



Vystavění klienta Rich Java Client, který
používá webovou službu

Obsah

Vystavění klienta Rich Java Client, který používá webovou službu 1

Úvod: Vystavění klienta Rich Java Client, který používá webovou službu	1
Modul 1: Návrh GUI klienta ve vizuálním editoru.	3
Lekce 1.1: Vytvoření projektu Java	3
Lekce 1.2: Přidání a rozvržení tabulky zaměstnanců	4
Lekce 1.3: Spuštění vizuální třídy	8
Modul 2: Svázání vizuálních komponent s webovou službou	10
Lekce 2.1: Instalace a implementace webové služby.	10
Lekce 2.2: Svázání tabulky zaměstnanců se zdrojem dat webové služby	12

Lekce 2.3: Svázání polí s podrobnostmi s výběrem tabulky	17
Lekce 2.4: Svázání tlačítka Aktualizovat s vázacím programem akce	21
Lekce 2.5: Zpřístupnění tlačítka Odstranit a dialogového okna Potvrdit odstranění	23
Lekce 2.6: Nastavení akcí a vazeb pro přidání nového zaměstnance	25
Lekce 2.7: Programování chování tlačítka Zrušit	30
Lekce 2.8: Nastavení filtru na tabulce zaměstnanců	31
Souhrn: Vystavění klienta Rich Java Client, který používá webovou službu	32

Vystavění klienta Rich Java Client, který používá webovou službu

Tento výukový program vás naučí, jak používat vizuální editor Javy pro vystavění klienta Rich Java Client, který se připojuje k webové službě. Klient, kterého vystavíte v tomto výukovém programu, se jmenuje My Company Directory.

My Company Directory je aplikace Javy, která se používá pro uchování adresáře zaměstnanců společnosti. Aplikace se připojuje k ukázkové webové službě, která poskytuje metody pro vytváření, načítání, aktualizaci a výmaz záznamů zaměstnanců.

Klient se vizuálně sestavuje ve vizuálním editoru Javy pomocí komponent Swing. Vizuální editor Javy poskytuje sadu pomocných tříd (zdroje dat, datové objekty a vázací programy) nebo připojení a práci s webovou službou. Webová služba se nasazuje lokálně na vaší vlastní instalaci serveru IBM WebSphere Application Server v6.0 a nástroje vám pomohou generovat proxy prostředí Java pro vašeho klienta na základě souboru WSDL (Web Services Description Language).

Prohlédnout dokončený produkt

Cíle výuky

V tomto výukovém programu se naučíte následující:

- jak používat vizuální editor Javy pro návrh a rozvržení uživatelského rozhraní
- jak svázat prvky rozhraní s datovými objekty a webovou službou

Požadovaný čas

2 hodiny a 15 minut

Související informace

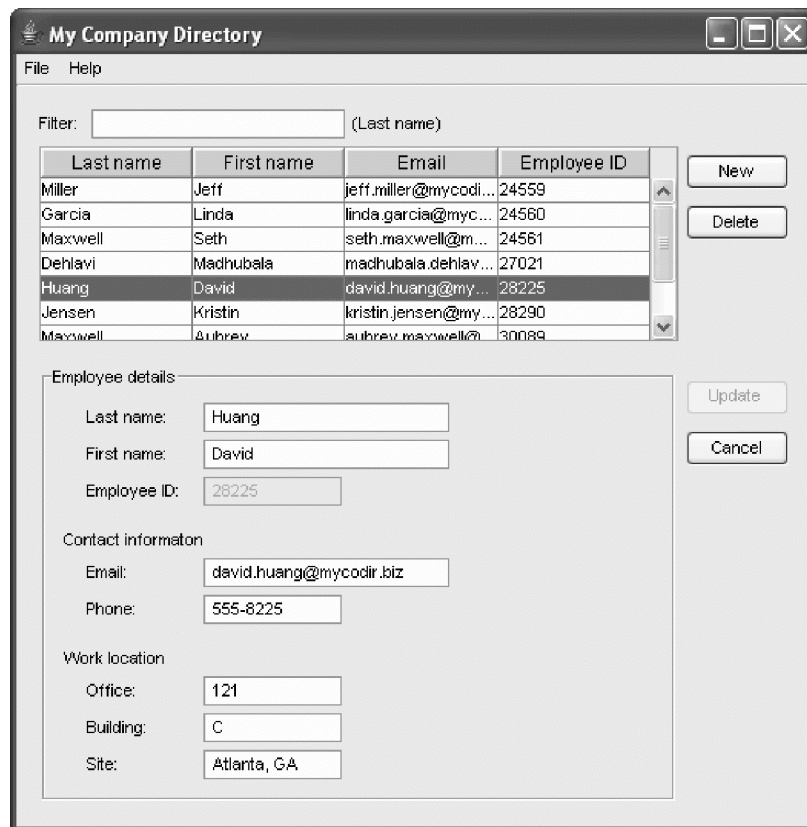


Zobrazit PDF verzi

Výukový program: Ahoj světe Javy

Úvod: Vystavění klienta Rich Java Client, který používá webovou službu

Grafické uživatelské rozhraní (GUI) pro tohoto klienta pro vás bylo z převážné části předem vystavěno pomocí komponent Swing. V tomto prvním modulu dokončíte rozvržení základních komponent GUI pomocí vizuálního editoru Javy. Ve druhém modulu svážete komponenty GUI se zdrojem dat, službami a objekty webové služby vrácenými zdrojem dat. Ve vizuálním editoru Javy budete používat zdroje dat, datové objekty a vázací programy, což jsou instance pomocných tříd, které Vizuální editor Javy generuje a vaše aplikace používá.



Viz obrázek dokončeného produktu:

Cíle výuky

V tomto výukovém programu se naučíte následující:

- jak používat vizuální editor Javy pro návrh a rozvržení uživatelského rozhraní
- jak svázat prvky rozhraní s datovými objekty a webovou službou

Požadovaný čas

Pro dokončení celého výukového programu budete potřebovat přibližně 2 hodiny a 30 minut.

Systémové požadavky

- WebSphere Application Server v6.1. Tento server je již možná nainstalován spolu s produktem nebo jej můžete nainstalovat vlastní samostatnou instalací. Scénář tohoto výukového programu po vás požaduje implementovat ukázkovou webovou službu na serveru WebSphere Application Server, který běží lokálně.

Ukázková webová služba může běžet na jiných serverech, ale tento výukový program byl testován na pouze serverech WebSphere Application Server v6.0 a v6.1.

Požadavky

Měli byste znát následující koncepty:

- základní vývoj Javy
- základní principy webové služby
- základní znalosti pracovního prostředí, jako je práce s projekty a navigace v perspektivách a pohledech

Modul 1: Návrh GUI klienta ve vizuálním editoru

V tomto modulu se naučíte, jak používat vizuální editor Javy pro přidání vizuální komponenty do aplikace, vizuální rozvržení a nastavení omezení uspořádání. Závěrečná lekce tohoto modulu vám ukáže, jak spustit soubor Javy, abyste si prohlédli, jak bude vypadat jako skutečná aplikace.

