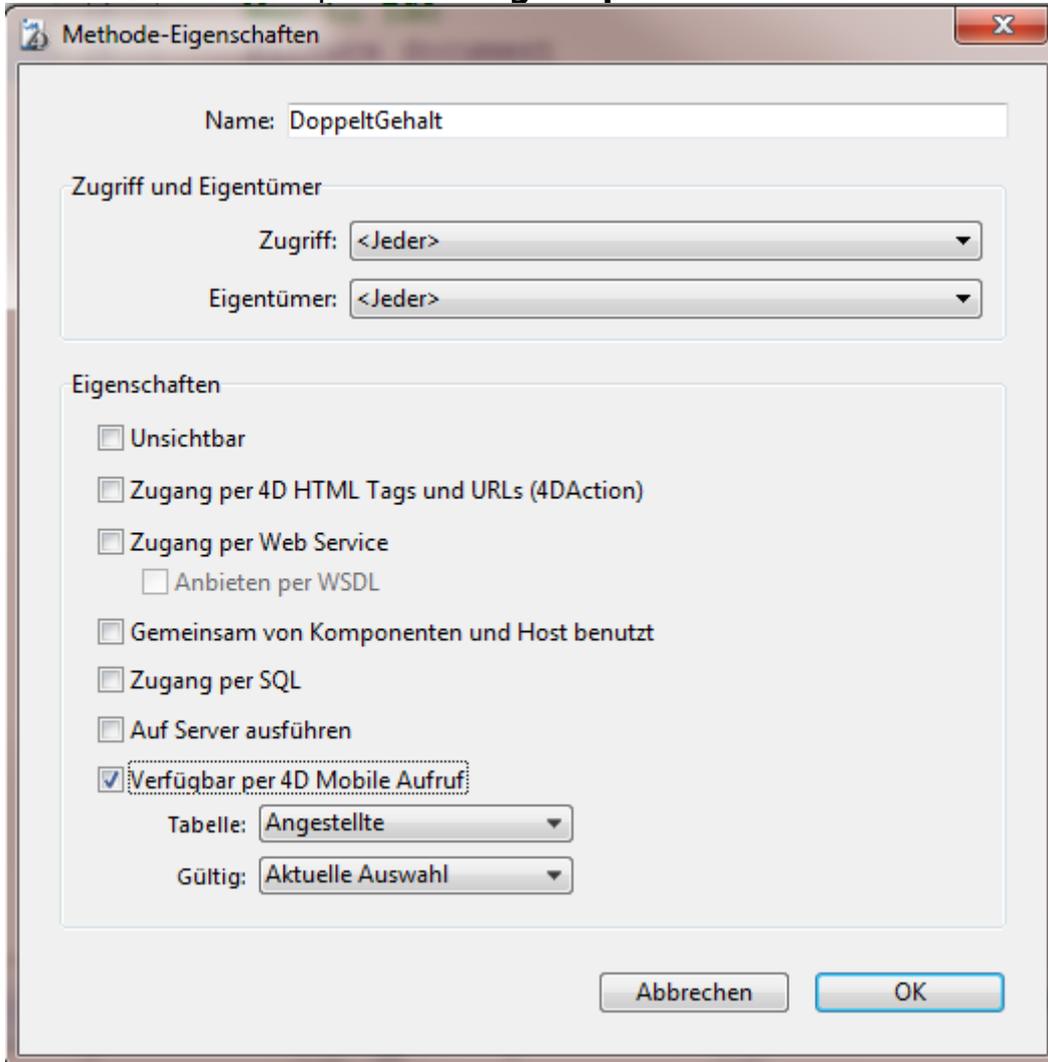
Hinweis: Ist die 4D Methode einer bestimmten Gruppe zugewiesen, müssen Sie sicherstellen, dass auch die 4D Mobile Gruppe dazugehört.

Gehen Sie folgendermaßen vor:

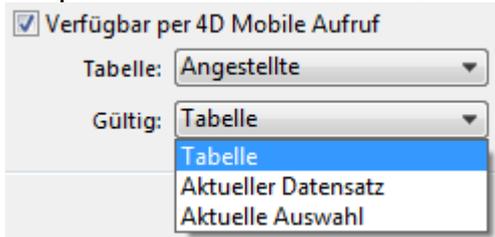
1. Zeigen Sie das Dialogfenster "Methode-Eigenschaften" an.

Hinweis: Sie können es über das Kontextmenü der Seite **Methoden** im Explorer öffnen oder über das Menü **Methode/ Methode-Eigenschaften** im Methodeneditor.

2. Markieren Sie die Option **Verfügbar per 4D Mobile Aufruf**:



3. Definieren Sie den Kontext der Ausführung für die Projektmethode über die DropDown-Menüs **Tabelle** und **Gültig**.



Diese Einstellungen sind erforderlich, damit die Logik von Wakanda beachtet wird. Weitere Informationen dazu finden Sie im nächsten Abschnitt.

4. Klicken Sie auf die Schaltfläche **OK**, um die Änderungen zu bestätigen. Projektmethoden, die über 4D Mobile verfügbar sind, erscheinen im 4D Explorer auf der Seite Methoden unter der Rubrik "4D Mobile Methoden" (siehe nächster Abschnitt "Explorer").

Tabelle und Bereich der Projektmethoden

Wenn Sie eine Projektmethode über 4D Mobile Anfragen zugänglich machen, müssen Sie über die Parameter **Tabelle** und **Gültig** explizit den Kontext deklarieren:

- **Tabelle:** Tabelle, die der Projektmethode zugewiesen werden soll. Verwendet die Methode die Daten einer bestimmten Tabelle, können Sie diese Tabelle auswählen. Ist die Methode nicht einer einzelnen Tabelle zugeordnet, können Sie jede veröffentlichte Tabelle verwenden. Wollen Sie dagegen nur Methoden veröffentlichen, die zur Geschäftslogik Ihrer 4D Anwendung passen, können Sie

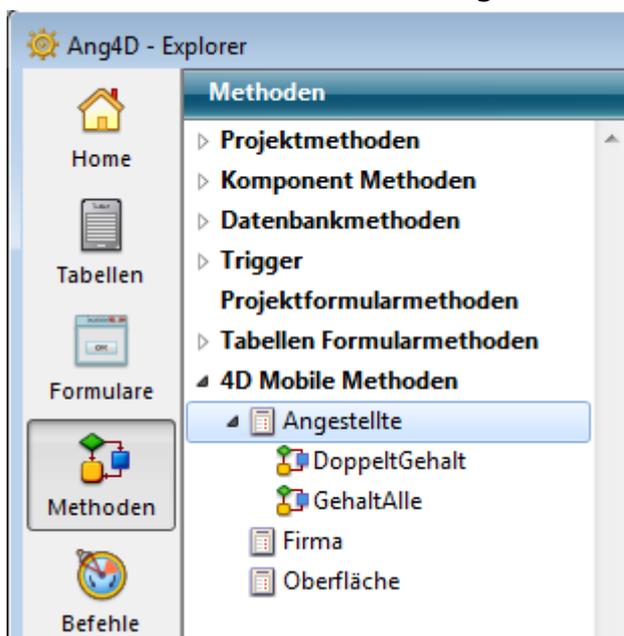
eine spezifische Tabelle, z.B. [4DMobileInterface] anlegen und veröffentlichen, und dieser alle in 4D Mobile veröffentlichten Projektmethoden zuweisen.

- **Gültig:** Datensatzbereich, auf den die Methode angewendet wird. Diese Deklaration ist erforderlich, da Methoden auf der Wakanda Seite Eigenschaften von JavaScript Objekten sind und sich nur über diese Objekte aufrufen lassen. Jede veröffentlichte 4D Methode muss explizit dem Kontext der Anwendung zugeordnet werden, in dem sie aufgerufen wird: **Tabelle, Aktueller Datensatz und Aktuelle Auswahl.**
 - **Tabelle:** Diese Option gibt an, dass die 4D Methode mit allen Datensätzen der angegebenen Tabelle ausgeführt wird.
Auf der Wakanda Seite wird die Methode in einem Objekt vom Typ *Datastore Class* aufgerufen.
 - **Aktueller Datensatz:** Diese Option gibt an, dass die 4D Methode mit dem aktuellen Datensatz der angegebenen Tabelle ausgeführt wird.
Auf der Wakanda Seite wird die Methode in einem Objekt vom Typ *Entity* aufgerufen.
 - **Aktuelle Auswahl:** Diese Option gibt an, dass die 4D Methode mit der aktuellen Auswahl der Datensätze der angegebenen Tabelle ausgeführt wird.
Auf der Wakanda Seite wird die Methode in einem Objekt vom Typ *Entity Collection* aufgerufen.

