

servoy Next Generation Platform

der Schnellstart

Servoy 6.0 – v. 1.0 – 10. Februar 2012

Copyright © 1997 – 2012 Servoy BV Alle Rechte vorbehalten

Servoy und das Logo von Servoy sind Warenzeichen von Servoy BV und Servoy Inc. Alle anderen Warenzeichen sind Eigentum ihrer jeweiligen Eigentümer.

Dieses Dokument ist eine Übersetzung des Quick Start Guides auf <http://wiki.servoy.com> und daher nicht immer auf dem letzten Stand. Den aktuellen Stand finden Sie unter <http://wiki.servoy.com/display/public/DOCS/Quick-Start+Guide>

Servoy USA	Internationale Niederlassungen
Servoy Inc. 299 West Hillcrest Drive Suite 115 Thousand Oaks, CA 91360 Telefon: 805 624 4959 Fax: 805 624 4958	Servoy BV De Brand 26 3823 LJ Amersfoort Niederlande Telefon: +31-33-4559877 Fax: +31-84-2242422 E-Mail: support@servoy.com Web: http://www.servoy.com

Inhalt

Einführung in die Konzepte von Servoy	3
Herunterladen und Installieren von Servoy Developer	9
Einführung in die Entwicklungsumgebung von Servoy	10
Erstellen Sie eine Solution	13
Datenbank-Verbindungen verwalten	15
Formulare erstellen	18
Clients starten und debuggen	21
Formulare innerhalb eines Formulars platzieren	22
Implementieren der Geschäftslogik	25
Gestaltung mit CSS	30
Eine Anwendung bereitstellen	35
Eine Solution aus Servoy Developer exportieren	36
Installieren und Ausführen des Servoy Application Server	38
Eine Solution in den Application Server importieren	40
Eine Solution im Client ausführen	42

Einführung in die Konzepte von Servoy

Über Servoy

Ein Alleinstellungsmerkmal von Servoy ist, dass es sich sowohl um ein Rapid Application Development (RAD)-Tool als auch um eine Plattform für die Bereitstellung von Produktionsanwendungen handelt. Es ist unter der Verwendung von Standardkomponenten und -protokollen vollständig in Java geschrieben und bietet so eine Skalierbarkeit bis hin zur Bereitstellung für eine unternehmensweite Verwendung sowie eine unvergleichliche Produktivität der Entwickler.

Unterstützung unterschiedlicher Plattformen

Entsprechend der Versprechen von Java können in Servoy entwickelte Anwendungen auf allen modernen Betriebssystemen verwendet werden. Dazu gehören Windows, Mac, Linux, Solaris und andere, ohne dass dafür der Code neu kompiliert werden muss.

Unterstützung unterschiedlicher Browser

Außerdem können in Servoy geschriebene Anwendungen in jedem modernen Browser ausgeführt werden. Dazu gehören der Internet Explorer, Firefox, Safari, Opera, Chrome usw. Es ist nicht erforderlich, eine Anwendung neu zu schreiben oder anzupassen, um mit allen Browsern die gleiche Darstellung zu gewährleisten. Außerdem werden in keinem Fall proprietäre Protokolle oder Add-Ins für die Browser benötigt.

Eine einheitliche Codebasis

Servoy bietet zwei Clientoptionen an und jede in Servoy entwickelte Anwendung kann mit jeder dieser beiden Optionen ausgeliefert werden, ohne dass sie ganz oder teilweise neu geschrieben werden muss.

Der Smart Client bietet ein Client-/Server-Setup, bei dem die Anwendungen das native Look-and-Feel des Betriebssystems des Clients annehmen.

Der Webclient bietet eine dynamische Browsererfahrung an, die durch AJAX, CSS und HTML angereichert wird. Die Entwickler müssen nie ihr eigenes HTML, clientseitiges JavaScript oder AJAX schreiben, auch wenn dies möglich ist, wenn die Entwickler dies wünschen.

Unterstützung verschiedener Datenbanken

Servoy ist KEINE Datenbank. Es gibt viele gute Datenbanken auf dem Markt und Servoy kann sich mit jedem modernen RDBMS verbinden, auch mit einigen, die nicht so modern sind.

Mehrschichtige Architektur

Servoy ist ein mächtiger Application Server, der Verbindungen zwischen den Clients und den Backend-Ressourcen verwaltet. Diese Architektur hat erhebliche Vorteile gegenüber der zweischichtigen Architektur, die die meisten 4GL-Werkzeuge anbieten, bei denen die Clients sich direkt mit den Backend-Datenbanken verbinden. Ein Application Server hat gegenüber einer zweischichtigen Anwendung die folgenden Vorteile:

- Die Belastung der Backend-Datenbanken wird erheblich gemindert.
- Die Performance wird verbessert.
- Die Skalierbarkeit wird verbessert.
- Die Sicherheit wird erhöht.
- Die Möglichkeit, über ein WAN (Wide Area Network) zu arbeiten.
- Data Broadcasting: Die Möglichkeit aller Clients, die Änderungen an den Daten in Echtzeit zu sehen, ein Feature, das Servoy einzigartig macht.

Keine Installation

Servoy-Anwendungen müssen nie auf der Client-Maschine installiert werden.

Smart Client-Anwendungen werden mit Hilfe der Java WebStart-Technologie automatisch heruntergeladen und installiert. Die Entwickler können neue Änderungen auf einfache Weise an die Anwender ausrollen, indem sie mit einem einzelnen Klick ein neues Build auf den Anwendungs-Server exportieren. Der Anwendungs-Server muss nicht neu gestartet werden und auch die verbundenen Clients benötigen keinen Neustart. Mehr noch, der Anwendungs-Server unterstützt die Versionsverwaltung für ein Roll-Back und Roll-Forward.

Kurz gesagt: keine Installation, keine Verwaltung.

Konzepte für die Programmierung

Solutions

Eine Solution ist eine einzelne in Servoy Developer erstellte Anwendung, die Formulare, Methoden, Variablen und andere Elemente enthält.

Solutions können für sich alleine ausgeführt werden, sie können aber auch ein Modul darstellen. Ein Modul ist eine Solution, die Teil einer anderen Solution ist. Sie können eine Solution explizit als Modul markieren, aber dies ist nicht zwingend erforderlich, um eine Solution als Modul zu verwenden.

Formulare

Ein Formular (Form) ist ein Servoy-Objekt, das für das Ändern, Ansehen, Erstellen, Finden und Löschen von Daten verwendet wird. Formulare können Felder, Beschriftungen, Grafiken, Tabs, Linien, Schaltflächen, Portale und freie Formen beinhalten – jedes dieser Objekte verfügt zur Designzeit über seine eigenen Befehle, Ereignisse und Eigenschaften.

In der Regel basieren Formulare auf einer einzelnen Tabelle in der Datenbank, was bedeutet, dass die Daten dieser Tabelle an das Formular gebunden sind.

Außerdem können Formulare Methoden und Variablen enthalten, die an dieses Formular gebunden sind, wodurch ein Formular in Servoy eine Bereichseinheit wird. Anders ausgedrückt können Formulare nur Methoden und Variablen enthalten, die sich im Wirkungsbereich des Formulars befinden. Vererbte Formulare können außerdem Methoden und Variablen enthalten, die vom übergeordneten Formular vererbt wurden.

Formularelemente

Ein Formularelement ist ein Servoy-Element, das für einen bestimmten Zweck auf einem Formular platziert werden kann. Zu den Formularelementen gehören:

- Label – Text, der auf einem Formular angezeigt werden kann. In der Regel handelt es sich dabei zur Laufzeit um statischen Text, der durch den Endanwender weder geändert noch kopiert werden kann.
- Field – Ein Element, das mit einem Datenprovider verbunden ist. Ein Feld kann durch den Endanwender zur Laufzeit geändert werden und es gibt viele Anzeigetypen (z.B. Text, Auswahlfelder).
- Button – Ein spezieller Typ der Labels, der vorprogrammiert ist, so dass er auf einem Formular wie ein anklickbares Objekt aussieht. In der Regel ist ein Button an eine Methode gebunden.
- Linien und Formen – verschiedene Zeichenobjekte, die aus Gründen der Ästhetik oder des Designs der Benutzeroberfläche auf ein Formular gezogen werden können.
- Portal – ein Objekt, das verbundene Daten in einer One-to-Many-Relation in Feldern anzeigt.
- Tabpanel – ein Objekt, das die Anzeige mehrerer Formulare auf einem Formular ermöglicht. Die zusätzlichen Formulare können mittels einer Relation (die auch miteinander verbundene Daten anzeigt) oder als nicht durch eine Relation verbundene Formulare eingebunden werden.
- Beans – Sie können einem Formular eine Java Bean hinzufügen und auf diese Weise die Funktionalität der Endanwender-Oberfläche in Servoy erweitern. Ihnen steht eine Java API zur Verfügung, die Sie nutzen können, um Beans zu schreiben, die Sie anschließend in Servoy nutzen.

Dataprovider

Ein Dataprovider verwaltet Informationen. Dabei handelt es sich nicht nur um Felder der Datenbank, sondern es kann sich auch um Berechnungen, Aggregationen, globale Variablen oder Formularvariablen handeln.

Methoden

Servoy's Scripting Engine verwendet Methoden, um Codeteile zu identifizieren, die zusammengehören. Im Grunde entspricht eine Methode in Servoy weitgehend einer Funktion in JavaScript. Es handelt sich um eine Zusammenfassung von Code in einer Einheit, die verwendet wird, um in Servoy unterschiedliche Aufgaben zu erfüllen. Eine Methode kann andere Methoden aufrufen, Laufzeiteigenschaften für Servoy's Formularelemente setzen und die Funktionen aufrufen, die in den Scripting-Bibliotheken zur Verfügung stehen.

Funktionen

Eine Funktion entspricht in Servoy einer Methode. Es handelt sich um eine JavaScript-Funktion und wird wie im folgenden Beispiel gezeigt in erster Line als Aufruf einer Methode betrachtet.

```
function onActionAdd(event)
```

Alle Methoden, die in Servoy geschrieben werden, beginnen in der .js-Seite, in der sie geschrieben werden, mit dem Wort function.

Plugin

Ein Plugin erweitert die Funktionalität von Servoy mit Hilfe der Programmiersprache Java. Innerhalb einer Solution erfolgt der Zugriff über das Scripting. Die Plugins können dem Servoy-Programmierer zusätzliche Funktionen zur Verfügung stellen und auch Verbindungen auf Ressourcen außerhalb von Servoy erstellen. Für das Schreiben von Plugins für Servoy steht eine API zur Verfügung.