Zapamatujte si: Než začnete tento modul, měli byste mít požadované znalosti uvedené v úvodu výukového programu.

Cíle výuky

Po dokončení lekcí v tomto modulu budete rozumět konceptům a vědět, jak provést následující:

- vložit a rozvrhnout tabulku v rozhraní Javy
- spustit vizuální třídu a otestovat svou práci

Požadovaný čas

Dokončení tohoto modulu zabere přibližně 15 minut.

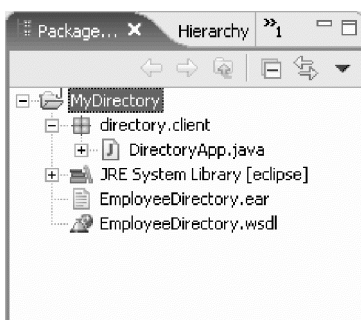
Lekce 1.1: Vytvoření projektu Java

V této lekci vytvoříte projekt MyDirectory jeho naimportováním do pracovního prostoru. Projekt zahrnuje jak jednoduchou třídu Java, tak další soubory, které se použijí později.

Jelikož hlavním smyslem tohoto výukového programu je navázání vizuálních komponent na webovou službu, je již většina rozhraní GUI Java pro aplikaci "My Company Directory" navržena.

Projekt MyDirectory je hlavní projekt Javy, se kterým budete v tomto výukovém programu pracovat. Obsahuje soubor DirectoryApp.java, který obsahuje hlavní aplikaci Javy, kterou sestavujete. Tento výukový program zahrnuje několik verzí projektu MyDirectory, které by vám mohly pomoci: jednu pro spuštění každého modulu a dokončenou verzi úplného projektu.

1. Import projektu MyDirectory.
2. V Průzkumníku balíčků v perspektivě Javy se ujistěte, že projekt MyDirectory vypadá jako následující obrázek:



Kontrolní bod lekce

V této lekci jste importovali vzorový projekt MyDirectory, což je výchozí bod tohoto výukového programu.

Projekt MyDirectory zahrnuje následující prostředky:

- DirectoryApp.java: Soubor Javy, který obsahuje aplikaci, kterou v tomto výukovém programu rozvíjíte. Soubor DirectoryApp.java je v balíčku Java se jménem directory.client.
- EmployeeDirectory.ear: Podniková aplikace, která obsahuje ukázkovou webovou službu. V modulu 2 budete nasazovat tuto webovou službu na lokální instalaci serveru WebSphere Application Server v6.0.
- EmployeeDirectory.wsdl: Soubor XML, který používá WSDL (Web Services Description Language) pro popis ukázkové webové služby, kterou budete nasazovat. V modulu 2 budete používat tento soubor WSDL pro generování proxy prostředí Java, který bude vaše aplikace používat.

Lekce 1.2: Přidání a rozvržení tabulky zaměstnanců

V této lekci budete používat vizuální editor Javy pro přidání podokna JScrollPane a tabulky JTable do aplikace. V dalších cvičeních budete programovat tabulku JTable, aby získala data od webové služby, která vrací seznam všech zaměstnanců v adresáři společnosti.

Poté, co přidáte tabulku JTable, budete používat pohled návrhu vizuálního editoru Javy pro přizpůsobení rozvržení JTable, aby odpovídalo následujícím specifikacím:

- Roztáhnete tabulku JTable přes tři buňky horizontálně a přes dvě buňky vertikálně.
- Přidáte vlevo vsazení 15 pixelů.
- Přejmenujete JTable na employeesTable.

Zobrazit

Otevření souboru DirectoryApp.java ve vizuálním editoru Javy

Otevření souboru DirectoryApp.java ve vizuálním editoru Javy:

1. V pohledu Průzkumník balíčků perspektivy Java rozbalte projekt MyDirectory a balíček directory.client.
2. Klepněte pravým tlačítkem myši na soubor DirectoryApp.java a vyberte **Otevřít pomocí → Vizuální editor**. Vizuální editor Javy nahraje třídu Javy a zobrazí návrh v oblasti grafického plátna.

Tip:

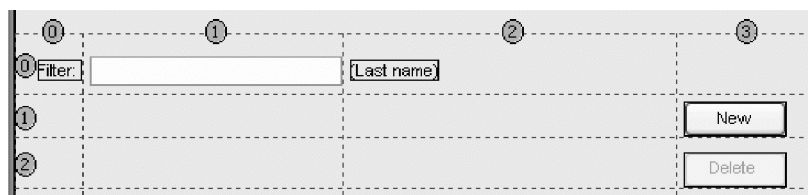
- Abyste změnili vzhled a chování použité vizuálním editorem Javy, přejděte na **Okno → Preference → Java → Vizuální editor** a uveďte vzhled a chování Swing. Preference se projeví při dalším otevření třídy. Tento výukový program používá standardní vzhled a chování Windows.
- Aby se vizuální editor stal předvoleným editorem pro všechny soubory Javy, můžete klepnout na **Okno → Preference** a přejít na stránku **Pracovní prostředí → Přidružení souboru**, kde můžete definovat své preference.

Přidání tabulky JTable na podokno JScrollPane

Hlavní okno DirectoryApp.java používá JFrame s JPanel pro hlavní panel. JPanel se v naší aplikaci jmenuje jContentPane. jContentPane byl nastaven, aby používal typ správce rozvržení GridBagLayout. GridBagLayout je účinné schéma rozvržení založené na mřížce buněk, které lze obsadit vizuálními komponentami. Vizuální editor Javy usnadňuje práci s GridBagLayout tím, že zobrazuje hranice mřížky. Také zobrazuje značky umístění, když uvolníte nové komponenty na mřížku, a zobrazuje odkazy na komponenty, které na GridBagLayout přesunujete nebo měníte jejich velikost.

Přidání tabulky zaměstnanců (javax.swing.JTable) do uživatelského rozhraní DirectoryApp.java:

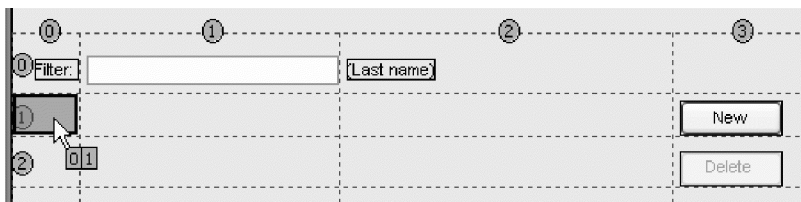
1. Klepněte pravým tlačítkem myši na podokno jContentPane v pohledu Java Beans a vyberte **Zobrazit mřížku**. Červená tečkovaná čára zobrazuje hranici mřížky a modré kroužky s čísly indikují čísla řádků a sloupců. Například si všimněte, že tlačítko **Nový** je v řádku 1 (mřížka y) a sloupci 3 (mřížka x).