Warnung: Verändern Sie die Veröffentlichung oder den Kontext einer Projektmethode auf 4D Seite, müssen Sie das remote Modell auf der Wakanda Seite erneut laden, damit diese Änderungen berücksichtigt werden.

Explorer

Sind 4D Mobile Services aktiviert, erscheinen die in 4D Mobile veröffentlichten Tabellen und die zugewiesenen Projektmethoden im 4D Explorer auf der Seite "Methoden" unter dem Eintrag **4D Mobile Methoden**:



Die Wakanda Applikation konfigurieren

Auf der Wakanda Enterprise Seite können Sie eine Verbindung mit einer 4D Anwendung herstellen:

- entweder über das Dialogfenster "Connect to Remote Datastore" (in Wakanda Enterprise Studio),
- oder durch Ausführen einer JavaScript Methode (`mergeOutsideCatalog()`, `openRemoteStore()` oder `addRemoteStore()`).

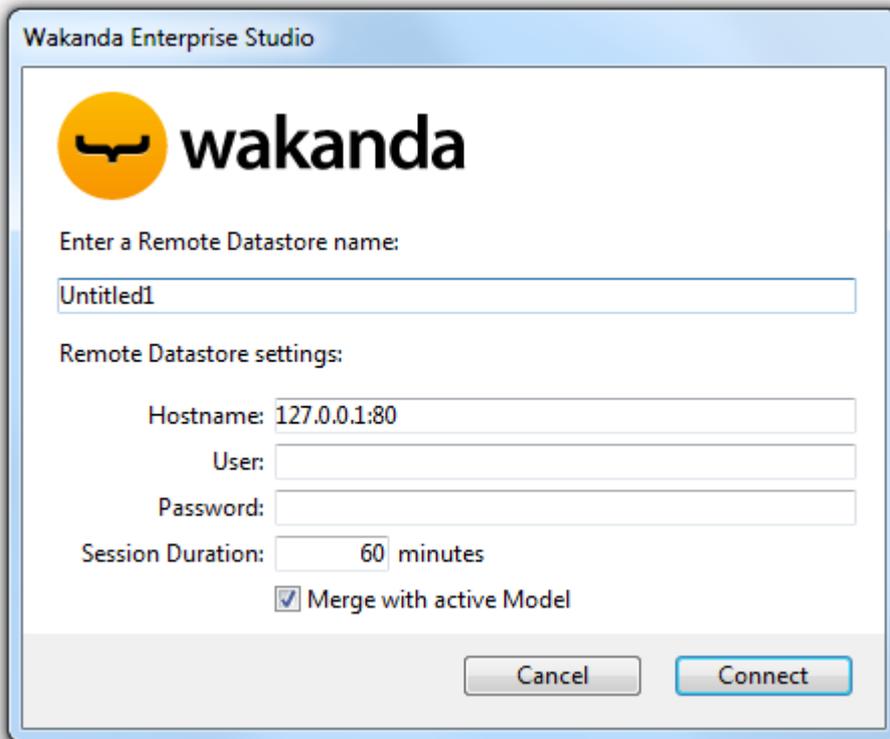
Ist eine Verbindung zwischen Wakanda und 4D hergestellt, kann die Wakanda Applikation alle veröffentlichten Tabellen, Attribute und Projektmethoden der 4D Anwendung als lokale Objekte verwenden.

Sie können auch zusätzlichen JavaScript Code ausführen. Sie könnten z.B. Eigenschaften der remote Attribute lokal verändern, Klassen erweitern oder berechnete Attribute hinzufügen.

Dialogfenster Connect to Remote Datastore verwenden

In Wakanda Enterprise Studio öffnet der Befehl **Connect to Remote Datastore...** (im Menü **File** und im Kontextmenü des Projekts) einen Link zu einem remote *Datastore*. Das kann eine 4D Datenbank oder eine andere Wakanda Applikation sein. In beiden Fällen muss der HTTP Server des remote *Datastore* gestartet werden, damit Wakanda Enterprise Studio auf das remote Model zugreifen kann. Ist der Link definiert, wird er immer automatisch wiederhergestellt, wenn die Anwendung über die Verbindungsparameter geöffnet wird, die in der Datei ".waRemoteConfig" gesichert werden (siehe unten).

Wählen Sie den Befehl **Connect to Remote Datastore...**, erscheint das Verbindungsfenster:

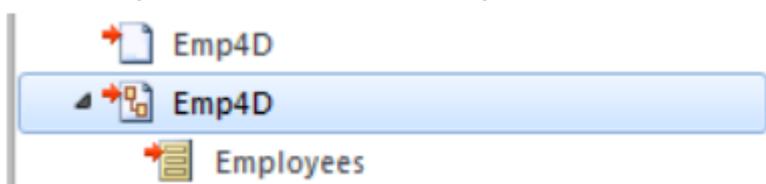


Es enthält folgende Verbindungsparameter:

- **Remote Datastore Name:** Lokaler Name des remote Catalog, angezeigt im Solution Explorer. Deaktivieren Sie die Option **Merge with active Model**, wird dieser Name als Datastore ID anstelle "ds" (siehe unten) verwendet. In diesem Fall sollten Sie auf die Verwendung kompatibler Zeichen achten (siehe [Programming and Writing Conventions](#) in der Wakanda Dokumentation).
- **Hostname:** Adresse des remote Datenserver (HTTPS für bessere Sicherheit verwenden)
- **User** und **Password:** Benutzername und Kennwort zum Öffnen der 4D Mobile Session in der 4D Anwendung
- **Session Duration:** Anzahl der Minuten (standardmäßig 60) zum Aufrechterhalten der Session, die mit der remote 4D Anwendung verbunden ist. Dieser Parameter wird nur berücksichtigt, wenn die Verbindung mit ausgefüllten Bereichen für Benutzer und Kennwort geöffnet wird (Es wird dringend geraten, 4D Mobile Zugriffe auf die 4D Seite zu schützen).
- **Merge with active Model** (Option ist standardmäßig markiert): Verknüpfen Sie den remote *Datastore* mit dem aktiven Modell des Projekts (Objekt **ds**), so dass die remote *Datastore Klassen* im Namensbereich **ds** und insbesondere in der Liste der Klassen des Wakanda GUI Designer enthalten sind. Weitere Informationen dazu finden Sie im Abschnitt **Integration in aktives model oder eigenes model**.

Parameter Dateien

Ist eine Verbindung zwischen Wakanda und 4D Server über den Befehl "Connect to a Remote Datastore" hergestellt, erstellt Wakanda Enterprise Studio automatisch zwei Dateien (Icons mit rotem Pfeil) im Ordner des Projekts:



- Die erste Datei (Endung ".waRemoteConfig") sichert die Verbindungsparameter, wie Sie im Dialogfenster definiert wurden,
- Die zweite Datei (Endung ".waRemoteModel") enthält die lokale Darstellung des Modells des remote *Datastore*. Sein Inhalt lässt sich im Fenster Wakanda Model Editor anzeigen (aber nicht verändern).

Hinweis: Sie können die Endung der Datei in einem Hilfetipp sehen. Er erscheint, wenn die Datei im Explorer von Wakanda Studio ausgewählt ist.

Eine JavaScript Methode verwenden

Mit Wakanda Enterprise Server können Sie ein Link zu einer 4D Anwendung durch Ausführen einer JavaScript Methode einrichten. Die Verbindungsmethode muss in der Regel in dem Code liegen, der läuft, wenn die Application (bootstrap.js) oder das Model (model.js) geöffnet ist, damit der Link für jede Session verfügbar ist.

Es gibt drei Methoden zum Einrichten eines 4D Mobile Link:

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

Der Hauptunterschied zwischen diesen Methoden ist die Art, wie Objekte aus dem remote *Datastore* in die Wakanda Applikation integriert werden:

model.mergeOutsideCatalog() verbindet den remote Catalog mit dem aktiven Model, während **addRemoteStore()** und **openRemoteStore()** eigenständige Models generieren. Weitere Informationen dazu finden Sie im unteren Abschnitt **Integration in aktives model oder eigenes model**.

Methode `mergeOutsideCatalog()` ausführen

Die JavaScript Methode **mergeOutsideCatalog()** bestimmt einen Catalog mit remote Daten und verbindet ihn mit Ihrem aktuellen Wakanda Modell. Sie müssen diese Methode in der .js Datei aufrufen, die dem aktuellen Modell zugewiesen ist und von Wakanda Server ausgeführt wird.