Das Konzept der Datenverwaltung

Die im Folgenden vorgestellten Konzepte beschreiben die Datenverwaltung in Servoy.

Die Tabelle

Eine Tabelle ist ein vorgegebenes Format von Datensätzen und Feldern, die die Speicherung von Informationen in einer relationalen Datenbank definieren. Daten von erstellten oder bereits vorhandenen Tabellen können in einem oder mehreren Formularen in einer Servoy-Solution verwendet werden.

Column

Eine Column (Feld) ist ein Attribut einer Tabelle in der Datenbank.

Record

Ein Record ist ein einzelner Datensatz in einer Tabelle innerhalb einer SQL-Datenbank.

Foundset

Das Foundset ist ein Servoy-Objekt, das eine Abfrage einer einzelnen Tabelle in der Datenbank repräsentiert. Foundsets behandeln das Caching der Daten für die Anzeige, ohne dass hierfür eine erneute Abfrage der Datenbank erforderlich ist.

Jedes Formular in Servoy, das auf einer Tabelle basiert, enthält ein darunter liegendes Foundset, das die Daten für das Formular verwaltet. Formulare, die auf der gleichen Tabelle basieren, teilen sich ein Foundset, es sei denn, sie werden mittels Scripting oder eine Eigenschaft zur Designzeit auf ein anderes Foundset umgeleitet.

Foundsets werden mittels Scripting erstellt, kontrolliert und zugewiesen, auch Änderungen am Foundset werden mittels Scripting vorgenommen.

Relation

Eine Relation ist ein Servoy-Objekt, das die Verbindung zwischen zwei Tabellen in der Datenbank definiert. Eine Standard-Relation wird durch den Primärschlüssel einer Tabelle definiert, der mit dem Fremdschlüssel einer anderen Tabelle verbunden wird. Hier einige Features einer Relation:

- Die Tabellen können durch Gleichheit der Schlüssel verbunden werden, aber auch durch andere Operatoren (<, >, like usw.)
- Die Tabellen können auf zwei unterschiedlichen Datenbankservern auf unterschiedlichen Plattformen vorhanden sein.
- Die Tabellen können über mehr als ein Feld miteinander verbunden sein.
- Auf der linken Seite der Verbindung kann eine globale Variable verwendet werden. Dieses Feature ist hilfreich, wenn eine Ergebnismenge gefiltert werden soll, indem der Wert der globalen Variablen geändert wird.
- Sowohl die referentielle Integrität als auch weitergegebene Löschvorgänge können erzwungen werden.

Dataset

Ein Dataset ist ein Servoy-Objekt, das Datensätze und Inhalte von Feldern enthalten kann. Datasets können sowohl über Scripting erzeugt werden als auch die Inhalte einer SQL-Abfrage enthalten. Anders als die Foundsets repräsentieren Datasets keinen gültigen Link auf eine Datenbank-Ressource.

Calculation

Eine Calculation (Berechnung) ist ein abgeleiteter Wert, der sich wie ein weiteres Feld in der Tabelle der Datenbank verhält. Die darin enthaltenen Daten sind das Ergebnis einer in JavaScript geschriebenen Funktion, die einer Methode von Servoy entspricht. Der wichtigste Unterschied besteht darin, welche Werte der Funktion zur Verfügung stehen. Eine Funktion, die das Ergebnis für eine Berechnung liefert, kann nicht auf Formular-Variablen oder auf Werte von Formular-Elementen zugreifen, da sie in der Datenschicht der Anwendung arbeitet. Der Funktion stehen die aktuellen Felddaten sowie alle damit zusammen hängenden Daten zur Verfügung.

Es gibt zwei Varianten der Calculations: Gespeicherte Calculations und nicht gespeicherte Calculations. Gespeicherte Calculations befinden sich in Feldern der Datenbank, während sich nicht gespeicherte Calculations ausschließlich im Speicher befinden, wenn die Solution ausgeführt wird.

Calculations werden berechnet, wenn ihr Wert abgefragt wird, beispielsweise, wenn sie in der Benutzeroberfläche angezeigt werden, im Scripting oder auch in einer anderen Berechnung verwendet werden und der Wert noch nicht berechnet oder als ungültig markiert wurde. Eine Calculation wird dann als ungültig markiert, wenn in einem der Datenprovider, auf denen die Calculation beruht, eine Änderung vorgenommen wird.

Wird die Calculation in einem Element der Benutzeroberfläche angezeigt, erhalten diese eine Nachricht, dass sie sich selbst aktualisieren müssen und die Calculation wird ausgeführt, damit der neue Wert berechnet wird.

Handelt es sich um eine gespeicherte Calculation, wird der Datensatz, in dem sie sich befindet, als geändert markiert (er erscheint in der Liste der geänderten Datensätze) und wird bei der nächsten Speicherung in die Datenbank geschrieben.

Aggregation

Eine Aggregation ist ein Wert, der von mehreren Datensätzen innerhalb einer Tabelle abgeleitet wird. Sie wird auf der Basis der Werte innerhalb eines Feldes berechnet. Ähnlich wie bei den Calculations werden die Metadaten einer Aggregation mit der Solution gespeichert, sie basieren aber auf einer Tabelle. Aggregationen können die Funktionen count, sum, maximum, minimum und average verwenden.

Variable

Eine Variable kann definiert werden und enthält zur Laufzeit Daten, ist aber nicht an ein Feld in einer Datenbank gebunden. Variablen können auf einem Formular als Dataprovider verwendet werden und sie können wahlweise den Wirkungsbereich Formular oder Global haben.

Herunterladen und Installieren von Servoy Developer

Der erste Schritt für die Installation von Servoy ist der Download des Installers von unserer Website.

Auswahl des richtigen Installers

Sie haben die Auswahl zwischen zwei Varianten des Servoy All-In-One Installers:

- Den Cross Platform Installer (.jar)
- Den Windows Executable Installer (.exe)

Den Cross Platform Installer können Sie auch unter Windows verwenden, allerdings blocken einige Virens Scanner die Ausführung von .jar-Dateien. Aus diesem Grund stellen wir als Alternative den Windows Executable Installer zur Verfügung.

Den Installer ausführen

Um die Installation zu starten führen Sie den Installer aus. Abhängig von der Plattform klicken Sie entweder doppelt auf die Datei `servoy_installer.jar` oder führen Sie den folgenden Befehl auf der Befehlszeile bzw. im Terminal-Fenster aus:

```
Java -jar servoy_installer.jar
```

Wenn Sie doppelt auf die Datei klicken öffnet die Datei `servoy_installer.jar` als Archivdatei und wird nicht ausgeführt, da die Dateinamenerweiterung nicht zum Öffnen mit Java konfiguriert wurde. Entweder verwenden Sie den Aufruf über die Befehlszeile, nutzen die Option „Öffnen mit“ des Betriebssystems oder Sie konfigurieren das Betriebssystem, so dass .jar-Dateien mit Java geöffnet werden.

Wird der Installer ausgeführt, öffnet sich der Dialog des Installers. Für eine Standardinstallation folgen Sie einfach den Anweisungen des Installers und ändern Sie die Einstellungen nicht. Sie können die Speicherorte ändern und Shortcuts umbenennen, wenn dies für Sie sinnvoll ist.

Weitere Informationen über den All-In-One Installer von Servoy lesen Sie den Eintrag [Running the All-In-One Installer](#) in dieser Dokumentation.

Beachten Sie bitte

Wenn Sie die Standardeinstellungen beibehalten wird automatisch die Datenbank-Engine PostgreSQL mit den Beispieldatenbanken und den Beispielsolutions installiert.

Einführung in die Entwicklungsumgebung von Servoy

Überblick

Servoy Developer ist ein RAD-Werkzeug für das Erstellen von Software-Anwendungen. Die Software umfasst einen grafischen WYSIWYG- (what-you-see-is-what-you-get)-Formular-Generator zusammen mit hoch entwickelten Werkzeugen für die Modellierung der Daten, einer Scripting-Engine sowie stabiler APIs.

Wie in der Version 4.0 wird Servoy Developer als Plug-In für die integrierte Entwicklungsumgebung Eclipse ausgeliefert. Dabei handelt es sich um einen Open Source-Industriestandard. Eclipse ist die weltweit meist verwendete IDE, berühmt für ihre reichhaltigen Features, ihre Produktivität und Erweiterbarkeit. Entwickler, denen Eclipse vertraut ist, fühlen sich in Servoy zu Hause.

Die Konzepte von Eclipse

Workspace

In einem Workspace speichert der Entwickler seine Arbeit. Es handelt sich um eine Dateiablage auf der Entwicklerrmaschine, in der sowohl jedes Projekt als auch die Metadaten für Eclipse abgelegt werden. Der Entwickler kann mehrere Verzeichnisse als Workspace anlegen, er kann aber immer nur mit einem Workspace gleichzeitig arbeiten.

Projekte

Eclipse organisiert den Code in Projekten (Projects). In Servoy Developer finden Sie die folgenden Projektarten:

- Resource Projects – in diesen Projekten werden Informationen über Datenbanken, Styles, Benutzer und i18n-Daten gespeichert.
- Solution Projects – alle Solutions werden in Servoy Developer in separaten Projekten gespeichert.

Perspektiven

Als Perspektive wird in Eclipse eine Sammlung von Ansichten für einen bestimmten Zweck bezeichnet. Kurz gesagt enthält eine Perspektive alles, was Sie in Servoy Developer auf dem Bildschirm sehen.

Wenn Sie Servoy Developer zum ersten Mal öffnen sehen Sie nach dem Schließen des Willkommen-Bildschirms als erstes die Servoy Design Perspektive. Diese Perspektive enthält Ansichten für die Arbeit mit Servoy Solutions sowie alle darin enthaltenen Teile. Andere Perspektiven, die bei der Programmierung mit Servoy verwendet werden, enthalten Debug, Resource und Team Synchronizing und andere.

Die Perspektiven können nach dem Geschmack des Entwicklers angepasst werden. Ansichten können verschoben oder gelöscht werden und die Perspektiven können unter anderen Namen gespeichert werden, so dass es für den Entwickler einfach ist, zu Servoy zu wechseln.