2. V obvyklé paletě vizuálního editoru Javy vyberte komponentu Swing **JTable na JScrollPane** , která je v zásuvce palety **Komponenty Swing**.

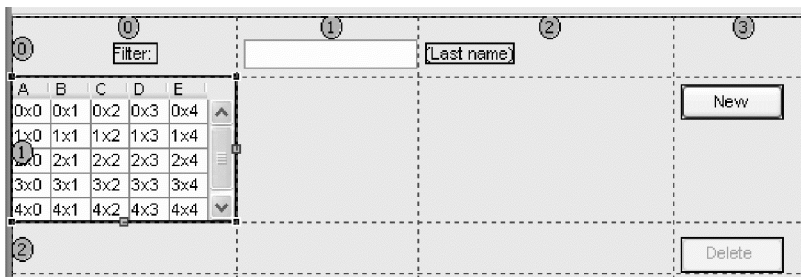
Tip: Podle předvolby je paleta sbalená na pravé straně oblasti návrhu. Můžete změnit velikost palety anebo ji přemístit.

3. Přesuňte ukazatel myši nad buňku v mřížce ve sloupci 0, řádku 1:



- Když pohybuje ukazatelem myši po mřížce, u ukazatele myši se zobrazují dva očíslované čtverečky, které udávají souřadnice x a y v mřížce v závislosti na pozici ukazatele myši.
- Pokud držíte ukazatel myši přesně nad hranicí mřížky, je možné vytvořit nové řádky a sloupce a přechíslovat existující řádky a sloupce. V tomto případě indikují toto chování žluté čtverečky na ukazateli myši, žluté pruhy v mřížce a žluté popisky sloupce a řádku - ukazují vliv, který bude umístění mít.

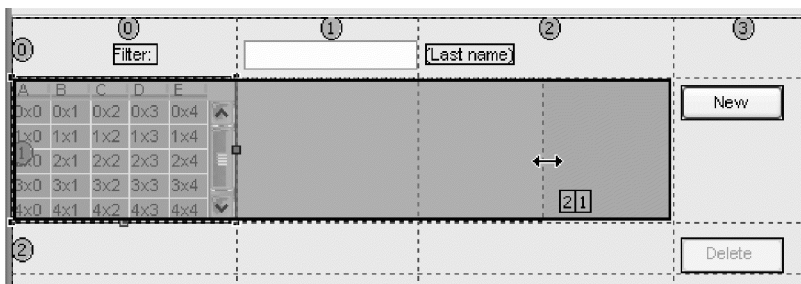
4. Klepnutím na levé tlačítko umístíte JScrollPane a JTable na buňku na sloupci 0 a řádku 1:



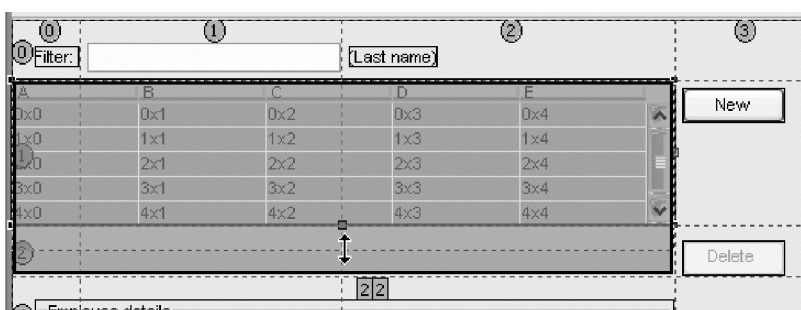
Roztáhněte JScrollPane a JTable přes několik sloupců a řádek mřížky

Nyní je potřeba roztáhnout podokno JScrollPane (a podrízenou tabulku JTable) na tři sloupce a dva řádky, aby se zajistilo lepší chování při změně velikosti. Abyste zvětšili rozsah tabulky na požadované sloupce a řádky:

1. Vyberte podokno JScrollPane v oblasti návrhu nebo v pohledu Java Beans (měl by stále být vybrán, protože jste ho právě přidali). Všimněte si malých zelených čtverečků vpravo a dole od JScrollPane. Tyto manipulátory na úpravu velikosti použijete pro tažení JScrollPane, aby obsáhl několik sloupců a řádků.
2. Klepněte a držte levé tlačítko myši nad zeleným manipulátorem napravo od JScrollPane.
3. Táhněte ukazatel myši doprava, dokud umístění nebude udávat sloupec 2, řádek 1. Tmavě šedé stínování bude také určovat buňky, které bude komponenta okupovat, až uvolníte tlačítko myši.



4. Uvolníte tlačítko myši. JScrollPane nyní obsahuje tři sloupce.
5. Zopakujte podobný proces a táhněte dolní manipulátor JScrollPane, dokud JScrollPane nebude zasahovat do řádku 2:



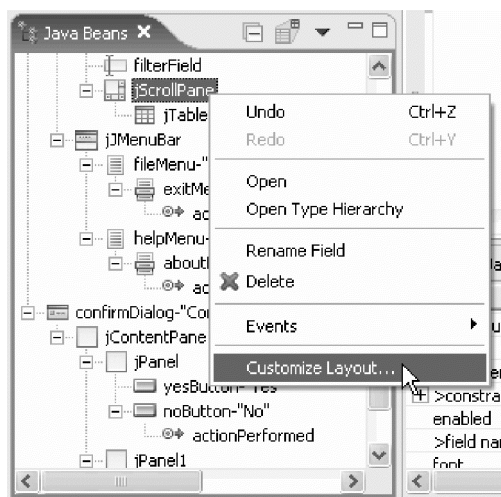
Přizpůsobte rozestup podokna JScrollPane v rámci GridBag

Další funkcí správce GridBagLayout je to, že můžete uvést různá omezení pro další úpravy rozvržení. Například, můžete uvést následující omezení:

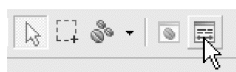
- **Ukotvení:** komponentě lze přiřadit orientaci ukotvení v rámci příslušné buňky, což posléze ovlivní, jak se komponenta bude pohybovat při změně velikosti aplikace. Například lze komponentu ukotvit vlevo nahoře, vlevo uprostřed, uprostřed nebo vpravo dole.
- **Výplň:** komponenta může dostupný prostor v rámci své buňky nebo buněk okupovat buď horizontálně nebo vertikálně nebo obojí.
- **Vsazení:** komponentě lze přiřadit vlastní vyplnění nahoře, dole, vlevo a vpravo, abyste zajistili rozestupy mezi komponentou a hranicí mřížky.