Es gibt zwei Syntaxarten:

- Direkte Syntax:

```
model.mergeOutsideCatalog(localName, address, user, password);
```

- Syntax mit einem Objekt:

```
model.mergeOutsideCatalog(localName, { hostname: address, user: userName, password: password, jsFile: jsFilePath, timeout: minutes });
```

Die objektbezogene Syntax hat den Vorteil, dass Sie eine Datei .js hinzufügen können, die nach der Verbindung mit der 4D Anwendung ausgeführt wird. Diese Datei kann den Catalog, auf den die remote Anwendung zugreift, auf lokaler Ebene verändern.

Parameter	Typ	Beschreibung
localName	String	Lokaler Name des remote Catalog
ipAddress	String	Adresse des remote Data Servers (HTTPS für zusätzliche Sicherheit verwenden)
userName	String	Benutzername zum Öffnen der Session
password	String	Kennwort zum Öffnen der Session
jsFilePath	String	(optional) Relativer Pfadname der JavaScript Datei, die im gleichen Ordner wie das Model liegt (siehe Externe Datei ändern)
timeout	Num	(optional) für Client Verbindung zur 4D Anwendung in Minuten (standardmäßig 60)

Weitere Informationen dazu finden Sie unter der Methode [mergeOutsideCatalog\(\)](#) im Wakanda Handbuch *Server-Side API*.

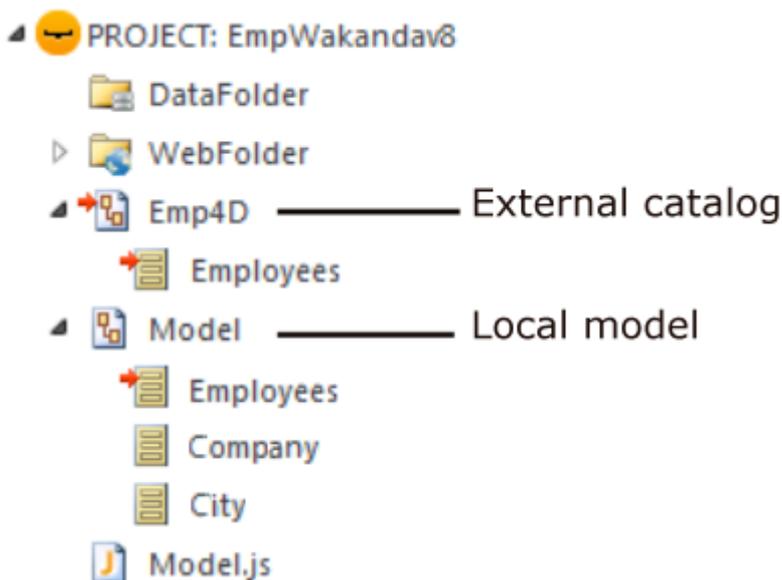
model

Das Objekt *model* gibt das aktuelle "Modell" der Wakanda Applikation an, d.h. die Collection der "Datastore Klassen" (Tabellen) und Methoden. In einer 4D Mobile Architektur kann das Wakanda Modell leer sein. Enthält die Wakanda Applikation bereits Objekte, werden die Klassen und Methoden aus der remote 4D Anwendung mit dem lokalen Modell zusammengelegt, wenn Sie die Methode **mergeOutsideCatalog()** verwenden.

Ist die Verbindung erfolgreich hergestellt, erscheinen die freigegebenen 4D Tabellen in der Liste der Klassen für das Modell auf der Wakanda Seite. In Wakanda Enterprise Studio sehen Sie die remote Tabellen in der Liste der Klassen im lokalen *model*.

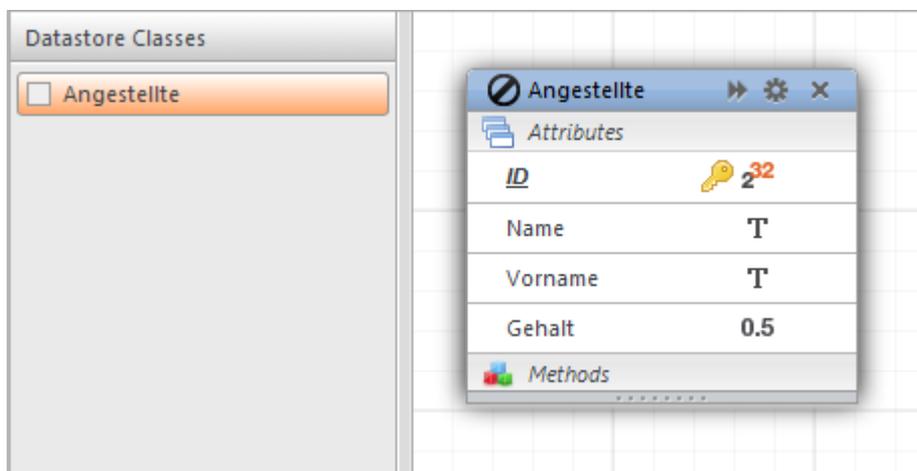
Externe Elemente werden mit roten Pfeilen gekennzeichnet:

Der externe Catalog wird auch in Wakanda Studio angezeigt, über einen spezifischen Catalog mit Namen *localName.waRemoteCatalog*. Er wird auch mit einem roten Pfeil gekennzeichnet:



Hinweis: Dateierweiterungen lassen sich in Wakanda Studio ausblenden.

Doppelklicken Sie auf diese Datei, sehen Sie den externen Catalog im model-Editor des Wakanda Studio:



Beispiel

- Beispiel für direkte Verbindung:

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

- Beispiel für Verbindung über ein Objekt:

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

openRemoteStore() und addRemoteStore()

Die Methoden **openRemoteStore()** und **addRemoteStore()** sind Alternativen zum Einrichten dynamischer Verbindungen zwischen einer Wakanda Applikation und einer 4D Anwendung.

Sie bieten wie **mergeOutsideCatalog()** dynamischen Zugriff auf die Daten von 4D Anwendungen, arbeiten aber in anderer Weise:

- Sie können während der Wakanda Session jederzeit auf ein remote Modell verweisen -- und nicht nur beim Laden der Lösung.
- Tabellen, Attribute und Methoden des externen Modells sind über einen separaten Datastore zugänglich; sie werden nicht mit dem lokalen Modell der Wakanda Applikation zusammengelegt (Zugriff über das Objekt ds).

openRemoteStore() gibt eine gültige Referenz nur im aktuellen JavaScript Kontext zurück, während **addRemoteStore()** die Referenz während der Session beibehält.

Weitere Informationen dazu finden Sie in der Beschreibung zu den Methoden [openRemoteStore\(\)](#) und [addRemoteStore\(\)](#) in der Wakanda Dokumentation.

Integration in aktives model oder eigenes model

Sie können die Verbindung mit dem remote 4D *datastore* auf zwei Wegen herstellen (über den Befehl "Connect to Remote Datastore" von Wakanda Studio oder durch Ausführen einer JavaScript Methode). Dabei müssen Sie jedem Fall angeben, ob die remote Klassen (Tabellen) mit dem aktiven *model* verknüpft oder in ein eigenes *model* gelegt werden sollen.

Folgende Tabelle zeigt die Übersicht:

Für...	Mit aktivem model verknüpfen	eigenes model verwenden
Dialogfenster "Connect to Remote Datastore"	<i>Merge with active Model</i> aktivieren	<i>Merge with active Model</i> deaktivieren
JavaScript Methode	mergeOutsideCatalog()	openRemoteStore() oder addRemoteStore()

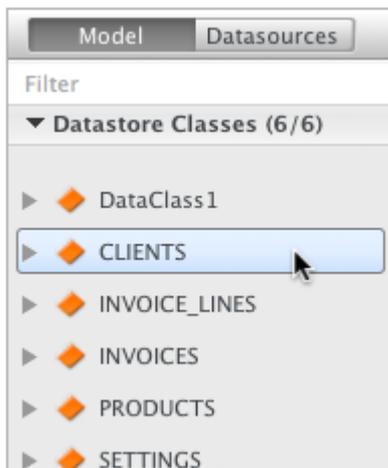
Integration in aktives model

Verknüpfen Sie remote 4D Tabellen mit dem aktiven *model*, werden sie als lokale Klassen in das Standardmodell der Applikation (datastore ist das **ds** Objekt) gesetzt. Der Zugriff ist wie folgt:

- Auf Server Seite greifen Sie über das **ds** Objekt auf 4D Tabellen und Methoden zu (siehe **4D Tabellen und Methoden aufrufen**). Beispiel:

```
var invoiceList = ds.INVOICES.all(); //Standard Zugriff auf die Tabelle INVOICES des Catalog
```

- Auf Client Seite haben Sie die automatischen Features der Wakanda Ajax Framework (WAF) library: remote 4D Tabellen sind über die high-level *Datasource* Objekte verfügbar, oder über das *Dataprovider* API für lower-level Zugriff.
- In Wakanda Enterprise Studio werden die Tabellen der 4D Datenbank über die lokalen Klassen im GUI Designer von Wakanda aufgelistet:



Diese Funktionsweise vereinfacht die Entwicklung von 4D Mobile Applikationen, können aber zu Namenskonflikten zwischen Tabellen führen, insbesondere, wenn die Web Applikation mehrere *Datastores* aufruft. In diesem Fall ist es u.U. besser, die remote Elemente in ein eigenes *model* zu legen.