Ansichten

Eine Ansicht (View) ist ein Fenster innerhalb einer Perspektive. Ansichten können dem Entwickler Informationen bereitstellen, Objekte oder Funktionen anzeigen, die in der Solution genutzt werden können oder sie können verwendet werden, um die Solution zu editieren. Editor-Ansichten werden verwendet, um Formulare oder Methoden in Servoy zu erstellen. Dies sind die Ansichten im Zentrum der Perspektive.

Die Design-Perspektive in Servoy

In der Servoy Design-Perspektive verbringt der durchschnittliche Servoy-Entwickler den Großteil seiner Zeit. Diese Perspektive enthält viele Ansichten, die den Entwickler bei seiner Produktivität und Effizienz unterstützen. Wir führen hier einige der wichtigeren Ansichten auf.

Der Solution Explorer

Die Solution Explorer-Ansicht ist eine hierarchische Baumansicht aller Objekte, die in Servoy Developer zur Verfügung stehen. Sie steht im Zentrum der Entwicklungsumgebung von Servoy; jede Aktion, die Sie in Servoy Developer ausführen können, finden Sie im Solution Explorer.

Der Solution Explorer enthält zwei Teile, den Baumansicht der Objekte, die am Kopf angezeigt wird, und eine Kontextliste, die Sie unterhalb der Baumansicht finden. Je nachdem, was in der Baumansicht ausgewählt ist, erscheinen in der Listenansicht unterschiedliche Objekte.

Tip

Wenn Sie mehr Raum wünschen, um die Baumansicht und die Liste im Solution Explorer anzuzeigen, klicken Sie auf die rechte Ecke der Solution Explorer-Ansicht und ziehen Sie sie auf die rechte Seite. Dadurch wird der Solution Explorer vergrößert und die Listenansicht wird auf der rechten Seite platziert. Dadurch wird der Platz für die Ansicht vergrößert.

Die Eigenschaften-Ansicht

Die Eigenschaften-Ansicht (Properties View), die standardmäßig auf der rechten Seite des Arbeitsbereichs angezeigt wird, zeigt verschiedene Designzeit-Eigenschaften an, die der Entwickler ändern kann. Die Eigenschaften, die angezeigt werden, sind davon abhängig, welches Element im Solution Explorer oder im Formular-Designer markiert ist. Alle Objekte, die auf einem Formular angezeigt werden, verfügen über verschiedene Eigenschaften, die er Entwickler ändern kann. Solution-Eigenschaften werden angezeigt, wenn im Solution Explorer die Solution ausgewählt wurde.

Der Formular-Designer

Die Formular-Designer Editoransicht (sie erscheint standardmäßig in der oberen Hälfte des Arbeitsbereichs) ermöglicht es dem Entwickler, den visuellen Teil ihrer Solutions zu erstellen. Nachdem das Formular im Formular-Designer geöffnet wurde, können verschiedene Steuerelemente auf dem Formular platziert werden. Diese Steuerelemente können geändert werden, indem sie markiert und gezogen werden (um die Größe oder die Position zu ändern) oder der Entwickler kann die Toolbars verwenden, die am Kopf des Arbeitsbereichs angezeigt werden, um Aufgaben wie die Ausrichtung oder Gruppierung von Steuerelementen zu erledigen.

Der Script-Editor

In der Script-Editor-Ansicht (die standardmäßig an der gleichen Position wie der Formular-Designer angezeigt wird) kann der Entwickler Methoden erstellen und ändern, die sie in ihren Solutions verwenden. Die Scriptseiten, die im Script-Editor erscheinen, sind entweder an ein Formular, die Solution (als globale Dateien) oder an eine Tabelle (Calculations) gebunden.

Die Problem-Ansicht

Die Problem-Ansicht (Problems View), die standardmäßig in der unteren Mitte des Arbeitsbereichs angezeigt wird, zeigt Fehler und Warnungen, die die aktuell aktive Solution und aktuelle Ressourcen betreffen. Einige Informationen über den Fehler oder die Warnung werden auch in dieser Ansicht angezeigt. Mit einem rechten Mausklick auf den Fehler oder die Warnung rufen Sie die Optionen auf, die Ihnen zur Verfügung stehen. Ein Doppelklick öffnet den Fehler in der entsprechenden Ansicht.

Die Arbeit in Servoy Developer speichern

Während Ihrer Arbeit an Ihrer Solution in Servoy Developer werden die Änderungen in den meisten Editoren und Ansichten auf die gleiche Weise gespeichert. Ihnen stehen dafür die folgenden Möglichkeiten zur Verfügung:

- Wählen Sie im Menü File File -> Save
- Die Tastaturkombination lautet Ctrl+s
- Ein Klick auf das Disk-Icon in der Developer-Toolbar
- Wenn Sie die Editoransicht schließen und Ihre Änderungen nicht gespeichert haben, rufen Sie damit einen Dialog auf, der Sie fragt, ob die Datei vor dem Schließen gespeichert werden soll.

Beachten Sie bitte:

Es ist nicht erforderlich, vor dem Schließen die Entwicklungsumgebung zu speichern. Die Elemente werden einzeln gespeichert. Sobald ein Element gespeichert wird, wird es in die Workspace-Datei geschrieben.

Erstellen Sie eine Solution

Durch das Erstellen einer neuen Solution (oder den Import einer Solution) in Servoy erstellen Sie ein neues Eclipse-Projekt, was bedeutet, dass im aktuellen Workspace-Verzeichnis ein neues Verzeichnis angelegt wird. Das neue Verzeichnis wird wie die Solution benannt.

Erstellen Sie eine neue Solution

1. Wählen Sie eine der folgenden Optionen:
 - a. Wählen Sie in Servoy Developer den Menüeintrag File -> New -> Servoy Solution.
 - b. Klicken Sie in der obersten Toolbar auf die Schaltfläche New. Daraufhin wird ein Fenster aufgerufen, in dem Sie den Knoten Servoy wählen und klicken Sie auf Next.
 - c. Klicken Sie mit der rechten Maustaste auf den Knoten All Solutions oder auf die aktive Solution. Wählen Sie aus dem Menü Create a new solution. Damit wird ein neues Fenster aufgerufen.
2. Jetzt sollte auf Ihrem Bildschirm ein Fenster mit der Überschrift „Create a New solution“ zu sehen sein. Geben Sie hier den Namen für die Solution ein.
3. Wählen Sie den Typen der Solution. In den meisten Fällen ist Normal ausreichend. Sie können auch angeben, dass Ihre Solution ein Modul ist, wenn Sie sich sicher sind, dass sie immer als Modul verwendet wird. Außerdem haben Sie die Wahl zwischen „Smart Client only“ und „Web Client“.
4. Wählen Sie ein Ressourcen-Projekt. In der Regel können Sie den Standardwert übernehmen, aber Sie können auch Ihr eigenes Projekt erstellen, wenn dies erforderlich sein sollte.
5. Klicken Sie auf Finish. Die neue Solution wird im Solution Explorer aktiv.

Tipp

Wir empfehlen, pro Workspace nur ein Ressourcen-Projekt zu verwenden, um Unordnung zu vermeiden.

Importieren Sie eine vorhandene Solution

Es kann der Fall auftreten, dass Sie eine vorhandene Solution aus einer Exportdatei (.servoy) verwenden wollen. Wenn Sie die Standardinstallation durchgeführt haben, finden Sie im Verzeichnis <servoy install folder> /application_server/solutions/examples Beispiel-Solutions, die Sie installieren und verwenden können.

So importieren Sie eine Solution in Servoy Developer:

1. Wählen Sie eine der folgenden Methoden:
 - a. Wählen Sie File -> Import. Damit rufen Sie ein Fenster auf, in dem Sie Servoy -> Import Solution wählen und klicken Sie auf Next.
 - b. Klicken Sie mit der rechten Maustaste auf den Knoten „All solutions“ im Solution Explorer und wählen Sie „Import a solution“.
2. Auf Ihrem Bildschirm sollten Sie jetzt ein Fenster mit der Überschrift „Import a solution“ sehen.
3. Wählen Sie die .servoy-Datei, die Sie importieren wollen.

Wenn Sie Servoy unter Java 6 Update 26 ausführen und auf die Schaltfläche „Browse“ klicken, wird der Dialog „File Select“ nicht aufgerufen. Der Grund ist ein Problem in diesem Update von Java, das Oracle hoffentlich im nächsten Update beheben wird. Ihnen stehen zwei Workarounds zur Verfügung:

- Geben Sie den Pfad manuell im Feld neben der Schaltfläche ein.
 - Führen Sie ein Downgrade auf eine ältere Java-Version aus.
4. Lassen Sie für die Beispiel-Solutions die anderen Optionen auf den bereits eingestellten Werten.
 5. Sie können entscheiden, ein anderes Ressourcen-Projekt zu verwenden (vgl. den Tipp weiter oben).
 6. Klicken Sie auf Finish.
 7. Es ist möglich, dass noch weitere Dialoge angezeigt werden, die Sie nach dem Import von Beispieldaten, Styles o.ä. fragen. Wenn Sie sich die Beispiel-Solutions ansehen, beantworten Sie diese Fragen einfach mit Yes. Sind Styles vorhanden, können Sie diesen Schritt entweder überspringen oder die Styles überschreiben.
 8. Klicken Sie im Ergebnisfenster auf OK. Die neu importierte Solution wird im Solution Provider die aktive Solution.

Datenbank-Verbindungen verwalten

Überblick

Nahezu jede Servoy-Anwendung verwendet eine Datenbank, um Benutzerdaten und Metadaten der Anwendung zu speichern. Servoy kann sich mit jeder Datenbank verbinden, für die ein JDBC-Treiber zur Verfügung steht. Servoy stellt Ihnen die Datenbank PostgreSQL zur Verfügung, für den Fall, dass Sie nicht bereits über ein Datenbank-Managementsystem (DBMS) verfügen.

Erstellen einer neuen Datenbank in PostgreSQL

Sie können in Servoy Developer aus der Solution Explorer-Baumansicht heraus eine neue Datenbank erstellen.