Přizpůsobení ukotvení, výplně a vsazení pro podokno JScrollPane:

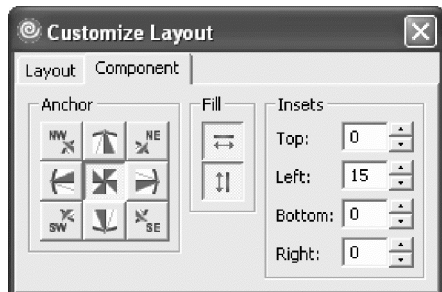
1. Klepněte pravým tlačítkem na podokno JScrollPane v pohledu návrhu nebo v pohledu Java Beans a vyberte **Přizpůsobit rozvržení**.



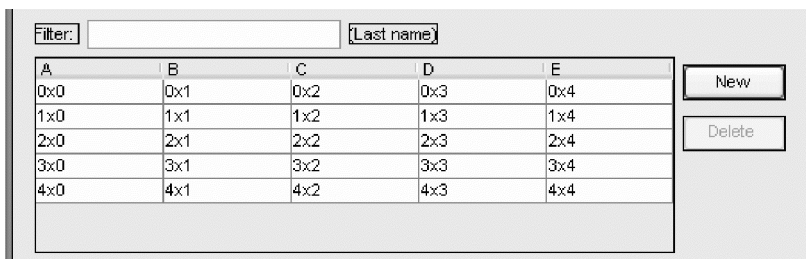
Tip: Dialogové okno Přizpůsobit rozvržení může zůstat otevřené, když vybíráte a měníte rozvržení pro různé komponenty. Dialogové okno Přizpůsobit rozvržení můžete kdykoliv otevřít klepnutím na tlačítko Přizpůsobit rozvržení v pruhu nabídky:



2. Na kartě Komponenta v dialogovém okně Přizpůsobit rozvržení se ujistěte, že je stisknuté tlačítko Ukotvit střed.
3. Ujistěte se, že jsou stisknutá tlačítka **Vyplnit horizontálně** a **Vyplnit vertikálně**.
4. Přidejte vsazení vlevo (15 pixelů), aby byl rozestup na levé straně podokna JScrollPane podobný jako mají další vizuální komponenty aplikace.



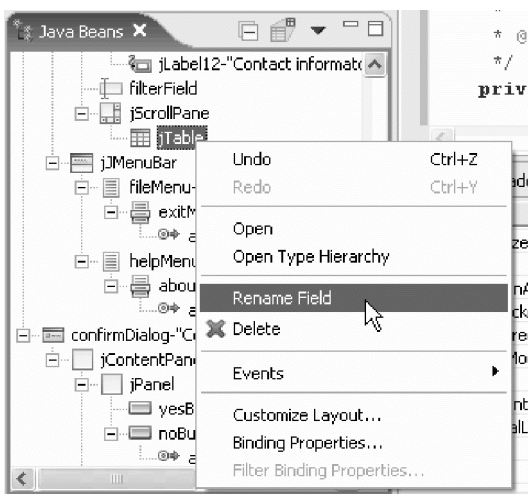
Tabulka je nyní zarovnána například podle popisu **Filtr**.



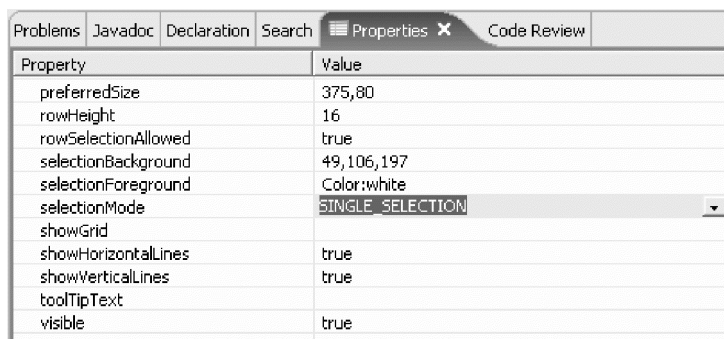
Přejmenujte tabulku JTable na užitečnou hodnotu a nastavte ji, aby vybírala pouze jediný řádek

Protože budete později s tabulkou pracovat, bude pro vás užitečné přejmenovat instanci JTable a její metodu získávání. Přejmenování tabulky:

1. V pohledu Java Beans klepněte pravým tlačítkem na komponentu jTable a vyberte **Přejmenovat pole** z rozevírací nabídky.



2. Zapište `employeesTable` a klepněte na **OK**. Tabulka JTable se nyní jmenuje `employeesTable` a metoda pro vytvoření instance je `getEmployeesTable`.
3. Nastavte tabulku tak, aby povolovala pouze výběr jediného řádku.
 - a. V pohledu návrhu vyberte tabulku `employeesTable`.
 - b. V pohledu Vlastnosti vyberte vlastnost **selectionMode** a nastavte ji na `SINGLE_SELECTION`.



- c. Uložte soubor `DirectoryApp.java`.

Kontrolní bod lekce

V této lekci jste se naučili, jak používat vizuální editor pro přidání tabulky do existujícího uživatelského rozhraní. Také jste se naučili, jak přizpůsobit rozvržení, umístění a rozestupy.

Lekce 1.3: Spuštění vizuální třídy

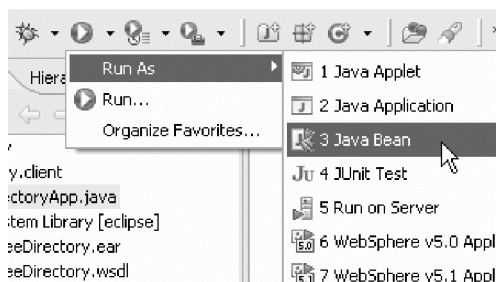
Nyní jste připraveni spustit aplikaci Javy a prohlédnout si její vzhled. Pracovní prostředí a vizuální editor umožňují rychle spustit aplikaci, a tyto kroky můžete ve svém prostředí provést kdykoliv, abyste otestovali skutečný běhový vzhled a chování třídy.