Eigenes model verwenden

Remote 4D Tabellen, die nicht mit dem aktiven Model verknüpft sind, verwenden ein "dediziertes" model. Dann verwenden remote Klassen einen Namensbereich, der exakt zu der *Datastore* passt, mit der die Anwendung verbunden ist und sind nicht im **ds** Objekt zugänglich. So lassen sich mehrere Tabellen mit demselben Namen in unterschiedlichen Datastores verwenden:

- auf Server Seite greifen Sie auf remote 4D Tabellen und Methoden über einen eigenen Catalog zu. Er hat den Namen, den Sie im Verbindungsparameter **Remote datastore name** (Dialogfenster) oder in *localName* (JavaScript Methode) angegeben haben. Haben Sie z.B. einen Link mit Namen "my4Dstore" erstellt, schreiben Sie im Code der Applikation wie folgt:

```
var invoiceList = my4Dstore.INVOICES.all(); //Zugriff auf die Tabelle INVOICES des Datastore  
my4Dstore
```

Diese Vorgehensweise hat jedoch in der aktuellen Version von Wakanda Enterprise gewisse Einschränkungen:

- Client Applikationen können über die WAF library oder über REST nicht direkt auf remote Klassen zugreifen
- Remote Klassen erscheinen nicht im *GUI Designer* des Wakanda Enterprise Studio.

Im allgemeinen wird empfohlen, für remote *Datastores* den Verknüpfungsmodus (merged mode) zu verwenden, wenn Ihre Client Applikationen direkt auf die Daten der 4D remote Tabellen zugreifen müssen.

Externe Datei ändern

Mit Wakanda Enterprise können Sie bestimmte Merkmale in der lokalen Version des externen Modells verändern. Das können Anpassungen, Optimierung oder Sicherheitsaspekte sein.

Dazu müssen Sie in eine *.js* Datei passenden JavaScript Code setzen. Diese Datei muss denselben lokalen Namen wie der Catalog plus die Endung *.js* haben und im gleichen Ordner wie das Modell liegen. **Beispiel:** Ist der Name des lokalen Catalog *Emp4D.waRemoteModel*, müssen Sie im Ordner des Modells eine Datei mit Namen *Emp4D.js* verwenden.

Hinweise:

- Ab Version 11 legt Wakanda Studio diese Datei standardmäßig an.
- Stellen Sie die Verbindung über eine JavaScript Methode her, können Sie über den Parameter *jsFile* auch einen anderen Namen verwenden.

Wakanda führt diese Datei aus, wenn der externe Catalog initialisiert ist. Sie können mit dieser Datei folgendes ausführen:

- Eigenschaften der Attribute der Datastore Klasse ändern, wie z.B. Ereignisse oder Bereich. Beispiel:

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

- Berechnete Attribute in Datastore Klassen hinzufügen. Beispiel:

```
model.className.calcAtt = new Attribute("calculated", "string"); model.className.calcAtt.onGet =  
function(); model.className.calcAtt.onSet = function();
```

- Alias Attribute in Datastore Klassen hinzufügen. Beispiel:

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

- Lokale Datastore Klassen erstellen, die von Tabellen des externen Catalogs abgeleitet sind, um an Clients gesendete Daten komplett zu steuern. Eine abgeleitete Datastore Klasse kann eine eigene Ansicht einer externen Tabelle zeigen, wobei der generelle Zugriff auf die erweiterte (parent) Datastore Klasse auf dem Wakanda Server erhalten bleibt. Beispiel:

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

- Attribute aus abgeleiteten lokalen Datastore Klassen entfernen; dies dient zur Sicherheit oder Optimierung des Netzwerkverkehrs. Beispiel:

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

Dieses Beispiel erstellt eine abgeleitete Klasse mit Namen "DerivedClass", ausgehend von der Klasse "My4DTable". Sie sendet nur die Attribute, die Sie über das Netzwerk verwenden wollen.

Weitere Informationen über JavaScript Code zum Arbeiten mit Modellen finden Sie im Kapitel [Model API](#) des Wakanda Handbuchs *Server Side API*.

Zugriffsrechte definieren

Sie können für Wakanda Server spezifische Rechte setzen: Überall für das remote Model bzw. für jede Klasse individuell. Weitere Information dazu finden Sie im Abschnitt [Assigning Group Permissions](#) der Wakanda Dokumentation.

4D Tabellen und Methoden aufrufen

4D Tabellen aufrufen

Der Zugriff auf 4D Tabellen, auf die in der Wakanda Applikation Bezug genommen wird, richtet sich danach, wie der externe Catalog integriert ist. Dies wird in Wakanda während der Verbindung zur remote Anwendung definiert (siehe **Integration in aktives model oder eigenes model**):

- Anbindung an aktives model (Standardoption): In diesem Fall werden remote Tabellen exakt wie lokale Klassen über das **ds** Objekt verwendet
- Über eigenes *model*: In diesem Fall sind remote Tabellen Eigenschaften des Objekts "dediziertes" model.

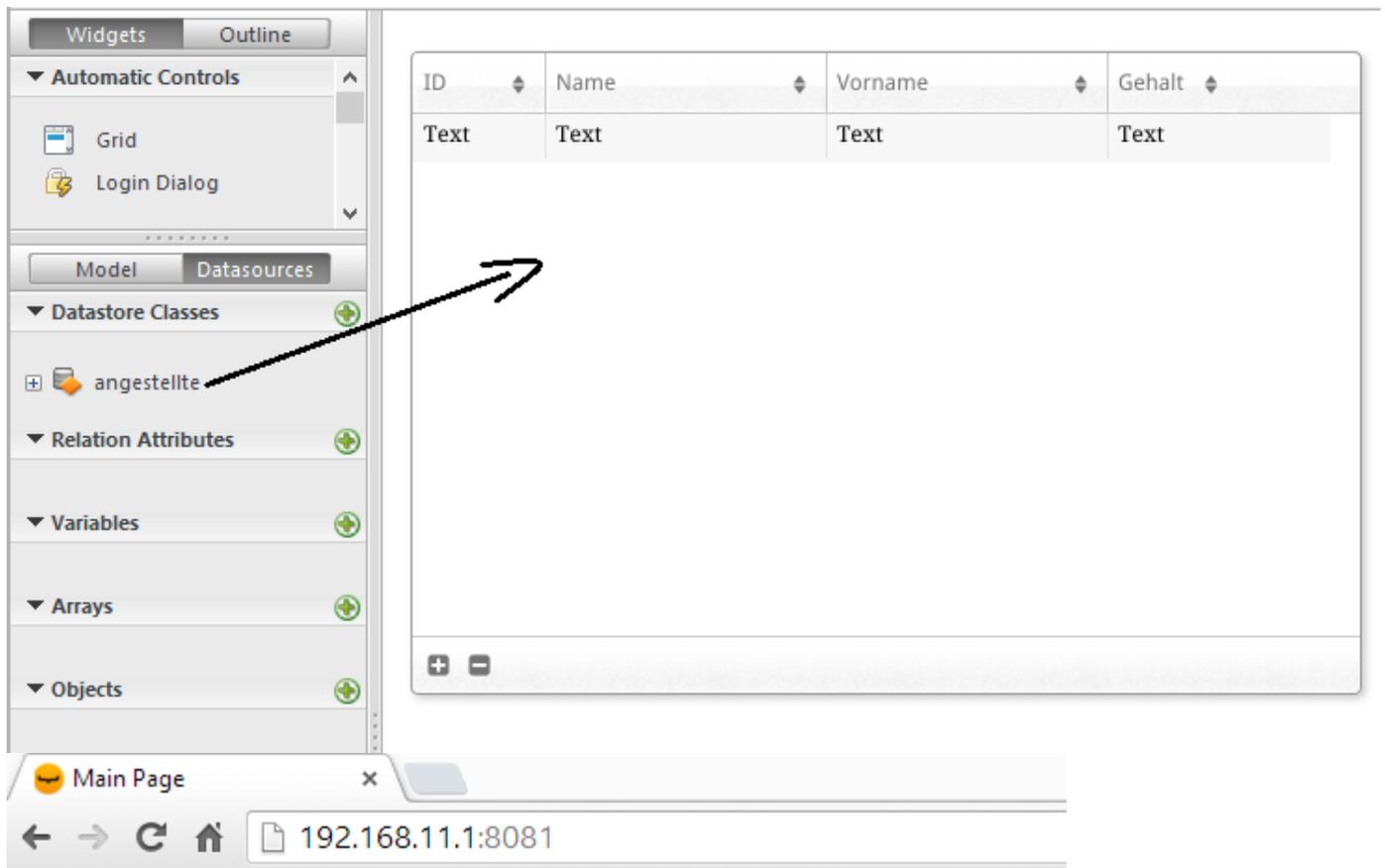
Tabellen mit Anbindung an aktives model

Bei Anbindung an das aktive *model* lassen sich 4D Tabellen, auf die in der Wakanda Applikation Bezug genommen wird, im server-seitigen JavaScript Code direkt als Eigenschaften des **ds** Objekts verwenden, genauso wie lokale *Datastore Klassen*.