1. Erweitern Sie den Knoten Database Servers (unterhalb von Ressourcen) im Tab Developer Solution Explorer und klicken Sie mit der rechten Maustaste auf eine beliebige PostgreSQL-Datenbank (alle Beispieldatenbanken, die Ihnen Servoy liefert, sind PostgreSQL), um das Kontextmenü aufzurufen.
2. Wählen Sie den Menüeintrag „Create PostgreSQL Database“.
3. Damit rufen Sie einen Dialog auf, in dem Sie einen Namen für die Datenbank eingeben.
4. Es wird ein weiterer Dialog aufgerufen, der Sie informiert, dass die Datenbank erstellt wird.
5. Ein letzter Dialog teilt Ihnen mit, wenn die Erstellung beendet ist. Klicken Sie dort auf OK.

Verbinden mit einer vorhandenen Datenbank

Sie können sich in Servoy Developer auch mit einer vorhandenen Datenbank verbinden.

1. Legen Sie den JDBC-Treiber fest, der für den Aufbau einer Verbindung mit Ihrer Datenbank erforderlich ist. In der Regel erfahren Sie von Ihrem DBMS-Hersteller oder aus der Dokumentation, welcher Treiber der richtige ist. Eine andere Hilfequelle kann das Servoy-Forum sein, da möglicherweise andere Entwickler Ihre Datenbank zusammen mit Servoy verwendet haben.
2. Installieren Sie den Treiben, indem Sie die .jar-Datei im Verzeichnis application_server/drivers in Ihrem Servoy-Installationsverzeichnis platzieren.
3. Starten Sie Servoy Developer neu.
4. Erstellen Sie eine neue Datenbankserver-Verbindung, indem Sie mit der rechten Maustaste auf den Knoten Database Servers (unterhalb von Resources) klicken und im Kontextmenü New Server wählen. Ein Untermenü bietet die Auswahl zwischen verschiedenen häufig verwendeten Datenbankoptionen, die in die Datenbank-Verbindung eingetragen werden, um das Setup zu vereinfachen. Sie können diese Antworten aber auch leer lassen.

Beachten Sie bitte

Auch wenn eine Datenbank hier aufgelistet ist, müssen Sie trotzdem im Verzeichnis drivers einen Treiber installieren, um sich mit der Datenbank zu verbinden.

5. Geben Sie den Namen des Servers, den Benutzernamen, das Passwort, die URL und den Treiber für Ihre Datenbank ein. Die korrekten Werte für die URL und den Treiber erfahren Sie in der Dokumentation Ihrer Datenbank oder durch die vorgeschlagenen Werte, die bereits eingetragen wurden, als Sie Ihre Datenbank im Menü New Server ausgewählt haben.

Beachten Sie bitte

Das Benutzerkonto sollte über Administratorrechte verfügen, so dass Sie in Servoy den vollständigen Zugriff auf Ihre Datenbankobjekte nutzen können.

6. Speichern Sie den Tab, indem Sie in der Toolbar auf die Schaltfläche Save klicken oder indem Sie die Tastenkombination Ctrl+s nutzen.
7. Jetzt sollte Ihre Datenbank in der Liste mit den Datenbankservern angezeigt werden.

Tabellen und Felder erstellen

Es ist sinnvoll, Ihr Datenmodell (Tabellen, Primärschlüssel, Fremdschlüssel und Felder) zu erstellen, bevor Sie mit Ihrer Solution beginnen. Sie können die Objekte in der Datenbank mit den Datenbank-Werkzeugen in der Verwaltung Ihrer Datenbank erstellen oder Sie können die Objekte aus Servoy Developer heraus erzeugen.

Das Erstellen der Tabellen

Auf die folgende Weise erstellen Sie in Servoy Developer eine Tabelle:

1. Klicken Sie mit der rechten Maustaste im Knoten Database Servers auf den Namen der Datenbank und wählen Sie im Kontextmenü Create Table oder Sie markieren den Namen der Datenbank und klicken anschließend in der Toolbar der Listenansicht des Solution Explorers auf die Schaltfläche Create Table.
2. Ein Dialog fragt Sie nach einem Namen für die neue Tabelle. Geben Sie den Namen ein und klicken Sie auf OK.
3. Die Tabelleneditor-Ansicht erscheint, so dass Sie neue Felder anlegen können.

Beachten Sie bitte:

Wird eine neue Tabelle erstellt, fügt Servoy im Tabelleneditor ein Feld für den Primärschlüssel ein. Alle in Servoy verwendeten Tabellen müssen über einen Primärschlüssel oder über eine eindeutige Identifikation für einen Datensatz verfügen.

Das Erstellen der Felder

Auf die folgende Weise erstellen Sie Felder in Servoy Developer:

1. Öffnen Sie den Tabelleneditor, indem Sie im Solution Explorer den Datenbankserver markieren und anschließend in der Listenansicht eine der folgenden Aktionen ausführen:
 - a. Doppelklicken Sie auf den Namen der Tabelle.
 - b. Klicken Sie mit der rechten Maustaste auf den Namen der Tabelle und wählen Sie im Kontextmenü den Eintrag Edit table/view.
 - c. Markieren Sie die Tabelle und klicken Sie in der Toolbar list view auf die Schaltfläche edit table.
2. Der Tabelleneditor wird aufgerufen und ermöglicht Ihnen, die Felder in der Tabelle zu ändern.
3. Um ein Feld hinzuzufügen klicken Sie auf die Schaltfläche Add. Ändern Sie den Namen des Feldes von „type here“ in den von Ihnen gewünschten Namen. Geben Sie den Feldtypen und gegebenenfalls die Größe an. Vergessen Sie nicht, am Ende Ihre Arbeit zu speichern.
4. Um ein Feld zu löschen, klicken Sie auf die Schaltfläche Remove.

Formulare erstellen

Überblick

Das wichtigste Objekt der Benutzeroberfläche in Servoy ist ein Formular. Formulare sind auch ein Gültigkeitsbereich in Servoy, was bedeutet, dass Formulare über Variablen und Methoden verfügen können, die nur zu ihnen gehören. Weitere Informationen über Gültigkeitsbereiche finden Sie im Eintrag Implementing Business Logic.

Ein Formular erstellen

Um Ihnen die Möglichkeiten aufzuzeigen, die der New Form Wizard bereitstellt, verwenden wir hier die Verbindung `example_data` und die Tabelle `customers`.

1. Bevor Sie einen Namen für das Formular eingeben, markieren Sie die dafür erforderliche Tabelle. Klicken Sie auf die Schaltfläche ... und öffnen Sie dadurch eine Baumansicht, in der Sie aus den verfügbaren Datenbankverbindungen die Tabelle auswählen. Soll das Formular nicht auf einer Tabelle basieren wählen Sie `-none-`. In unserem Beispiel wählen Sie `example_data.customers`.
2. Beachten Sie, dass der Name der Tabelle im Feld `name` angezeigt wird. Sie können diesen Namen ändern. Denken Sie daran, dass innerhalb einer Solution keine zwei Formulare den gleichen Namen haben dürfen. Für unser Beispiel nennen Sie Ihr Formular `customers_dtl`.

Tip

Namenskonventionen können Ihnen helfen, den Überblick über Ihre Formulare zu behalten, so dass die Formulare Ihnen bereits durch Ihren Namen mitteilen, welchem Zweck sie dienen.

3. Wenn Sie mit Stylesheets arbeiten, können Sie den Namen des von Ihnen gewünschten Stylesheets markieren.
4. Prüfen Sie, ob die genannte Solution die richtige Solution für Ihr Formular ist.
5. Klicken Sie auf Next.
6. Wählen Sie die Datenprovider, die auf dem Formular angezeigt werden sollen. Beachten Sie, dass Sie Felder, Calculations, Formularvariablen, globale Variablen und Dataprovider auswählen können. Wenn Sie bei der Auswahl die Shift-Taste gedrückt halten, markieren Sie einen Bereich, wenn Sie die Ctrl-Taste gedrückt halten, haben Sie die Möglichkeit mehrere Elemente auszuwählen, die nicht nebeneinander liegen müssen. Für unser Beispiel wählen Sie einige Felder aus der Tabelle.

7. Es gibt mehrere Möglichkeiten, wie die Datenprovider dem Formular hinzugefügt werden. In den meisten Fällen benötigen Sie nur die Option Place with Labels. Für dieses Beispiel verwenden Sie ausschließlich diese Option.

Beachten Sie bitte:

In der Regel werden Formulare in der Datensatz-Ansicht erstellt, aber wenn Sie „Fill Text property“ und „Place horizontal“ auswählen, wird das Formular in die Tabellenansicht gesetzt.

8. Klicken Sie auf Finish.
9. Das neu erstellte Formular wird in den Formular-Designer geladen.

Ein Formular ändern

Um im Formular-Designer ein vorhandenes Formular zu ändern führen Sie die folgenden Schritte aus:

1. Markieren Sie das Formular im Solution Explorer.
2. Führen Sie einen der folgenden Schritte aus:
 - a. Drücken Sie die Tastenkombination Ctrl+Shift+a
 - b. Klicken Sie mit der rechten Maustaste in der Baumansicht auf den Namen des Formulars oder auf einen der dem Formular untergeordneten Knoten und wählen Sie im Kontextmenü den Eintrag „Open in Form Designer“.
 - c. Klicken Sie mit der rechten Maustaste in der Listenansicht des Solution Explorers auf den Namen des Formulars und wählen Sie im Kontextmenü den Eintrag „Open in Form Designer“.
3. Jetzt sollte das Formular im Formular-Designer-Editor in der Mitte Ihres Arbeitsbereichs angezeigt werden.

Um im Script Editor ein Formular zu ändern führen Sie die folgenden Schritte aus:

1. Markieren Sie das Formular im Solution Explorer.
2. Führen Sie einen der folgenden Schritte aus:
 - a. Drücken Sie die Tastenkombination Ctrl+Shift+z.
 - b. Klicken Sie mit der rechten Maustaste in der Baumansicht auf den Namen des Formular oder auf einen ihm untergeordneten Knoten und wählen Sie im Kontextmenü den Eintrag „Open in Script Editor“.
 - c. Erstellen Sie eine neue Methode des Formulars. Details hierfür finden Sie im Eintrag Implementing Business Logic.

3. Jetzt sollte das Formular in der Script Editor–Ansicht in der Mitte Ihres Arbeitsbereichs angezeigt werden.

Beachten Sie bitte

Wenn Sie in einer Editor–Ansicht mit der rechten Maustaste klicken rufen Sie ein Kontextmenü auf, das die Option beinhaltet, das Formular in der anderen Editor–Ansicht zu öffnen. Ein Beispiel: Sie haben ein Formular im Form Designer geöffnet. Jetzt können Sie mit der rechten Maustaste klicken und im Kontextmenü den Eintrag „Open in Script Editor“ wählen.