Zobrazit

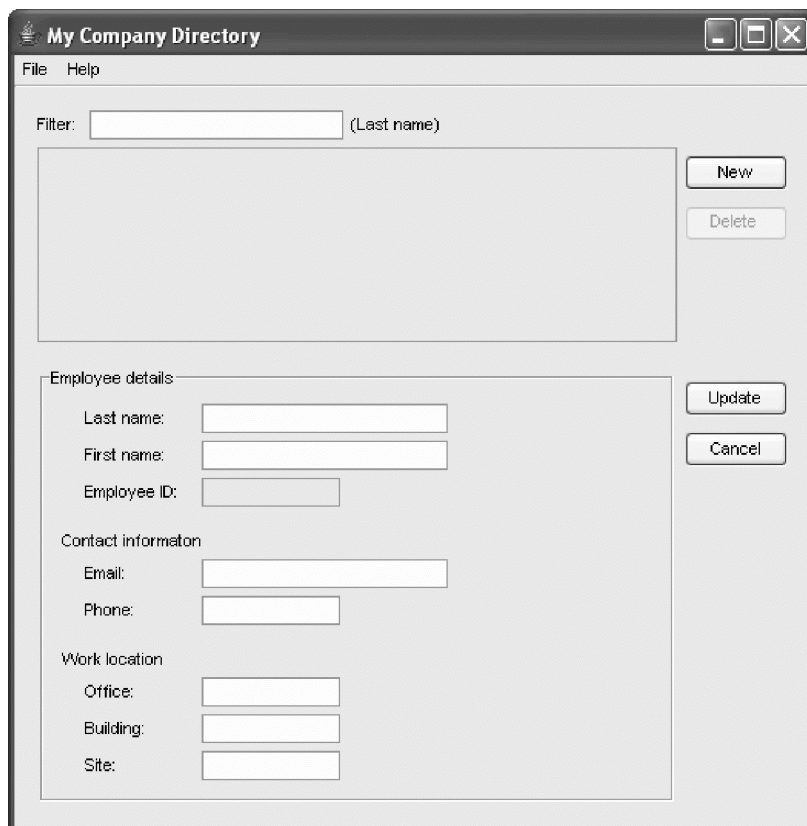
Vizuální editor Javy poskytuje spouštěč Java Bean, který je schopný spustit třídu bez metody main(). Když spouští vizuální třídu, spouští aplikaci v odděleném systému VM. Pokud spouštíte vizuální třídu aplikace Java, spouštěč se pokusí provést metodu main() v této třídě. V tomto výukovém programu zahrnuje vaše aplikace metodu main(), která vyvolává a zobrazuje rámec DirectoryApp JFrame, takže ji můžete spustit jako aplikaci nebo jako objekt bean Javy.

Spuštění souboru DirectoryApp.java jako objekt bean Javy:

1. Ujistěte se, že váš soubor DirectoryApp.java je otevřený ve vizuálním editoru Javy.
2. V pruhu nabídky klepněte na **Spustit → Spustit jako → Java Bean**.



Tip: Aplikace se otevře na vaší pracovní ploše ve vzhledu a chování Swing, které jste definovali v preferencích Vizuální editor Javy (**Okno → Preference → Java → Vizuální editor**). Případně můžete klepnout na **Spustit → Spustit** a definovat vzhled a chování pro příslušnou konfiguraci spuštění tohoto objektu Java bean. Pokud tuto aplikaci spouštíte jako aplikaci a ne jako objekt bean, bude také používat vzhled a chování Windows, protože je definované v metodě main(). Snímky obrazovky používané v tomto výukovém programu používají vzhled a chování Windows.



Kontrolní bod lekce

Protože jste pouze navrhli rozhraní, ale nenaprogramovali libovolné připojení dat nebo funkčnost událostí, nemůžete s aplikací nic dělat. Nicméně, můžete si prohlédnout základní rozvržení a vzhled, tak jak ho uvidí uživatel. Můžete zkusit klepnout na některá z tlačítek, ale zjistíte, že nic nedělají. Nicméně nabídky Soubor a Nápověda jsou již pro vás implementovány. Můžete si je vyzkoušet a prozkoumat kód Javy, abyste zjistili, jak se implementují s událostmi `actionPerformed`.

Probrané lekce

Tento modul vás seznámil s návrhem rozhraní pro klienta Rich pomocí vizuálního editoru Javy. Vedle návrhu vzhledu klienta musíte nicméně provést řadu dalších kroků, aby byl klient užitečný. Zejména budete muset zahrnout chování událostí a další logiku, a v tomto případě i vazby vizuálních prvků s některým typem zdroje dat.

V tomto modulu jste se naučili provést následující úlohy:

- naimportovat projekt Javy pomocí importu Project Interchange
- přidat tabulku `JTable` na podokně `JScrollPane` do vaší vizuální třídy
- používat správce `GridBagLayout` pro vizuální rozvržení tabulky na klientovi Rich
- spustit aplikaci, abyste si prohlédli skutečný vzhled klienta Rich Java Client

V dalším modulu, Modulu 2, svážete vizuální komponenty s webovou službou; vezmete jednoduché rozhraní My Company Directory a přeměníte ho ve výkonného klienta Rich, který používá metody webové služby pro vytvoření, načítání, aktualizaci a výmaz záznamů zaměstnanců z adresáře společnosti.

Modul 2: Svázání vizuálních komponent s webovou službou

Tento modul vás naučí, jak svázat vizuální prvky aplikace My Company Directory (tlačítka, tabulku zaměstnanců, pole a další akce) s webovou službou. Webová služba poskytuje skutečnou funkčnost pro vytváření, načítání, aktualizaci a výmaz zaměstnanců z ukázkového adresáře.

Cíle výuky

Po dokončení lekcí v tomto modulu budete rozumět konceptům a vědět, jak provést následující:

- svázat tabulku se zdrojem dat datové webové služby
- svázat pole s objekty
- programovat tlačítka s akcemi

Dokončení tohoto modulu zabere přibližně **2 hodiny**.

Lekce 2.1: Instalace a implementace webové služby

V tomto cvičení budete instalovat soubor ukázkové podnikové aplikace (EAR) do serveru WebSphere Application Server v6.1 a implementovat webovou službu EmployeeDirectory. Vaše aplikace bude používat webovou službu pro vytvoření, čtení, aktualizaci a vymazání záznamů o zaměstnancích.

Než začnete, musíte provést *jednu z následujících možností*, abyste se ujistili, že je váš projekt MyDirectory v řádném výchozím bodě:

- Proveďte “Modul 1: Návrh GUI klienta ve vizuálním editoru” na stránce 3.
nebo
- Import projektu MyDirectory na počátečním bodu Modulu 2

Tip: Pokud neuvedete jiný název projektu během importu, tato akce přepíše obsah projektu MyDirectory.

Váš projekt Java MyDirectory zahrnuje soubor EmployeeDirectory.ear. Použijete administrativní konzoli WebSphere pro instalaci podnikové aplikace EmployeeDirectory, která je obsažena v souboru EAR. Až nainstalujete aplikaci, implementujete také webovou službu obsaženou v této aplikaci. Dokončená aplikace My Company Directory používá tuto implementovanou webovou službu.