Hinweis: Das Objekt **ds** enthält die aktuelle *Datastore* der Wakanda Applikation. Beispiel für eine Suche in den Datensätzen der Tabelle [Angestellte]:

```
var emp = ds.Angestellte.query("Alter > :1",30);      // die Datensätze in der Tabelle Angestellte finden
// in denen in der Variablen <em>emp</em> Alter größer als 30 ist
```

Auf Client Seite können Sie auch die automatischen Abläufe von Datasources nutzen, die auf Datastore Klassen basieren und mit Widgets verknüpft sind. Weisen Sie z.B. die Datasource 'angestellte' einem Widget vom Typ 'Grid' zu, lässt sich die Liste der Angestellten automatisch anzeigen:



ID	Name	Vorname	Gehalt
1	Schneider	Hans	25000
2	Müller	Anton	35000
3	Weber	Maria	41000

3 item(s)

Ist die Tabelle einer Datasource zugewiesen, können Sie auch darüber auf die Daten zugreifen. Sie sortieren z.B. eine Collection Datensätze der Datasource 'angestellte'.

```
sources.angestellte.orderBy("Alter"); //sortiert die Collection Angestellte nach Alter
```

Weitere Informationen dazu finden Sie in der [Wakanda Dokumentation](#).

Tabellen in eigenem model

Die referenzierten 4D Tabellen werden im Server-seitigen JavaScript Code als Eigenschaften des Catalog verwendet, in den sie beim Erstellen des Links gelegt wurden. *Catalog* hat den Namen, der im Verbindungsparameter **Remote datastore**

name (Wakanda Studio Verbindungsdialog) oder in *localName* (JavaScript Methode) definiert wurde.

Haben Sie z.B. einen Link mit Namen "my4Dstore" erstellt und wollen eine Suche in den Datensätzen der Tabelle [Angestellte] durchführen, schreiben Sie folgenden Code:

```
var emp2 = my4Dstore.Angestellte.query("Alter > :1", 30);    // suche in den Datensätzen der Tabelle
Angestellte    // im Link mit Namen "my4Dstore"
```

Hinweis zur Implementation: Auf Client Seite ist mit der aktuellen Version von 4D Mobile für ein eigenes model derzeit kein Zugriff auf remote Klassen möglich.

4D Methoden aufrufen

Bereich und Objekte

4D Methoden, auf die in der Wakanda Applikation verwiesen wird, lassen sich direkt im JavaScript Code als Eigenschaften der Objekte **Datastore Klasse**, **Entity Collection** oder **Entity** verwenden, jeweils nach dem Bereich, der auf 4D Seite definiert wurde (siehe **Tabelle und Bereich der Projektmethoden**). Hier die Entsprechungen zwischen 4D und Wakanda:

4D Bereich	Wakanda Objekt
Tabelle	Datastore Klasse
Aktuelle Auswahl	Entity Collection
Aktueller Datensatz	Entity

Hinweis: Über *Datasources* lassen sich 4D Methoden auch auf der Client Seite aufrufen (siehe unten); in diesem Fall sind alle Methoden verfügbar und die *Datasource* wendet sie je nach Kontext automatisch auf die aktuelle Collection bzw. Entity an.

Führen Sie z.B. eine Suche über die Methode query aus (siehe voriger Abschnitt), gibt Wakanda eine Entity Collection zurück. In dieser Collection können Sie jede 4D Projektmethode ausführen, für die als Bereich "aktuelle Auswahl" definiert wurde.

Server oder Client API

Es gibt drei Wege, um 4D Methoden über JavaScript Code aufzurufen:

- Über JavaScript Code, der über [SSJS Datastore API](#). auf dem Server (SSJS) ausgeführt wird. Hier werden 4D Methoden als Eigenschaften der Objekte **Datastore Klasse**, **Entity Collection** oder **Entity objects** aufgerufen, wie oben beschrieben.

Beispiele:

```
var vTot = ds.Emp.raiseSalary(param))    //raiseSalary ist eine Eigenschaft der datastore Klasse
//der Catalog wird mit an das aktive model angebunden var vTot2 =
my4DStore.Company.first().capital(param))    //capital ist Eigenschaft der entity, da first() eine
entity zurückgibt    //das eigene model my4DStore wird verwendet
```

- Über JavaScript Code, der über das Wakanda Ajax Framework (WAF) auf dem Client ausgeführt wird, wie z.B. ein Browser. Hier gibt es zwei Möglichkeiten:
Hinweis zur Implementierung: In der aktuellen Version von Wakanda Enterprise ist Client Zugriff auf 4D Datenbankmethoden nur möglich, wenn die remote Anwendung angemeldet und an das aktive *model* angebunden ist.

•

- [WAF Datasource API](#): Dieses high-level API bietet viele automatische Funktionen zum Verwalten von Daten. Damit werden 4D Methoden als **Eigenschaften von Datasources, die Datastore Klassen zugeordnet sind**, aufgerufen und je nach Kontext automatisch auf die Datastore Klasse, die aktuelle Entity Collection oder die aktuelle Entity angewandt. Sie können Rückgabewerte der Methoden oder Fehler über asynchrone Syntax verwalten (das ist für Code, der auf dem Client ausgeführt wird, erforderlich). Beispiel:

```
sources.employee.raiseSalary(param, {onSuccess: function(event) { ... //auszuführender Code, wenn die Methode beendet ist} })))
```

Eine Callback Funktion ist hier nicht zwingend erforderlich, da Datasource Objekte automatische Funktionen haben, die z.B. dafür sorgen, dass die Daten in der aktuellen Collection nach einer Suche upgedatet werden.

- [WAF Dataprovider API](#): Über dieses low-level Client API können Sie direkt mit Objekten arbeiten. 4D Methoden werden, wie bei SSJS Datastore API, als Eigenschaften der Objekte **Datastore Klasse, Entity Collection** oder **Entity objects** aufgerufen. Sie müssen jedoch Rückgabewerte der Methode oder Fehler über asynchrone Syntax verwalten (das ist für Code erforderlich, der auf dem Client ausgeführt wird). Beispiel:

```
ds.Employee.raiseSalary(param, // Syntax ähnlich wie bei einem SSJS Aufruf {onSuccess: function(event) // da es client-seitiger Code ist, müssen Sie // die Callback Methode des asynchronen Aufrufs selbst verwalten { ... //auszuführender Code, wenn die Methode beendet ist} })))
```

Welchen Ort (Server oder Client) und welches API Sie wählen, richtet sich nach den Anforderungen der Applikation. Weitere Informationen dazu finden Sie in der Wakanda Dokumentation.

Parameter

Sie können, wie bei Standardmethoden, während dem Aufruf Parameter übergeben, die dann der Reihe nach in den Parametern \$1, \$2, etc. empfangen werden. In ähnlicher Weise gibt die Methode ein Ergebnis in der \$0 Variable zurück.

Hierzu ein Beispiel: Angestellte mit einem Gehalt unter 1500 sollen 5% mehr bekommen.