Ein Formular speichern

Um ein Formular zu speichern stehen Ihnen die gleichen Möglichkeiten zur Verfügung wie in jeder anderen Editor–Ansicht in Servoy Developer. Vergleichen Sie dazu den Eintrag „Einführung in die Entwicklungsumgebung von Servoy“.

Clients starten und debuggen

Überblick

In Servoy haben Sie die Möglichkeit, Ihre Anwendungen direkt innerhalb der Entwicklungsumgebung zu testen und zu debuggen. Dies gilt sowohl für den Smart Client als auch für den Web Client. Dies ist möglich, weil beim Start von Servoy Developer im Hintergrund auch eine lokale Instanz des Anwendungsservers gestartet wird. Dieser ermöglicht zwei Verbindungen durch den Client – jeweils eine für den Smart Client und eine für den Web Client.

Mehr noch – wenn Sie an Ihrer Anwendung Änderungen vornehmen, verteilt Servoy diese Änderungen an die Debug-Clients (Smart und Web), so dass Sie die Änderungen in Echtzeit sehen, testen und debuggen können, ohne dafür den Client neu zu starten.

Beachten Sie bitte

Das Debuggen von Browseranwendungen ist traditionell sehr schwierig, da die Logik asynchron zwischen dem Client und dem Server aufgeteilt wird. Da Servoy Ihre Geschäftslogik vollständig auf der Serverseite ausführt, können Sie Browseranwendungen direkt aus der IDE heraus testen und debuggen.

Den Debug-Smart Client aufrufen

- Um die aktive Solution im Debug-Smart Client aufzurufen, wählen Sie im Menü Actions -> Start SmartClient oder
- klicken Sie einfach auf die Schaltfläche Start SmartClient auf der Toolbar.

Den Debug-Web Client aufrufen

- Um die aktive Solution im Debug-Web Client aufzurufen, wählen Sie im Menü Actions -> Start WebClient oder
- klicken Sie einfach auf die Schaltfläche Start WebClient auf der Toolbar.

Formulare innerhalb eines Formulars platzieren

Formulare können mit Hilfe eines Tabpanels innerhalb anderer Formulare platziert werden. Ein Tabpanel kann mit der Solution andere Formulare anzeigen, auch Formulare aus anderen Modulen.

Ein Formular innerhalb eines Formulars platzieren

Um ein Formular innerhalb eines anderen Formulars zu platzieren benötigen wir die folgenden Servoy-Objekte:

- Eine Relation
- Ein anderes Formular
- Ein Tabpanel

Eine Relation erstellen

Bevor wir ein Formular auf einem anderen Formular platzieren können müssen wir eine Relation erstellen. Eine Relation ist ein Link zwischen zwei Tabellen oder einer globalen Variablen und einer Tabelle.

In diesem Beispiel erstellen wir eine Relation in der Datenbank `example_data` zwischen den Tabellen `customers` und `orders`.

So erstellen Sie eine Relation:

1. Führen Sie einen der folgenden Schritte aus:
 - a. Wählen Sie im Menü `File -> New -> Other`. Es erscheint ein Fenster mit einer Baumansicht. Wählen Sie `Servoy -> New Relations`. Erneut wird ein neues Fenster aufgerufen. Markieren Sie die Solution, in der die Relation platziert werden soll und geben Sie der Relation einen Namen.
 - b. Wählen Sie die Dropdown-Liste auf der Schaltfläche `New` in der `Developer-Toolbar` und wählen Sie dort `Other`. Damit rufen Sie ein Fenster mit einer Baumansicht auf. Wählen Sie `Servoy -> New Relations`. Erneut wird ein neues Fenster aufgerufen. Markieren Sie die Solution, in der die Relation platziert werden soll und geben Sie der Relation einen Namen.
 - c. Wählen Sie im `Solution Explorer` unterhalb der richtigen Solution den Knoten `Relations`. Klicken Sie in der `Toolbar` der `Listenansicht` des `Solution Explorers` auf die Schaltfläche `Create Relation`.
 - d. Klicken Sie mit der rechten Maustaste im `Solution Explorer` unterhalb der richtigen Solution auf den Knoten `Relations`. Wählen Sie im `Kontextmenü` den Eintrag `Create Relation`.

Tipp

Die letzten beiden Möglichkeiten sind eventuell sinnvoller, da bei diesen Vorgehensweisen `Servoy Developer` die Relation für Sie benennt.

2. Es erscheint in der Mitte des Servoy-Arbeitsbereichs eine Editor-Ansicht mit der neuen Relation. Ändern Sie den Namen jetzt nicht.
3. Markieren Sie die Tabelle auf der linken Seite der Relation, indem Sie auf die Schaltfläche ... klicken. Für dieses Beispiel wählen Sie `example_data.customers`.
4. Markieren Sie die Tabelle auf der rechten Seite der Relation, indem Sie auf die Schaltfläche ... klicken. Für dieses Beispiel wählen Sie `example_data.customers`.
5. Nachdem Sie die Tabellen ausgewählt haben, wird der Name für die Relation automatisch erzeugt. Dieser Name muss eindeutig sein, so dass der vorgegebene Name ausreichend ist und nicht geändert werden muss. So sollte auch in diesem Beispiel der vorgegebene Name in Ordnung sein.
6. Markieren Sie den Primärschlüssel der übergeordneten (linken) Tabelle und den entsprechenden Fremdschlüssel der untergeordneten (rechten) Tabelle. In diesem Beispiel achten Sie darauf, dass die `customer_id` der `customer_id` entspricht.

Beachten Sie bitte

Im Regelfall versucht Servoy Developer, diese ids zu verwenden, wenn sie den gleichen Namen haben und es eine Entsprechung zwischen dem Primärschlüssel und dem Fremdschlüssel gibt.

7. Richten Sie die Regeln für die Relation ein. In diesem Beispiel lassen Sie die Regeln auf ihren Standardwerten.
8. Speichern Sie den Relationen-Editor. Vergleichen Sie dazu den Eintrag „Einführung in die Entwicklungsumgebung von Servoy“.

Ein Formular erstellen

Bevor ein Formular auf einem anderen Formular platziert werden kann muss das Formular erstellt werden. Sie können neue Formulare nicht im gleichen Editor erstellen; jedes Formular öffnet seinen eigenen Editor. Informationen über das Erstellen von Formularen finden Sie im Eintrag „Formulare erstellen“.

Für dieses Beispiel erstellen Sie ein neues Formular auf der Basis der Tabelle `example_data.orders`. Setzen Sie das Formular auf Tabellenansicht.

Um ein Formular mit einer Tabellenansicht zu erstellen nutzen Sie die Anleitung im Eintrag „Creating Forms“, allerdings mit den folgenden Ausnahmen:

- Fügen Sie mit den aktivierten Optionen „Fill text property“ und „Place horizontal“ Datenprovider hinzu.
- Achten Sie darauf, dass sich das Formular in der Tabellenansicht befindet, indem Sie die Formulareigenschaft `view` anklicken, damit sie auf `Table view (locked)` gesetzt ist.

Ein Tabpanel erstellen

Ein Tabpanel ist ein Steuerelement/Objekt von Servoy, das auf einem Formular platziert werden kann. Weitere Informationen über Tabpanels finden Sie im Eintrag „Einführung in die Entwicklungsumgebung von Servoy“. In diesem Beispiel werden wir das neu erstellte Formular orders auf dem Detailformular customers platzieren.

Um ein Tabpanel auf einem Formular zu platzieren führen Sie die folgenden Schritte aus:

1. Stellen Sie sicher, dass das erforderliche Formular im Formular-Designer geöffnet und ausgewählt ist.
2. Führen Sie einen der folgenden Schritte aus:
 - a. Drücken Sie die Tastenkombination Ctrl+alt+t.
 - b. Wählen Sie im Menü Elements -> Place TabPanel.
 - c. Klicken Sie auf der Developer Toolbar auf die Schaltfläche TabPanel.
3. Es wird ein Fenster aufgerufen, auf dem Sie ein Tabpanel-Formular auswählen können.
4. Wählen Sie ein Formular.
 - a. Um ein verbundenes Formular auszuwählen, wählen Sie das Formular mit Hilfe der entsprechenden Relation. Für dieses Beispiel öffnen Sie den Knoten der Relation customer_to_orders und wählen Sie das neu erstellte Formular orders.
 - b. Um ein nicht verbundenes Formular auszuwählen, wählen Sie das Formular über den Knoten Unrelated.
5. Das Tabpanel wird mit dem gewählten Formular angezeigt.
6. Um ein anderes Formular im Tabpanel anzuzeigen wählen Sie das Tabpanel im Formular-Designer und verwenden Sie einen der in Schritt 2 beschriebenen Befehle.

Zusammenfassung

Das Formular customers sollte jetzt innerhalb des Tabpanels das Formular orders anzeigen. Da das Formular mittels einer Relation hinzugefügt wurde, gehören die Daten im Formular orders zum gewählten Kunden. Wird der Datensatz in Customer gewechselt, werden auch die Datensätze im Formular auf dem TabPanel neu geladen.

Implementieren der Geschäftslogik

Überblick

Obwohl Servoy vollständig auf Java basiert, müssen Sie während der Entwicklung oder Bereitstellung kein Java schreiben.

Stattdessen ist die gesamte Geschäftslogik vollständig mit JavaScript implementiert. JavaScript ist ein Internetstandard, einfach zu erlernen und die am weitesten verbreitete Skriptsprache weltweit. JavaScript ist erheblich produktiver als die Codierung in reinem Java und Servoy bietet robuste APIs an, mit deren Hilfe die Geschäftslogik einfach und schnell implementieren lässt.

Beachten Sie bitte

Entwickler, die mit JavaScript vertraut sind, könnten Probleme mit der Browserunterstützung und der Ausführungsgeschwindigkeit anführen. Hier möchten wir bemerken, dass Servoy kein JavaScript liefert. Der in Servoy geschriebene Code wird mit Mozillas Projekt Rhino ausgeliefert, einer auf Java basierenden JavaScript-Implementierung.

Was bedeutet das?

Alle Methoden werden in Java ausgeführt (was ganz erheblich schneller ist als interpretiertes JavaScript).