Instalace ukázkové aplikace EmployeeDirectory a implementace webové služby v prostředí serveru WebSphere Application Server v6.1:

1. Spusíte instanci aplikačního serveru z pracovního prostředí. Je několik různých způsobů, jak spustit server, ale tyto kroky popisují, jak to provést z pracovního prostředí:
 - a. Otevřete pohled Servery. Pro přidání pohledu Servery do perspektivy Java klepněte na **Okno → Zobrazit pohled → Další a vyberte Server → Servery**.
 - b. Pohled Servery uvádí servery, které jsou instalované a nastavené.
 - c. Klepněte pravým tlačítkem myši na server a vyberte **Spustit**. Když pohled Servery zobrazuje stav serveru jako **Spuštěný** nebo konzole uvádí **Server server1 open for e-business**, je server již spuštěn. Nyní můžete spustit administrativní konzoli.

Poznámka: Pokud v pohledu Servery není žádná instance serveru, vytvořte nový server:

- a. Klepněte pravým tlačítkem na pohled Servery a vyberte **Nový → Server**.
 - b. Pomocí průvodce Nový server přidejte server WebSphere Application Server v6.1.
2. Spusíte administrativní konzoli WebSphere. Opět je několik dalších způsobů, jak spustit administrativní konzoli, ale tyto instrukce popisují jak to provést z pracovního prostředí:
 - a. V pohledu Servery klepněte pravým tlačítkem na server, který jste právě spustili, a vyberte **Spustit administrativní konzoli**. Administrativní konzole WebSphere se otevře v okně prohlížeče.

- b. Zadejte uživatelské jméno a klepněte na **Přihlásit**. Otevře se uvítací stránka administrativní konzole. Uživatelské jméno, které zadáváte, se používá pouze pro trasování změn konfiguračních dat serveru specifických pro uživatele.
3. Použijte administrativní konzoli pro instalaci podnikové aplikace EmployeeDirectory.ear, která se nachází v projektu MyDirectory. Administrativní konzole používá přístup jako průvodce, aby vám usnadnila instalaci aplikací, kde se klepnutím na **Další** přesouváte ze stránky na stránku, dokud nejsou nastaveny všechny volby. Instalace ukázkové podnikové aplikace, která obsahuje webovou službu pro tento výukový program:
 - a. Na levé straně administrativní konzole rozbalte volbu **nabídky Aplikace** a klepněte na **Instalovat novou aplikaci**.
 - b. Vyberte **Lokální systém souborů** a do pole **Uveďte cestu** zadejte úplnou cestu do souboru EmployeeDirectory.ear, který je ve vašem projektu MyDirectory. Tip: Abyste získali úplnou cestu, klepněte pravým tlačítkem na soubor EmployeeDirectory.ear v Průzkumníku a vyberte **Vlastnosti**. Stránka Vlastnosti uvádí umístění souboru, které můžete zkopírovat a vložit do pole **Uveďte cestu**.
 - c. Klepejte na **Další**, dokud se nedostanete na stránku **Vybrat volby instalace**.
 - d. Vyberte **Implementovat webové služby**.
 - e. Klepejte na **Další**, dokud se nedostanete na stránku **Souhrn**, pak klepněte na **Dokončit**.
 - f. Když jste vyzváni k použití změn, které jste provedli v lokální konfiguraci, klepněte na odkaz **Uložit do hlavní konfigurace**. Projděte si změny a klepněte na tlačítko **Uložit**.
4. Použijte administrativní konzoli pro spuštění aplikace EmployeeDirectory:
 - a. Klepněte na **Aplikace** → **Podnikové aplikace**. Aplikace EmployeeDirectory je uvedena na seznamu jako aplikace instalovaná na serveru, ale její stav je Zastaveno.

Start Stop Install Uninstall Update Rollout Update Remove File Export Export DDL		
Select	Name ↕	Status ↕
☐	DefaultApplication	⇒
☒	EmployeeDirectory	✖
☐	Query	⇒
☐	SchedulerCalendars	⇒
☐	filetransfer	⇒
☐	ivtApp	⇒
Total 6		

- b. Vyberte zaškrtnávací políčko vedle EmployeeDirectory a klepněte na **Spustit**. Zpráva udává, že aplikace EmployeeDirectory byla úspěšně spuštěna a stavová ikona se změní na zelenou šipku.

Aplikace EmployeeDirectory nyní běží na lokálním hostiteli na portu 9080 a je možný přístup k webové službě. Po dokončení tohoto výukového programu můžete přejít zpět do administrativní konzole, ukončit aplikaci EmployeeDirectory a pak ji odinstalovat.

Pokud otevřete soubor EmployeeDirectory.wsdl ve vašem projektu MyDirectory (podle předvolby by se měl otevřít v grafickém editoru WSDL), můžete prozkoumat webovou službu, kterou jste právě implementovali. Pokud se soubor WSDL neotevře v editoru WSDL, schopnost Web Service Developer možná není v pracovním prostředí zapnutá. Schopnosti pracovního prostředí můžete uvést v Preferencích (**Okno** → **Preference** → **Pracovní prostředí** → **Schopnosti**).

Následující obraz z editoru WSDL zobrazuje operace dostupné ve službě EmployeeDirectory:



Editor WSDL můžete použít pro prozkoumání každé operace a jejích odpovídajících zpráv vzniklých při zpracování požadavku a návratových zpráv. To vám může pomoci porozumět webové službě a tomu, jak se používá v následujících cvičeních.

Lekce 2.2: Svázání tabulky zaměstnanců se zdrojem dat webové služby

Aplikace My Company Directory zobrazuje seznam všech aktuálních záznamů zaměstnanců v adresáři. Záznamy se zobrazují v tabulce JTable (employeesTable) s tříditelnými sloupci, které zahrnují příjmení, jméno, email a ID zaměstnance. Abyste získali záznamy pro tabulku, musíte svázat employeesTable s datovým objektem, který vrátí ukázka zdroje dat webové služby.

Zobrazit

Přehled datových objektů, zdrojů dat a vazacích programů

Abyste získali lokální datový objekt employeesTable, se kterým můžete pracovat, použijte vizuální editor pro přidání zdroje dat do vaší aplikace. Zdroj dat se připojí k serveru proxy ukázkové webové služby a objeví metodu webové služby, která je dostupná pro vaši aplikaci. Potom vyberete metodu služby getLightEmployeeRecord, která je k dispozici ze zdroje dat. Nakonec svážete tabulku employeesTable ve vaší aplikaci s poli, která se vrátí v objektu dat řádku (lightEmployeeRecordRows).