- Auf 4D Seite ist die Methode **IncreaseSalary** über 4D Mobile veröffentlicht und für "Aktuelle Auswahl" gültig. Ihr Code lautet wie folgt:

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

- Auf Wakanda Seite führen Sie auf dem Server folgenden Code aus:

```

var emp = ds.Employees.query("salary < :1",1500);           // emp enthält die Collection Employees mit
einem Gehalt unter 1500   emp.IncreaseSalary(1.05);       //enthält die Collection Employees mit
einem Gehalt unter 1500   //Sie könnten auch schreiben:   //"ds.Employees.query("salary
< :1",1500).IncreaseSalary(1.05);

```

Mit dem Befehl **MOBILE Return selection** können Sie eine 4D Auswahl auch direkt als eine Wakanda Collection zurückgeben. Hierzu ein Beispiel:

```

//Projektmethode FindCountries
//FindCountries( string ) -> object

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

```

Update von 4D Kontext

Beim Aufrufen einer 4D Methode über den Wakanda Link geschieht folgendes:

- Bei Anwendung auf eine Auswahl (entity collection) wird das die aktuelle Auswahl. 4D wird auf den ersten Datensatz dieser Auswahl gesetzt, ohne ihn zu laden oder die Links zu aktivieren. Ist die Auswahl leer, gibt die Funktion **Selected record number** 0 anstatt 1 zurück.
- Bei Anwendung auf einen Datensatz (entity), wird dies der aktuelle Datensatz. Die aktuelle Auswahl wird auf diesen einen Datensatz reduziert und die Funktion **Selected record number** gibt 1 zurück.
Hinweis: Zur Optimierung und um unnötiges Sperren zu vermeiden, wird der Datensatz im Nur-Lesen Modus geladen. Ist die Tabelle dagegen im Lese-/Schreibmodus, können Sie über den Befehl **LOAD RECORD** erzwingen, dass der Datensatz bei Bedarf im Lese-/Schreibmodus geladen wird.
- Bei Anwendung auf eine Tabelle (*datastore Klasse*) sind weder die aktuelle Auswahl noch der aktuelle Datensatz betroffen.

Beachten Sie, dass der 4D Kontext nach Ausführen einer Methode über 4D Mobile zurückgesetzt wird:

- Auswahlen werden auf 0 reduziert
- Datensätze werden vom Stapel genommen und entladen
- lokale Auswahlen und Mengen für den Prozess werden aufgehoben
- Alle während der Ausführung der Methode geöffneten Transaktionen werden annulliert
- Die Konfiguration automatischer Verknüpfungen nach Feldern, Suchzielen oder Suchen auf dem Server werden zurückgesetzt
- Druckaufträge werden annulliert
- Fenster werden geschlossen
- alle SQL, PHP oder HTTP Verbindungen werden geschlossen

Fehler für Bereich

Achten Sie darauf, dass der Bereich der 4D Methode zum Typ des aufgerufenen Wakanda Objekts passt. Sonst gibt Wakanda den Fehler "*TypeError: 'undefined' is not a function*" zurück.

Nehmen wir beispielsweise die 4D Methode "getcursel" mit folgendem Code:

```
$0:=Records in selection([Table_1])
```

Auf Wakanda Seite gibt es die Methode *run*:

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

Die Methode **query()** gibt eine Collection zurück. Ist für die Methode *getcursel* als Bereich (Gültig) "aktueller Datensatz" gewählt, gibt Wakanda folgenden Fehler zurück:

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

Verknüpfungen verwenden

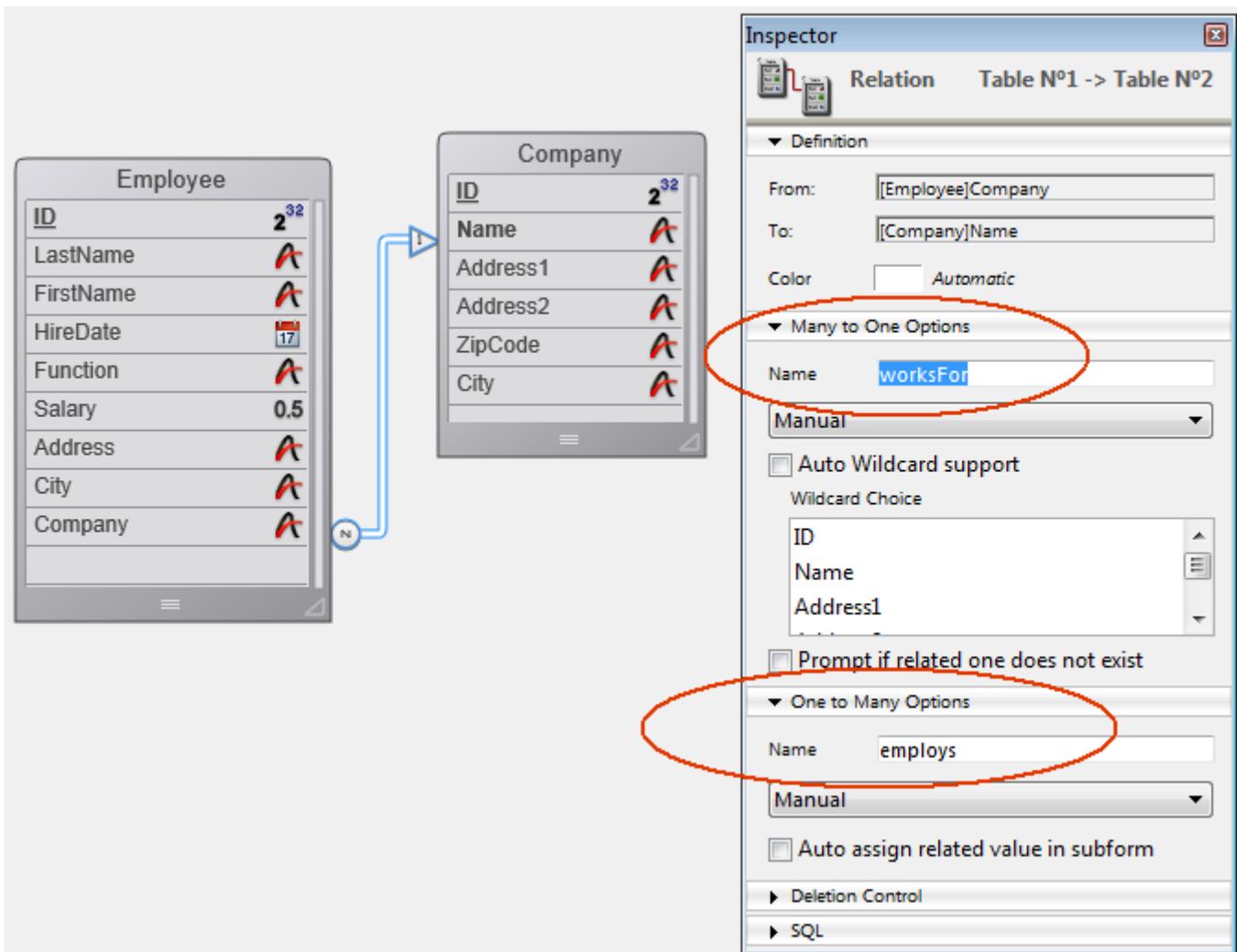
Verknüpfungen, die zwischen 4D Tabellen bestehen, werden im Kontext eines 4D Mobile Links transparent genutzt. Sie werden jedoch in Wakanda auf model-Ebene anders dargestellt. Im model Editor sind Relationen mit spezifischen Attributen verbunden, genannt Relation Attribute. Über diese Attribute lassen sich verknüpfte Daten direkt anzeigen oder Suchen durchführen. Weitere Informationen dazu finden Sie im Abschnitt "[Attributes](#)" der Wakanda Dokumentation.

Für jede Verknüpfung auf 4D Seite werden in der model Darstellung auf Wakanda Seite zwei Relation Attribute hinzugefügt:

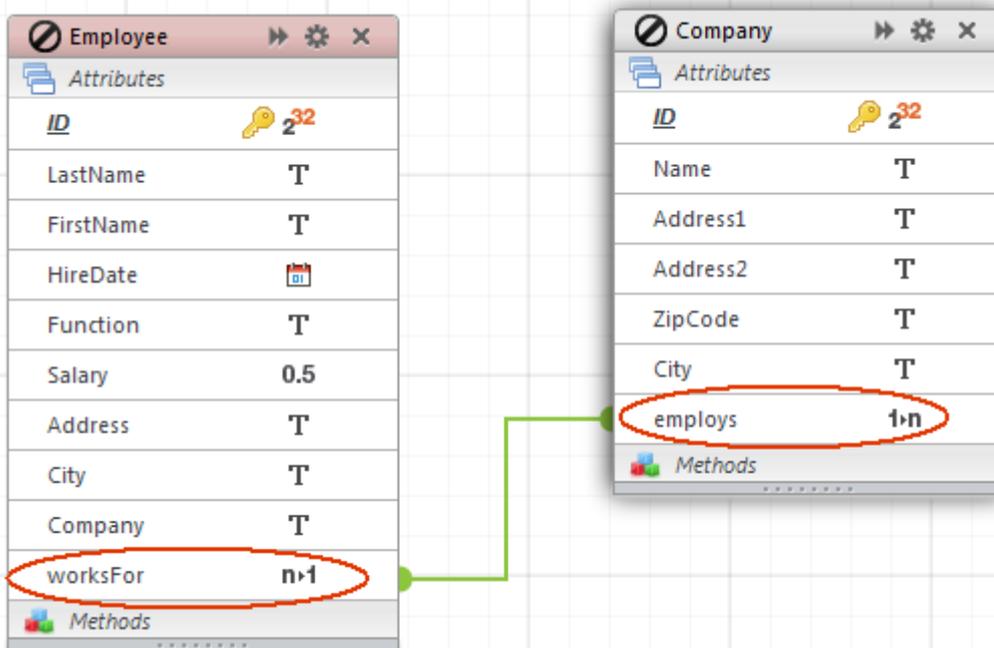
- ein n->1 Attribut in der Quelltable (Klasse) der Relation
- ein 1->n Attribut in der Zieltabelle (Klasse) der Relation

Beide Attribute erhalten den Namen der Verknüpfung, der auf 4D Seite im Inspektorfenster jeweils für die Viele-zu-Eine und die Eine-zu-Viele Verknüpfung definiert wurde.