Die Geschäftslogik wird niemals in den Browser geladen oder dort ausgeführt, so dass Probleme mit der Unterstützung unterschiedlicher Browser vermieden werden. Erfahrene Entwickler können optional APIs von Drittherstellern verwenden, um Java-Code direkt in ihren Servoy-Methoden zu verwenden.

Die Gültigkeitsbereiche

Der Gültigkeitsbereich (Scope) definiert die Domain, in der ein Code ausgeführt wird und er legt weitergehend den Namespace fest, durch den die per Skript geschriebenen Objekte referenziert werden. JavaScript-Code (Funktionen und Variablen) können einen dieser beiden Gültigkeitsbereiche aufweisen:

Global: Der Code steht in der Datei `globals.js`. Auf ihn wird über den Namespace `globals` zugegriffen. Ein Beispiel:

```
globals.createNewCustomer(); // ruft die globale Methode auf
```

Formular: Der Code steht in der Datei `formName.js`. Auf ihn wird über den Namespace `forms.formName` zugegriffen. Auch hier ein Beispiel:

```
forms.customers.controller.newRecord(); // ruft ein Formularobjekt in einem anderen Gültigkeitsbereich auf
```

Im Vergleich:

```
function createNewCustomer(){ // Eine Methode, die im Formular  
    'customers' definiert wurde  
  
    controller.newRecord(); // ruft das gleiche Formularobjekt im  
    Gültigkeitsbereich des Formulars auf. Beachten Sie, dass der vollständig  
    qualifizierte Namespace nicht erforderlich ist.  
  
}
```

Eine Variable deklarieren

Eine globale Variable deklarieren (zwei Möglichkeiten)

- In der Baumansicht des Solution Providers navigieren Sie zum Knoten Active Solution -> Globals -> variables. Klicken Sie mit der rechten Maustaste auf diesen Knoten und wählen Sie im Kontextmenü den Eintrag Create Variable. Wählen Sie einen Variablennamen, einen Datentyp und optional einen Anfangswert und klicken Sie auf OK. Die Variablendeklaration wird in der Datei globals.js generiert, die im Script Editor geöffnet wird.
- In der Baumansicht des Solution Explorers navigieren Sie zum Knoten Active Solution -> Globals -> variables. Markieren Sie den Knoten und klicken Sie in der unteren Toolbar des Solution Explorers auf die Schaltfläche Create Variable. Wählen Sie einen Variablennamen, einen Datentyp sowie optional einen Anfangswert und klicken Sie auf OK. Die Variablendeklaration wird in der Datei globals.js generiert, die im Script Editor geöffnet wird.

Eine Formularvariable deklarieren (zwei Möglichkeiten)

- In der Baumansicht des Solution Explorers navigieren Sie zum Knoten Active Solution -> Forms -> myForm -> variables. Klicken Sie mit der rechten Maustaste auf diesen Knoten und wählen Sie im Kontextmenü den Eintrag Create Variable. Wählen Sie einen Variablennamen, einen Datentyp und optional einen Anfangswert und klicken Sie auf OK. Die Variablendeklaration wird in der Datei globals.js generiert, die im Script Editor geöffnet wird.
- In der Baumansicht des Solution Explorers navigieren Sie zum Knoten Active Solution -> Forms -> myForm -> variables. Markieren Sie den Knoten und klicken Sie in der unteren Toolbar des Solution Explorers auf die Schaltfläche Create Variable. Wählen Sie einen Variablennamen, einen Datentyp sowie optional einen Anfangswert und klicken Sie auf OK. Die Variablendeklaration wird in der Datei globals.js generiert, die im Script Editor geöffnet wird.

Eine Methode erstellen

Erstellen Sie eine globale Variable auf eine von zwei Möglichkeiten

- In der Baumansicht des Solution Explorers navigieren Sie zum Knoten Active Solution -> Globals. Klicken Sie mit der rechten Maustaste auf diesen Knoten und wählen Sie im Kontextmenü den Eintrag Create Method. Wählen Sie einen Methodennamen und klicken Sie auf OK. Die Methodendeklaration wird in der Datei globals.js generiert, die im Script Editor geöffnet wird.
- In der Baumansicht des Solution Explorers navigieren Sie zum Knoten Active Solution -> Globals. Markieren Sie den Knoten und klicken Sie in der unteren Toolbar des Solution Explorers auf die Schaltfläche Create Method. Wählen Sie einen Methodennamen und klicken Sie auf OK. Die Methodendeklaration wird in der Datei globals.js generiert, die im Script Editor geöffnet wird.

Erstellen Sie eine Formularvariable auf eine von zwei Möglichkeiten

- In der Baumansicht des Solution Explorers navigieren Sie zum Knoten Active Solution -> Forms -> myFormnode. Klicken Sie mit der rechten Maustaste auf diesen Knoten und wählen Sie im Kontextmenü den Eintrag Create Method. Wählen Sie einen Methodennamen und klicken Sie auf OK. Die Methodendeklaration wird in der Datei myForm.js generiert, die im Script Editor geöffnet wird.
- In der Baumansicht des Solution Explorers navigieren Sie zum Knoten Active Solution -> Forms -> myForm. Markieren Sie den Knoten und klicken Sie in der unteren Toolbar des Solution Explorers auf die Schaltfläche Create Method. Wählen Sie einen Methodennamen und klicken Sie auf OK. Die Methodendeklaration wird in der Datei myForm.js generiert, die im Script Editor geöffnet wird.

Implementieren einer grundlegenden Geschäftslogik

Um eine Geschäftslogik zu implementieren erstellen Sie eine Methode und füllen den Body der JavaScript-Funktion mit ausführbarem Code.

Das folgende Beispiel implementiert die Funktionalität, den ausgewählten Datensatz auf einem Formular anzuzeigen:

```
function nextRecord() {  
  
    var index = controller.getSelectedIndex(); // den aktuellen Index  
    speichern  
  
    controller.setSelectedIndex(index+1); // den Index um 1 erhöhen  
  
}
```

Dieses Beispiel verwendet das Objekt Controller des Formulars, das Teil der JavaScript API ist, die von Servoy bereitgestellt wird.

Die Entwickler müssen sich nicht an die API erinnern oder sie überprüfen. Die Scripting-APIs sind selbstdokumentierend und im Script Editor lassen sich auf einfache Weise Codefragmente einfügen.

Ein Codefragment einfügen

Jede Scripting-API (auch Methoden, die durch einen Entwickler geschrieben wurden) können auf zwei unterschiedliche Weisen direkt in den Script-Editor eingefügt werden:

- In der Baumansicht des Solution Explorers navigieren Sie zu einem Knoten für eine Ressource, die ein Script aufnehmen kann (z.B. Active Solution -> Forms -> myForm -> controller) und markieren diese. In der Liste der Methoden und Eigenschaften, die Sie im unteren Teil des Solution Explorers sehen, klicken Sie mit der rechten Maustaste auf die Methode oder Eigenschaft, die Sie aufrufen wollen und wählen Sie im Kontextmenü den Eintrag Move Code. Der Code wird in den Script Editor kopiert und mit dem korrekten Namespace für den aktuellen Gültigkeitsbereich referenziert. Es ist möglich, dass Sie spezielle Argumente eingeben müssen.
- In der Baumansicht des Solution Explorers navigieren Sie zu einem Knoten für eine Ressource, die ein Script aufnehmen kann und markieren diese. In der Liste der Methoden und Eigenschaften, die Sie im unteren Teil des Solution Explorers sehen, markieren Sie die Methode oder Eigenschaft, die Sie aufrufen wollen und klicken Sie auf die Schaltfläche Move Code in der unteren Toolbar des Solution Explorers. Der Code wird in den Script Editor kopiert und mit dem korrekten Namespace für den aktuellen Gültigkeitsbereich referenziert. Es ist möglich, dass Sie spezielle Argumente eingeben müssen.

Beispielcode einfügen

In jede Scripting-API kann der kommentierte Beispielcode auf zwei unterschiedliche Weisen direkt in den Script-Editor eingefügt werden:

- In der Baumansicht des Solution Explorers navigieren Sie zu einem Knoten für eine Ressource, die ein Script aufnehmen kann (z.B. Active Solution -> Forms -> myForm -> controller) und markieren diese. In der Liste der Methoden und Eigenschaften, die Sie im unteren Teil des Solution Explorers sehen, klicken Sie mit der rechten Maustaste auf die Methode oder Eigenschaft, die Sie aufrufen wollen und wählen im Kontextmenü den Eintrag Move Sample. Der lange auskommentierte Code wird in den Script Editor kopiert.
- In der Baumansicht des Solution Explorers navigieren Sie zu einem Knoten für eine Ressource, die ein Script aufnehmen kann und markieren diese. In der Liste der Methoden und Eigenschaften, die Sie im unteren Teil des Solution Explorers sehen, markieren Sie die Methode oder Eigenschaft, die Sie aufrufen wollen und klicken auf die Schaltfläche Move Sample in der unteren Toolbar des Solution Explorers. Der lange auskommentierte Code wird in den Script Editor kopiert.

Tip

Jede Scripting-API (auch Methoden, die durch einen Entwickler geschrieben wurden) kann automatisch durch Tastatureingaben vervollständigt werden.

Um die automatische Vervollständigung zu nutzen, beginnen Sie, den Code einzugeben, den Sie ausführen wollen und drücken dann auf der Tastatur Ctrl + Leertaste. Es wird eine Liste mit Scripting-Objekten eingeblendet, aus der Sie während der Eingabe wählen können. Dies ist eine sehr produktive und fehlerfreie Methode Code zu schreiben, wenn Sie mit den APIs etwas vertrauter sind.

Tip

Wenn Sie mit der Syntax von JavaScript nicht vertraut sind, ist dies auch OK. Navigieren Sie im Solution Explorer zum Knoten JS Lib. Dieses Objekt enthält eine Liste mit APIs für die Arbeit mit JavaScript und den nativen Datentypen. Es enthält auch die Syntax-Vervollständigung für häufig verwendete Anweisungen im Knoten Statements.

Gestaltung mit CSS

Überblick

Servoy unterstützt die Verwendung der Cascading Style Sheets (CSS) von HTML für die Gestaltung von Anwendungen. Diese Textdateien werden im Ressourcen-Projekt gespeichert. Ein Ressourcen-Projekt kann viele CSS-Dateien enthalten und in einer Solution kann mehr als eine CSS verwendet werden.