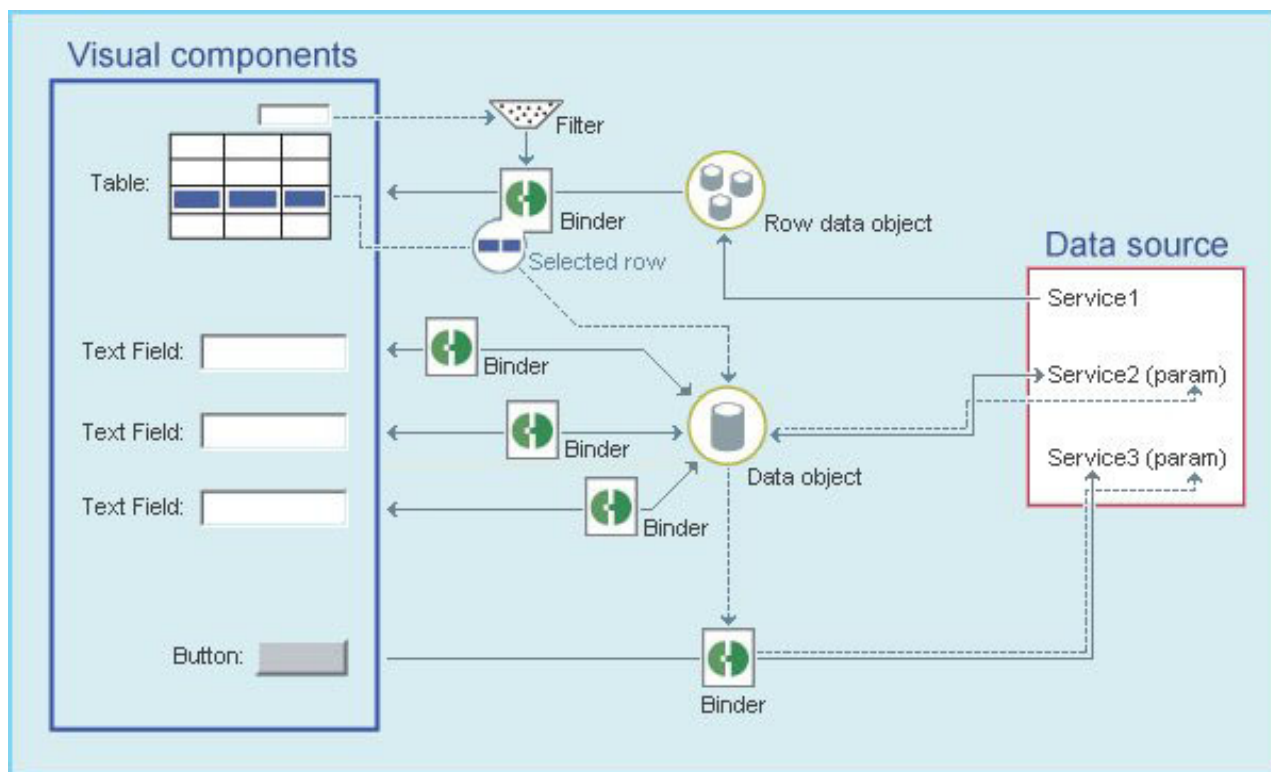
Můžete vytvořit všechny tyto zdroje dat a datové objekty rychle a snadno pomocí tříd vazacího programu vestavěných do vizuálního editoru Javy. Vizuální editor poskytuje sadu obecných rozhraní a tříd, které se generují do vašeho projektu, když svazujete vizuální komponenty s továrnami dat. Třídy vazacího programu se generují podle předvolby do balíčku jménem jve.generated. Vizuální editor poskytuje třídy vazacího programu jako generickou implementaci, kterou můžete dále upravovat a rozšiřovat, aby vyhovovala potřebám vaší aplikace. Tento výukový program ukazuje schopnosti a flexibilitu základního a snadného použití předvolených tříd vazacího programu.

Důležité: Než začnete toto cvičení, velmi doporučujeme, abyste si přečetli následující témata nápovědy. Tato témata vám mohou pomoci naučit se více o fungování a logice za datovými objekty, zdroji dat a vazacími programy, které vizuální editor Javy poskytuje:

- Vazací programy dat - přehled
- Odkaz rozhraní API vazacího programu

Pro tento výukový program budete používat zdroj dat webové služby, různé typy datových objektů a různé typy vazacích programů ve vaší aplikaci. Když přidáváte instance těchto objektů do aplikace, vizuální editor přidá potřebné třídy do balíčku jve.generated ve vašem projektu, kde můžete rozšiřovat, nahrazovat nebo přepisovat logiku vazeb dat. Vizuální editor Javy poskytuje vizuální podporu pro objekty vazeb tím, že na neformátované oblasti pohledu návrhu zobrazuje datové objekty, zdroje dat a vazací programy, které vaše aplikace používá. Vizuální editor vykresluje čáry mezi vizuálními komponentami a datovými objekty a zobrazuje tak aktuální vazby pro libovolný vybraný objekt.

Následující diagram je jednoduchým přehledem toho, jak jsou vizuální komponenty, vazací programy, datové objekty a zdroje dat v interakci. Aplikace, kterou sestavíte v tomto výukovém programu, ilustruje trochu komplexnější a kreativnější využití vazacích programů. Tento diagram nepředstavuje přímo vazací programy, datové objekty a zdroje dat ve vzorových aplikacích, které sestavujete.



Obrázek 1. Tento diagram uvádí ukázkový vztah mezi vizuálními komponentami, vázacími programy, datovými objekty a zdroji dat

V prvním kroku má každá vizuální komponenta vlastní vázací program, který ji přidruží k datovému objektu nebo, v případě tlačítka, k datovému zdroji. Vázací programy pro textová pole spojují pole s konkrétní vlastností datového objektu. Jak datový objekt řádku tak datový objekt v tomto diagramu získávají data z přímých volání služeb na zdroji dat. Datový objekt pro textová pole používá hodnotu klíče z vybraného řádku v tabulce jako argument pro volání služby Service2, která vrátí úplný záznam, který bude pravděpodobně zahrnovat více informací o vybraném řádku v tabulce. Tento úplný záznam se poté použije jako argument pro vázací program akcí tlačítka, když volá službu Service3, což může být metoda, která aktualizuje hodnoty zadané do polí. Podrobnější vysvětlení datových objektů, datových vázacích programů a zdrojů dat viz dříve uvedené odkazy.