Nehmen wir als Beispiel eine Struktur "Employees/Company" und erstellen eine Verknüpfung von der Tabelle [Employee] zur Tabelle [Company]. Über einen Namen können Sie diese Verknüpfungen genauer beschreiben. Wir nennen die Viele-zu-Eine Verknüpfung z.B. "worksFor", die Eine-zu-Viele Verknüpfung "employs":

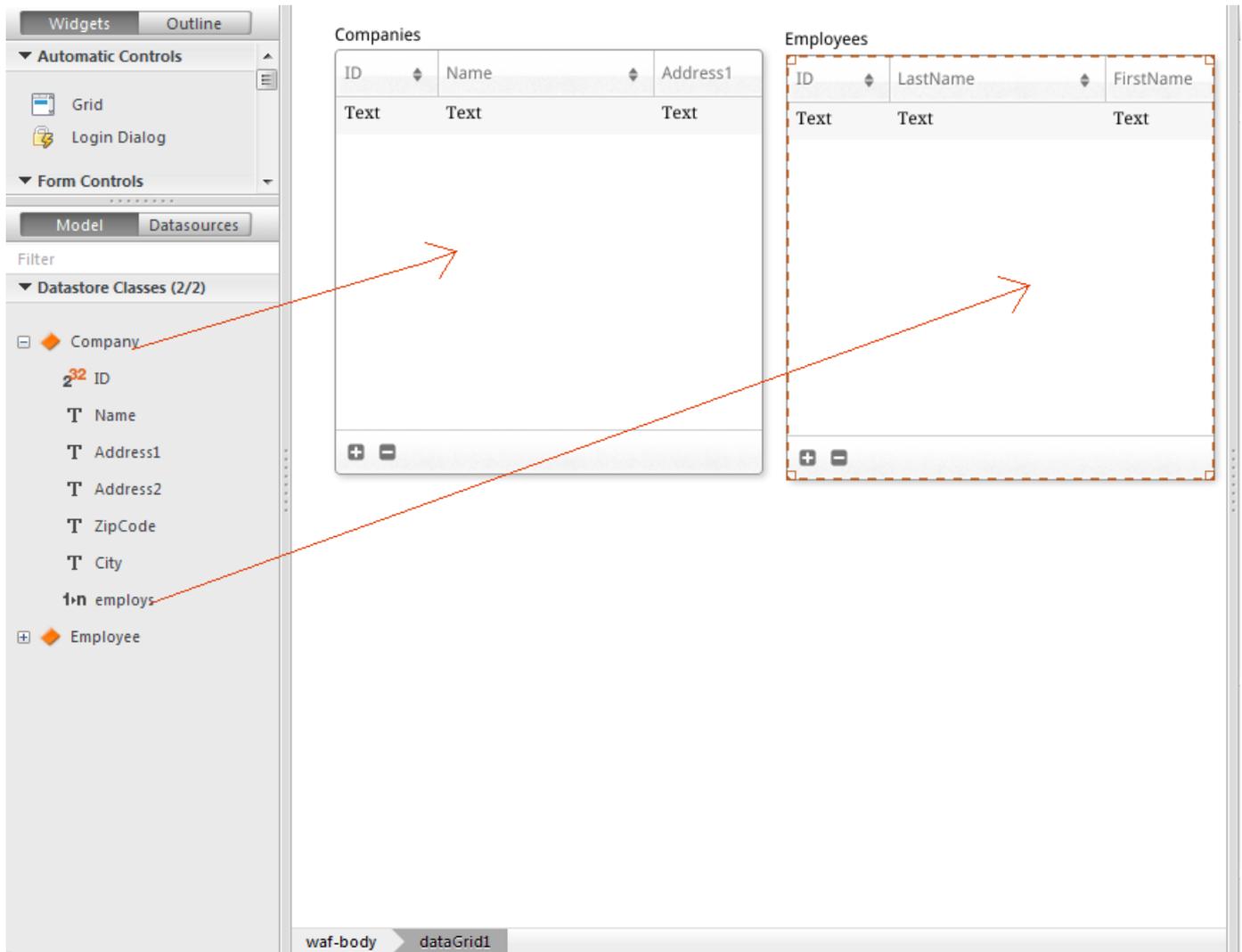


Diese Verknüpfungen werden auf Wakanda Seite über einen Link via Connector automatisch in zwei zusätzliche Relation Attribute umgesetzt. Das sieht im model Editor folgendermaßen aus:



Sie können den Verknüpfungen - und auch den dazugehörigen Relation Attributen - beliebige Namen geben, die sich nach der Logik Ihrer Anwendung richten. Auf diese Weise lassen sich diese Attribute auf der Wakanda Seite ganz einfach nutzen, um mit den verknüpften Daten zu arbeiten. Sie können insbesondere Widgets für Datasources erstellen, die auf Relation Attributen basieren. Diese

Widgets werden dann über Benutzeraktionen verwaltet und automatisch aktualisiert. Sie können z.B. eine Seite mit zwei Grids erstellen: Eins für die Liste der Firmen und ein anderes für die Angestellten innerhalb einer ausgewählten Firma.



Die entsprechenden Datasources werden automatisch angelegt und beide Grids werden während der Ausführung automatisch synchronisiert:

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)

4D Mobile Sessions verwalten

Überblick

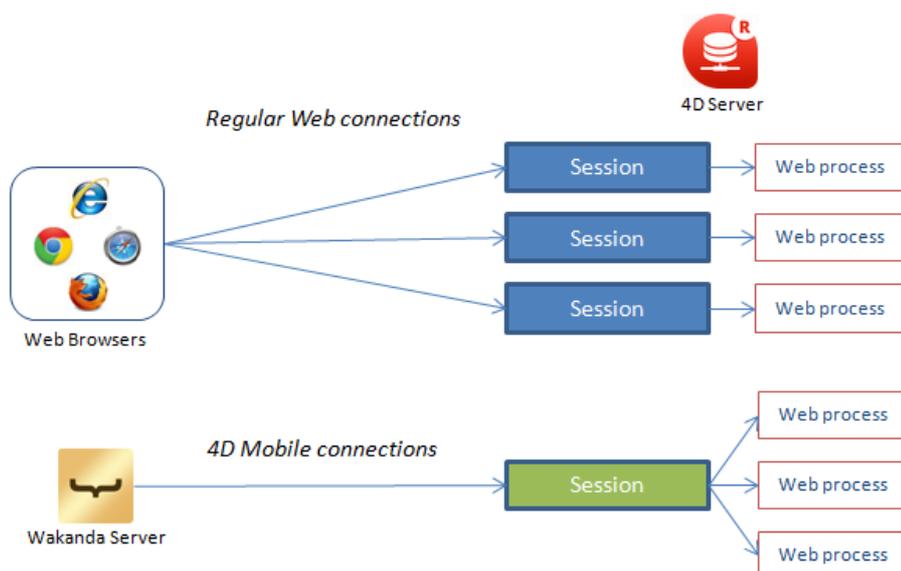
Ab 4D v15 R4 können Sie per Programmierung auf die ID einer 4D Mobile Session auf der 4D Server Seite zugreifen. Über dieses Feature können Entwickler Informationen zu Sessions lokal erhalten und setzen (siehe Beispiel unten).

4D Mobile Sessions werden über reguläre 4D Befehle für Web Sessions verwaltet. Einige 4D Web Befehle, die Funktion **WEB Get session process count** und die **Datenbankmethode On Web Close Process** unterstützen 4D Mobile Sessions.