Die Styles in einer CSS sind einem Formular sowie den Elementen auf einem Formular zugeordnet. Auch wenn eine Solution mehrere Style Sheets verwenden kann, kann immer nur ein Style Sheet gleichzeitig auf einem Formular verwendet werden.

Zu den Vorteilen der Verwendung von CSS in Servoy gehören:

- Die Benutzeroberfläche wird konsistent. Formularelemente in der Anwendung können die gleichen Styles verwenden.
- Sie ermöglichen es dem Entwickler, Styleänderungen für mehrere oder alle Formulare vorzunehmen, indem er im CSS einen einzigen Wert ändert.
- Style Sheets können in mehreren Anwendungen verwendet werden, wodurch die Anwendungen eines Unternehmens konsistent sind.
- Die Konsistenz einer Anwendung, die von einem Team entwickelt wird, wird gewährleistet. Es wird für das Team einfacher, allen Formularen das gleiche Aussehen zu geben.
- Style Sheets können programmgesteuert geändert werden. Dadurch ist es einem Entwickler möglich, Styles an den persönlichen Geschmack der einzelnen Anwender anzupassen oder in regelmäßigen Abständen den Style der Anwendung zu ändern. Detaillierte Informationen zu dieser Funktionalität finden Sie in der Beschreibung der Funktion `overrideStyle`.

Ein Style erstellen

So erstellen Sie in Servoy ein Style:

1. Führen Sie einen der folgenden Schritte aus:
 - a. Wählen Sie im Menü von Servoy Developer den Eintrag File -> New -> New Project. Es wird ein neues Fenster geöffnet. Markieren Sie den Knoten Servoy und wählen Sie New Style. Klicken Sie auf Next.
 - b. Klicken Sie in der oberen Toolbar auf die Schaltfläche New. Es wird ein neues Fenster geöffnet. Markieren Sie den Knoten Servoy und wählen Sie New Style. Klicken Sie auf Next.
 - c. Klicken Sie unterhalb des Knotens Resources auf den Knoten Styles. Wählen Sie im Menü Create new style. Es wird ein neues Fenster geöffnet.
 - d. Nachdem Sie den Knoten Styles markiert haben, klicken Sie mit der rechten Maustaste auf einen vorhandenen Style und wählen Sie im Kontextmenü den Eintrag Create New Style. Es wird ein neues Fenster geöffnet.
 - e. Nachdem Sie den Knoten Styles markiert haben, klicken Sie in der Toolbar der Listenansicht des Solution Explorers auf die Schaltfläche Create new style. Es wird ein neues Fenster geöffnet.
2. Geben Sie einen Namen für den neuen Style ein. Klicken Sie auf Next.
3. Wählen Sie, dass Sie den Beispieldinhalt hinzufügen wollen. Klicken Sie auf Next.
4. Die neue CSS-Datei erscheint in der Mitte Ihres Arbeitsbereichs in einer Editoransicht.

Tipp

Im Normalfall ist es einfacher, mit einem vorhandenen Style zu arbeiten. Den meisten Beispiel-Solutions ist ein Style zugewiesen und diese Styles werden importiert, wenn die Solution in Servoy Developer importiert wird. Sie können auch einen neuen Style erstellen und die verschiedenen Einträge in den Style kopieren.

Mit einem Style arbeiten

Mit den folgenden Arbeitsschritten öffnen Sie ein vorhandenes Style Sheet:

1. Klicken Sie unterhalb des Ressourcen-Projekts im Solution Explorer auf Styles und führen Sie einen der folgenden Schritte aus:
 - a. Markieren Sie den erforderlichen Style (den Sie in der Listenansicht des Solution Explorers finden) und klicken Sie in der Toolbar der Listenansicht des Solution Explorers auf die Schaltfläche Open style.
 - b. Klicken Sie doppelt auf den Style.
 - c. Klicken Sie mit der rechten Maustaste auf den Style und wählen Sie im Kontextmenü den Eintrag Open Style.
2. Der Style wird in einer Editoransicht in der Mitte Ihres Arbeitsbereichs geöffnet.

Die Struktur des Style Sheets

Ein Style Sheet enthält in Servoy die grundlegenden Style-Definitionen sowie die Style-Definitionsklassen.

In Servoy finden Sie die folgenden Style-Definitionen:

- form
- label
- button
- field
- check
- radio
- combobox
- tabpanel
- portal

Unterhalb dieser Style-Definitionen kann der Entwickler viele Style-Definitionsklassen erstellen. Als Beispiel könnte die Style-Definition `label` die Style-Definitionsklassen `label.title`, `label.small` und `label.bold` umfassen.

Jeder Definition und Definitions-Klasse können eine oder mehrere Eigenschaften zugewiesen werden. Eigenschaften, die innerhalb der Style-Definition angelegt wurden, werden an jede davon abgeleitete Style-Definitionsklasse vererbt (cascaded). Sehen Sie sich das folgende Beispiel an:

```
label
{
  color: #ffffff;
  border-style: solid;
  font: bold 10pt Verdana;
}

label.mytext
{
  color: red;
  vertical-align: middle;
  border-width: 1px 1px 1px 1px;
  border-color: #111111 #111111 #111111 #111111;
  margin: 2px 2px 2px 2px;
}
```

Beachten Sie, dass die Styles `border` und `font` in der Klasse `mytext` nicht geändert werden. Das bedeutet, dass ein Label, das den Style `mytext` verwendet, fett, in 10 Punkt Verdana dargestellt wird, da dies in der übergeordneten Style-Definition angegeben wurde. Die Farbe wurde in der Definitions-Klasse `mytext` überschrieben, so dass das Label rot dargestellt wird, nicht schwarz.

Styles in Formularen verwenden

Ein Style kann einem Formular auf zwei Arten hinzugefügt werden.

- Während der Erstellung des Formulars – Im Assistenten New Form finden Sie die Möglichkeit, einem Formular einen Style zuzuweisen. Detaillierte Informationen dazu finden Sie im Eintrag `Formulare erstellen`.
- In der Eigenschaften-Ansicht des Formulars – Ändern Sie die Eigenschaft `styleName`, so dass sie den Namen des Style-Sheets enthält, der verwendet werden soll.

Tipp

Wenn Sie Formulare erstellen, wird das Style Sheet, das Sie für das erste Formular verwendet haben, in weiteren Formularen automatisch vorgeschlagen. Wenn Sie im Assistenten kein Style Sheet angegeben haben und später mehreren Formularen Style Sheets zuweisen wollen, markieren Sie die Formulare im Solution Explorer und ändern Sie die Eigenschaft `Properties`. Damit ändern Sie die Eigenschaft auf allen Formularen, die Sie markiert haben.

Formularelementen Styles zuweisen

Jedes Element auf einem Formular, dem ein Style zugewiesen wurde, der eine Style-Definition für diesen Elementtypen enthält, übernimmt die Darstellung aus dem Eintrag der Style-Definition. Wenn ich als Beispiel den Style Sheet `test` verwende, übernimmt jedes Label, das auf dem Formular platziert wird, den Style `label` aus dem Style Sheet.

Um für ein gegebenes Element einen anderen als den Standard-Style zu verwenden, ändern Sie den Eintrag `styleClass` in der Eigenschaften-Ansicht für dieses Element. Dafür führen Sie die folgenden Schritte aus:

1. Markieren Sie das betreffende Element.
2. In der Eigenschaften-Ansicht suchen Sie die Eigenschaft `styleClass`.
3. Wenn Sie in das Feld klicken wird eine Dropdown-Liste aufgerufen, die alle für das Element verfügbaren Style-Klassen enthält. Diese Liste wird im Kontext aufgerufen, was bedeutet, dass Sie nur Style-Klassen für Labels sehen, wenn das Element ein Label ist, nur Feldklassen, wenn es sich um ein field handelt usw.
4. Markieren Sie in der Liste die gewünschte Klasse.
5. Wenn Sie die Standard-Style-Definition verwenden wollen, wählen Sie `DEFAULT`.

Eine Anwendung bereitstellen

Inhaltsverzeichnis

- Eine Solution exportieren
- Den Servoy Application Server installieren
- Eine Solution importieren
- Eine Solution im Client ausführen

Eine Solution aus Servoy Developer exportieren

Nachdem Sie Ihre Solution entwickelt haben und bereit sind, Ihre Solution den Kunden bereitzustellen, müssen Sie die Solution aus der Entwicklungsumgebung in die Bereitstellungsumgebung (den Application Server) verschieben. Der erste Schritt für die Bereitstellung besteht darin, Ihre Solution in eine Form zu bringen, die verschoben werden kann. Dafür exportieren Sie die Solution.

Eine Solution exportieren

Um eine Solution aus Servoy Developer zu exportieren führen Sie die folgenden Schritte aus:

1. Führen Sie einen der folgenden Schritte aus:
 - a. Wählen Sie im Menü File -> Export. Es wird ein neues Fenster aufgerufen. Wählen Sie in der Baumansicht den Eintrag Servoy -> Export Active Solution und klicken Sie auf die Schaltfläche Next.
 - b. Klicken Sie im Solution Explorer mit der rechten Maustaste auf die aktive Solution. Wählen Sie im Kontextmenü den Eintrag Export Solution. Es wird ein neues Fenster aufgerufen.
2. Wählen Sie einen Namen und einen Speicherort für die Export-Datei und klicken Sie anschließend auf Next.
3. Jetzt steht Ihnen eine Liste mit Optionen zur Verfügung. Für eine einfache Solution können Sie diese Optionen leer lassen. Zu den Optionen, die Ihnen zur Verfügung stehen, gehören:
 - a. Protect Solution with Password – Für den Import der Solution ist ein Passwort erforderlich. Dies ist ein zusätzlicher Schritt, um nicht autorisierten Personen den Zugriff auf Ihre Solution zu verwehren.
 - b. Export referenced modules – Ist diese Option aktiv, werden auch die Module exportiert, die der Solution zugeordnet wurden. Wenn Sie diese Option anklicken rufen Sie damit ein neues Fenster auf, in dem Sie wählen können, welche Module exportiert werden sollen.
 - c. Export all tables form referenced servers – Wenn diese Option aktiv ist, werden alle Tabellen aller verwendeten Server in die Solution exportiert. Ist die Option nicht aktiv, werden nur die Tabellen in die Solution exportiert, die aktuell verwendet werden.
 - d. Export solution sample data – Wenn diese Option aktiv ist, werden die Daten aus der aktuellen Verbindung für jede Tabelle exportiert. Sie können eine Höchstmenge der Datensätze angeben oder Sie können alle Datensätze exportieren.