4D Mobile Session versus Web Session

4D Mobile Sessions und Web Sessions sind zwei unterschiedliche Sitzungsarten. Obwohl sie einige Konzepte und Befehle gemeinsam nutzen, haben sie nicht die gleichen Eigenschaften. Der Hauptunterschied ist die Relation zwischen einer Session, einem Prozess und einem Prozesskontext:

- Eine Web Session ist mit einem einzelnen Web Prozess verknüpft; über das Feature zur automatischen Session-Verwaltung lässt sich der Prozesskontext (Variableninstanzen, Auswahlen, etc.) der Session wiederverwenden.
- Eine 4D Mobile Session kann mit mehreren Web Prozessen verknüpft sein; jeder Prozesskontext wird nach Ausführen der Prozessmethode automatisch zurückgesetzt.



Für die gemeinsame Nutzung von Informationen zwischen 4D Mobile Web Prozessen innerhalb einer Session sind spezifische Implementierungen auf 4D Server erforderlich.

Unterstützte Befehle ohne Änderung

Folgende Befehle zum Verwalten von Web Sessions unterstützen 4D Mobile Sessions.

WEB CLOSE SESSION (SessionID)

Der Befehl **WEB CLOSE SESSION** schließt die 4D Mobile Session mit der ID *SessionID*. Da eine 4D Mobile Session mehrere Prozesse verwalten kann, fragt dieser Befehl alle zugeordneten Web Prozesse ab, um ihre Ausführung zu beenden.

WEB Get Current Session ID -> SessionID

Die Funktion **WEB Get Current Session ID** gibt die UUID zurück, die der aktuellen 4D Mobile Session zugewiesen ist.

WEB GET SESSION EXPIRATION (SessionID; Ablaufdatum; Ablaufzeit)

Der Befehl **WEB GET SESSION EXPIRATION** gibt Informationen zum Ablauf des Cookie für eine 4D Mobile Session zurück.

Für alle Prozesse innerhalb einer 4D Mobile Session wird dasselbe Cookie verwendet.

Funktion WEB Get session process count

Über die Funktion **WEB Get session process count** können Sie die Anzahl der vorhandenen Prozesse für eine bestimmte Sitzung erhalten.

- Für reguläre Web Sessions gibt die Funktion immer 1 zurück (eine Web Session = ein Prozess)
- Für 4D Mobile Sessions gibt die Funktion alle dazugehörigen Web Prozesse zurück. Das ist z.B. hilfreich, um eine Schleife auf alle Prozesse einer 4D Mobile Session auszuführen

Datenbankmethode On Web Close Process (bisher On Web Session Suspend)

4D ruft die **Datenbankmethode On Web Close Process** jedes Mal auf, wenn ein Web Prozess gerade seine Ausführung beendet. Sie unterstützt vollständig Prozesse der 4D Mobile Session: In diesem Kontext wird sie für jeden geschlossenen Web Prozess aufgerufen, so dass sie alle vom jeweiligen Prozess der 4D Mobile Session erzeugten Daten sichern können, wie z.B. eine Variable, eine Auswahl, etc.

Hinweis: Für reguläre Web Sessions wird die **Datenbankmethode On Web Close Process** jedes Mal in der Web Session aufgerufen, z.B. wenn der einmalige Prozess in der Web Session geschlossen wird.

Beispiel

Wollen Sie die Information zwischen mehreren Prozessen einer einzelnen 4D Mobile Session gemeinsam nutzen oder wiederverwenden, können Sie über die UUID der 4D Mobile Session die Daten dieser Session identifizieren. Sie wollen z.B. nach einer Suche von Datensätzen auf dem 4D Server eine temporäre Auswahl beibehalten, so dass nachfolgende Anfragen über REST in der gleichen Session direkt auf diese Auswahl zugreifen können. Dafür schreiben Sie nach der Suchanweisung folgenden Code:

//Eine Interprozess Auswahl mit Session UUID erstellen

COPY NAMED SELECTION([Emp];"<>EmpSel"+**WEB Get Current Session ID**)

//Sie können diese Auswahl aus der gleichen Session später wiederverwenden

USE NAMED SELECTION([Emp];"<>EmpSel"+**WEB Get Current Session ID**)

Sicherheit der 4D Mobile Applikation

Werden über 4D Mobile veröffentlichte Daten aus Tabellen einer 4D Anwendung in den Wakanda Catalog übernommen, müssen Sie bestimmte "sensible" Ressourcen vor Zugriff schützen.

Web Applikationen können im Gegensatz zu 4D Anwendungen nicht das Interface zum Steuern freigegebener Daten verwenden. So sind z.B. auch Felder, die nicht im Formular erscheinen, für Benutzer zugänglich. Über HTTP Anfragen und JavaScript können böswillige Benutzer faktisch jede Informationen von einem ungenügend geschützten Web Server abrufen.

Dieser Abschnitt kann nicht alle Sicherheitsmaßnahmen für 4D Mobile Applikationen auflisten. Er gibt Ihnen aber einige Hinweise, wie Sie sensible Daten in gewisser Weise absichern können.

- **4D Mobile Zugriffe auf die 4D Anwendung schützen:** Sie können die 4D Mobile Verbindungsanfragen (via REST) steuern. Dafür verwenden Sie entweder:
 - 4D Kennwörter (siehe Abschnitt [Automatische Steuerung über 4D Kennwörter](#)),
 - die [Datenbankmethode On 4D Mobile Authentication](#).
- **4D Mobile Veröffentlichung auf 4D Seite steuern:** Jede Tabelle, Methode und jedes Attribut lässt sich einzeln über 4D Mobile veröffentlichen. Geben Sie nur die unbedingt notwendigen Daten und Methoden frei. Sie müssen beispielsweise keine Felder veröffentlichen, die nicht verwendet werden.
- **Sensible Daten schützen:** Über die Sicherheitssysteme von Wakanda können Sie Inhalte steuern, die über Browser zugänglich sind. Es gibt verschiedene Wege, drei davon erläutern wir hier näher:
 - **Reichweite anpassen:** Das gilt für Attribute und Methoden aus der 4D Anwendung in Wakanda auf model Ebene (siehe Eigenschaft **scope** für [Attributes](#) oder für [Methods](#) in der Wakanda Dokumentation). Sie können dafür die Option **Public on Server** festlegen. Dann sind sie für Code, der auf dem Server läuft, frei zugänglich, jedoch nicht auf Web Clients. Diese Option können Sie in der zusätzlichen .js Datei festlegen, die über die Methode **mergeOutsideCatalog()** ausgeführt wird. Weitere Informationen dazu finden Sie auf der Seite [Model API](#) im Wakanda Handbuch *Server-Side API*.
 - **Berechnete Attribute verwenden:** Berechnete Attribute arbeiten wie Standard Attribute, ihre Werte werden jedoch über spezifische Funktionen (onGet, onSet...) zurückgegeben, die bei Zugreifen auf die Felder ausgeführt werden. Von daher genügt es, wenn nur die berechneten Attribute freigegeben werden und nicht die Felder der 4D Anwendung. Zugriffe auf die 4D Felder erfolgen vom Wakanda Server aus in sicherer Form.

Sie können berechnete Felder in einer zusätzlichen .js Datei hinzufügen, die über die Methode **mergeOutsideCatalog()** ausgeführt wird. Weitere Informationen dazu finden Sie auf der Seite [Attributes](#) im Wakanda Handbuch *Server-Side API*.

- **Erweiterte Datastore Klassen und [restricting queries](#) kombinieren:** Dieses leistungsstarke Konzept unterstützt nicht nur veröffentlichte Attribute, sondern auch die anzeigbaren Daten. Eine *Datastore Klasse* erweitern heißt, eine Kopie erstellen (die abgeleitete Klasse), die Sie verändern können, also berechnete Attribute hinzufügen oder vorhandene Attribute löschen. Sie können auch eine *eingeschränkte Suche* zuordnen: In diesem Fall löst jeder Zugriff auf die Daten der abgeleiteten Klasse diese Suche automatisch aus, und gibt nur die Datensätze zurück, die zum Suchkriterium passen. So können Sie die Daten mit dem Benutzer verbinden, der an den Wakanda Server angemeldet ist. Bei einer Business-Lösung gibt die Suche z.B. alle Kunden zurück, die zum aktuellen Vertriebsmitarbeiter gehören. Natürlich ist über Web Clients nur der Zugriff auf die abgeleitete Klasse möglich.

Sie können erweiterte *Datastore Klassen* und *restricting queries* in der .js Konfigurationsdatei des externen Modells anlegen (siehe **Externe Datei ändern**). Weitere Informationen dazu finden Sie auf der Seite [Programming Restricting Queries](#) der Wakanda Dokumentation.

Hinweis: [restricting queries](#) in 4D Mobile werden ab folgenden Versionen unterstützt:

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