Vorsicht

Ist die Datenmenge zu groß. ist es möglich, dass nicht alle Datensätze exportiert werden.

Tip

Auch wenn Sie die Beispieldaten exportiert haben, werden diese beim Import nicht in die Tabelle importiert, wenn die Zieltabelle bereits Daten enthält.

- e. Export i18n data – Exportiert i18n-Schlüssel, die in der Solution verwendet werden.
 - f. Export users – Exportiert die Servoy-Nutzer, die in der User- und Gruppen-Security angegeben wurden.
4. Klicken Sie auf Finish (wenn Sie die Option Export referenced modules aktiviert haben, müssen Sie die zu exportierenden Module auswählen und anschließend auf Finish klicken).
 5. Die .servoy-Datei erscheint am angegebenen Speicherort.

Installieren und Ausführen des Servoy Application Server

Überblick

Die Debug-Clients sind eine gute Möglichkeit, um während der Entwicklung zu sehen, wie die Solution ausgeführt wird. Aber um zu sehen, wie die Solution in einer Produktivumgebung ausgeführt wird, müssen Sie Ihre Solution auf einem Application Server bereitstellen. In diesem Eintrag behandeln wir die Installation und die Ausführung des Servoy Application Servers.

Den Server installieren

Der Deployment Guide enthält die Informationen, die Sie für die Installation des Application Servers benötigen. Vergleichen Sie den Eintrag [Installing the server](#).

Wenn Sie auf einer Maschine bereits Servoy Developer installiert haben, ist der Application Server bereits installiert, da alle Installationen von Servoy Developer auch den Application Server installieren.

Den Server ausführen

Der Deployment Guide enthält die Informationen, die Sie für die Verwaltung des Application Servers benötigen. Vergleichen Sie den Eintrag [Operating the server](#).

Sie können den Application Server auch aus Ihrer Developer-Installation heraus ausführen. Dafür führen Sie die folgenden Schritte aus:

Beachten Sie bitte

Diese Anweisungen gehen davon aus, dass der Datenbank-Server ausgeführt wird.

Unter Windows:

- Beenden Sie Servoy Developer.
- Klicken Sie auf Start. Wechseln Sie zum Menü Servoy (oder wie immer das Menü während der Installation genannt wurde) und wählen Sie dort **Servoy Server** **oder**
- Öffnen Sie ein Befehls-Fenster, navigieren Sie zum Verzeichnis `application_server` unterhalb des Installationsverzeichnisses von Servoy und führen Sie den folgenden Befehl aus:

```
servoy_server.bat
```

Für MacOS oder Linux:

- Beenden Sie Servoy Developer.
- Öffnen Sie ein Terminal, navigieren Sie zum Verzeichnis `application_server` unterhalb des Installationsverzeichnisses von Servoy und führen Sie den folgenden Befehl aus:

```
./servoy_server.sh
```

Eine Bemerkung zu den Datenbank-Servern

Der Servoy Application Server erfordert für seine Arbeit eine Datenbank. Die Verbindung `repository_server` ist erforderlich, ansonsten wird der Application Server nicht gestartet.

In einer Standard-Installation ist die Verbindung `repository_server` bereits auf die Datenbank konfiguriert, die von Servoy bereitgestellt wird. Wenn Servoy Developer gestartet wird, wird automatisch auch diese Datenbank geöffnet.

In Versionen von Servoy, die PostgreSQL verwenden, wird die Datenbank angehalten, wenn Servoy Developer geschlossen wird. Um die Datenbank über die Befehlszeile zu starten wechseln Sie zum Verzeichnis `application_server` und führen Sie den folgenden Befehl aus:

```
postgres_db\bin\pg_ctl start -D database -l postgres_db
\postgres_log.txt
```

Eine Solution in den Application Server importieren

Überblick

Um eine Solution zu importieren müssen Sie zunächst über einen laufenden Servoy Application Server verfügen. Vergleichen Sie dazu den Eintrag **Installieren und Ausführen des Servoy Application Server**. Nachdem wir einen Server haben, können wir das Server Administration Web Utility verwenden, um den Server zu verwalten. Eine der damit zusammenhängenden Aufgaben ist der Import von Solutions.

Auf das Application Server Administration Utility zugreifen

So greifen Sie auf das Application Server Administration Utility zu:

- Öffnen Sie einen Browser
- Geben Sie die folgende URL ein:

http://<your_application_server>:8080/servoy-admin

Ersetzen Sie <your_application_server> durch die Adresse Ihres Application Servers.

Die Solution importieren

Um eine Solution in das Administration Utility zu importieren führen Sie die folgenden Schritte aus:

1. Prüfen Sie, dass der Application Server nicht als ein Team-Repository gestartet wurde.
 - a. Gehen Sie zum Wert des `servoy.application_server.startRepositoryAsTeamProvider` nahe dem Fuß der Hauptseite des Administration Utility.
 - b. Steht der Wert auf True ändern Sie ihn auf False und speichern Sie die Einstellungen, indem Sie auf die Schaltfläche Save Settings klicken.
 - c. Starten Sie den Application Server (suchen Sie dafür nach einer Schaltfläche oben auf der Hauptseite).
2. Markieren Sie im Menü auf der linken Seite den Link Solutions.
3. Am Ende der Solution-Liste klicken Sie auf den Link Import a Solution. Die Seite Solution Import wird aufgerufen.
4. Geben Sie den Speicherort der .servoy-Datei ein. Wenn Sie keine .servoy-Datei haben, vergleichen Sie dazu den Eintrag Eine Solution aus Servoy Developer exportieren.
5. Für einen typischen Import ändern Sie die Standardeinstellungen nicht. Einige der häufiger verwendeten Einstellungen erläutern wir im Kasten am Ende dieses Eintrags.

6. Klicken Sie auf die Schaltfläche Import. Ein Fenster mit einer Zusammenfassung wird aufgerufen.

7. Am Ende des Imports zeigt dieses Fenster an, ob der Import erfolgreich war.

Um zu prüfen, ob Ihre Solution importiert wurde, klicken Sie erneut auf den Link Solutions, um sich die Liste der Solutions anzeigen zu lassen.

Einige wichtige Import-Optionen

Zu den häufiger verwendeten Import-Optionen gehören:

- Allow reserved SQL keywords – Diese Option ermöglicht es, für Namen innerhalb der Datenbank reservierte SQL-Schlüsselworte zu verwenden. Ist diese Option nicht aktiv, schlägt der Import fehl, sobald unabhängig von der Datenbank-Plattform ein reserviertes Wort auftaucht. Prüfen Sie Ihre DBMS-Dokumentation, welche Worte reserviert sind und daher Ihre Installation gefährden.
- Display Data model changes – Alle Änderungen, die an der Datenbank vorgenommen wurden, werden während des Importlaufs auf dem Status-Bildschirm angezeigt. Sind viele Änderungen vorhanden, kann dies eine längere Liste werden.
- Import solution sample data – Sind in der Exportdatei der Solution Daten gespeichert, wird versucht, sie in die Datenbank zu kopieren. Enthält die Tabelle in der Datenbank bereits Daten, werden die Beispieldaten nicht übernommen, unabhängig davon, ob Sie diese Option aktiviert haben oder nicht.
- i18n data – Sie können angeben, dass i18n-Daten, die in der Export-Datei gespeichert sind, importiert werden sollen. Außerdem können Sie angeben, dass nur neue Schlüssel importiert werden sollen, die in der aktuellen i18n-Datenbank nicht vorhanden sind.

Eine Solution im Client ausführen

Überblick

Um eine Solution im Client ausführen zu können muss sich die Solution in einem laufenden Application Server befinden.

Wie Sie einen Application Server starten erfahren Sie im Eintrag Installieren und Ausführen des Servoy Application Servers.

Wie Sie Ihre Solution in den Application Server laden erfahren Sie in den Einträgen „Eine Solution aus Servoy Developer exportieren“ und „Eine Solution in den Application Server importieren“.

Zugriff auf den Application Server

Der Zugriff auf den Application Server erfolgt über einen Browser. Die URL des Application Servers sieht wie folgt aus:

```
http://<application_server_address>:8080
```

Ersetzen Sie <application_server_adress> durch die aktuelle Adresse Ihres Servers. Wenn Sie auf einen lokal ausgeführten Application Server zugreifen wollen, verwenden Sie als Adresse für den Server einfach localhost.

Der Zugriff auf den Server durch eine Firewall

Wenn Sie durch eine Firewall auf Ihren Application Server zugreifen wollen, müssen Sie auf den Ports 800 und 1099 den TCP-Traffic zulassen, um sich korrekt verbinden zu können.

Ihre Solution ausführen

Ihre Solution kann vom Application Server wahlweise in einem Smart Client oder einem Web Client ausgeführt werden.

Einen Smart Client ausführen

Um eine Solution im Smart Client auszuführen, folgen Sie den folgenden Schritten:

1. Öffnen Sie einen Browser und geben Sie die URL des Servers zusammen mit der Nummer des Ports ein. Es wird eine Client Launch-Seite geöffnet.
2. Klicken Sie auf den Icon. Der Client Launch-Prozess wird gestartet.

Beachten Sie bitte

Ist die Java-Laufzeitumgebung auf dem Rechner nicht installiert, können Sie die durch einen Klick auf den Link am Fuß der Launch-Seite installieren. Dieser Link prüft auch, ob die Laufzeitumgebung von Java bereits installiert wurde.

3. Wenn Sie danach gefragt werden, wählen Sie die Solution, die Sie öffnen wollen.
4. Die Solution wird anschließend im Smart Client geöffnet.

Einen Web Client ausführen

Um eine Solution im Web Client auszuführen, folgen Sie den folgenden Schritten:

1. Öffnen Sie einen Browser und geben Sie die folgende URL ein:

```
http://<application_server_address>:8080/servoy-webclient
```

Ersetzen Sie dabei <your_application_server> durch die Adresse Ihres Application Servers.

2. Ein Menü zeigt Ihnen alle zur Verfügung stehenden Solutions an.
3. Wählen Sie die Solution, die Sie öffnen wollen.
4. Die Solution wird im Browserfenster geöffnet.

Weitere Möglichkeiten

Sie können Ihre Solutions auch auf andere Weise öffnen. Vergleichen Sie dazu die Einträge Smart Client und Web Client im Deployment Guide.