Generování proxy prostředí Java pro webovou službu ve vašem projektu pomocí souboru WSDL

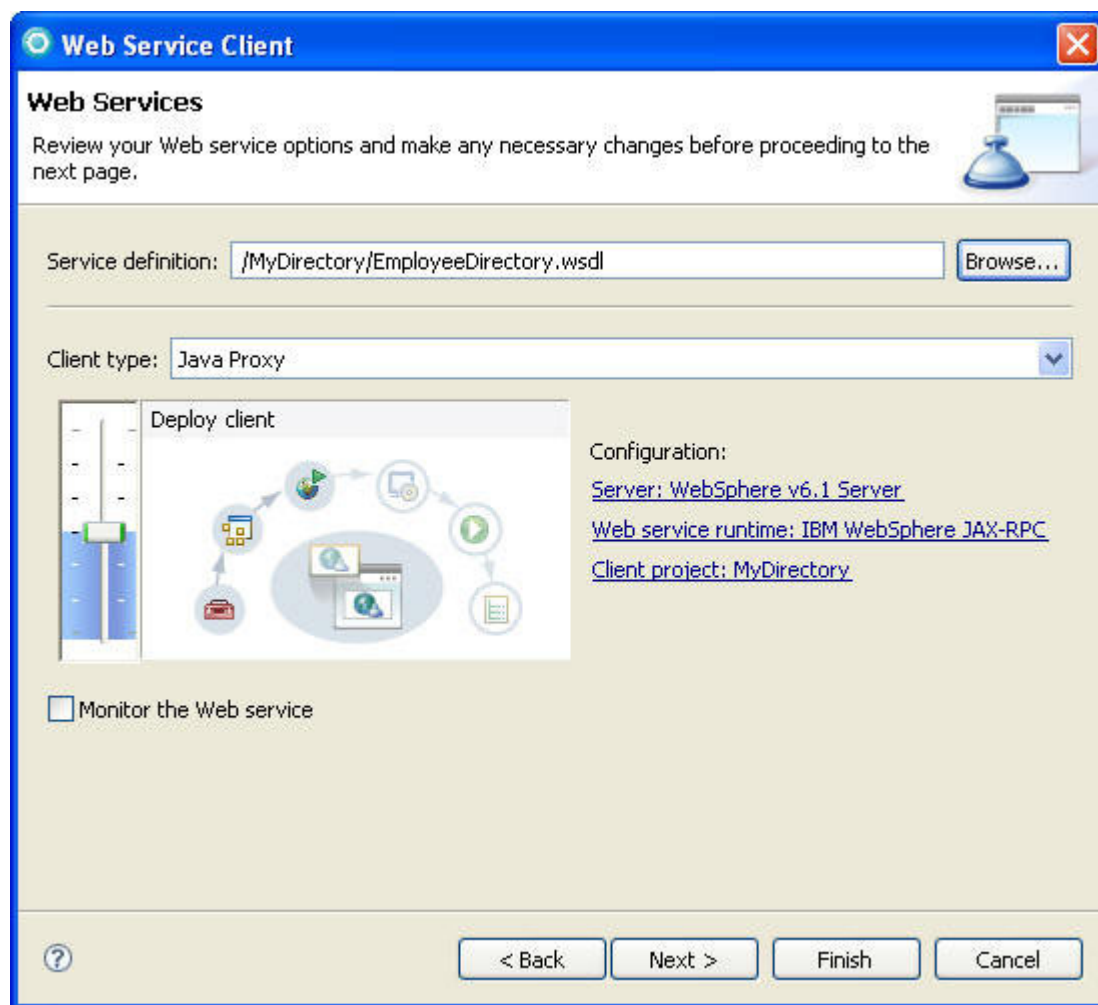
Abyste mohli pracovat s webovou službou spuštěnou na serveru, vyžaduje vaše aplikace Java proxy prostředí Java, nebo klienta, se kterým bude vzájemně reagovat. Pomocí souboru WSDL můžete generovat proxy prostředí Java ve vašem projektu Java pomocí průvodce klientem webové služby. Projekt MyDirectory zahrnuje soubor EmployeeDirectory.wsdl, který použijete pro vytvoření tohoto serveru proxy. Poté, co vytvoříte proxy prostředí Java, můžete vytvořit zdroj dat, který představuje webovou službu, a začít vytvářet vazby mezi vizuálními komponentami.

Důležité: Soubor WSDL, který se používá v tomto cvičení, předpokládá, že jste implementovali webovou službu na lokální instalaci serveru WebSphere Application Server a použili předvolený port pro lokálního hostitele (<http://localhost:9080>). Pokud jste soubor EAR implementovali jinak, musíte příslušným způsobem upravit soubor WSDL, než budete pokračovat.

Vytvoření proxy prostředí Java webové služby ve vašem projektu:

1. V hlavní nabídce klepněte na **Soubor** → **Nový** → **Další** a vyberte průvodce **Webové služby** → **Klient webové služby**. Pokud se kategorie Webové služby nezobrazuje, vyberte **Zobrazit všechny průvodce**.
2. Pomocí průvodce definujte klienta webových služeb:

- Jako **Definici služby** zadejte soubor WSDL, který je součástí projektu MyDirectory: /MyDirectory/EmployeeDirectory.wsdl
- V poli **Typ klienta** vyberte **Proxy prostředí Java**.
- Nastavte posuvný ovladač na **Implementovat klienta**.
- Ujistěte se, že server a běhová komponenta webové služby jsou pro váš server řádně nastavené. Tento výukový program byl testován na serverech WebSphere v6.0 a WebSphere v6.1 s běhovou komponentou IBM WebSphere JAX-RPC.
- Ujistěte se, že klient proxy prostředí Java je výstupem projektu MyDirectory.



- Klepněte na **Dokončit**. Průvodce Klient webové služby generuje proxy prostředí Java do nového balíčku (directory.service) ve vašem projektu.

Vazba tabulky employeesTable s objektem dat řádku vráceným webovou službou

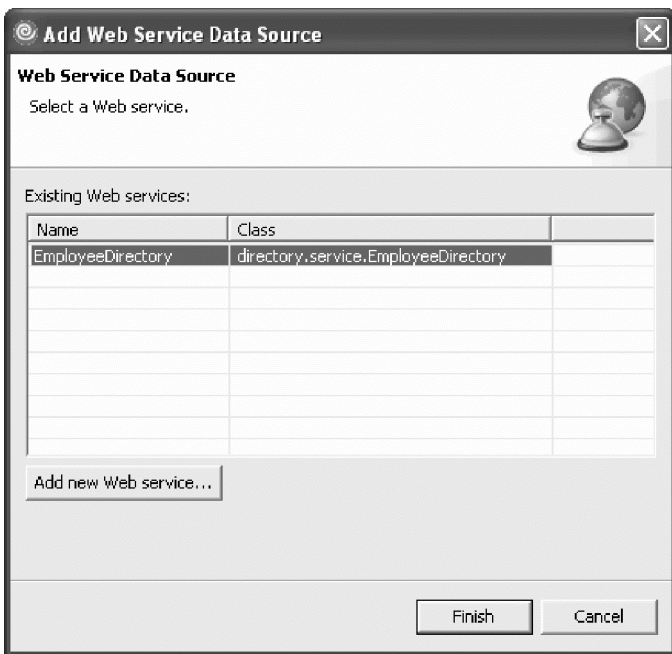
Protože tabulka employeesTable je první vizuální komponenta, kterou svazujete v této aplikaci, musíte vytvořit zdroj dat, který odkazuje na ukázkový server proxy webové služby, který jste právě přidali do svého projektu. Až budete svazovat další vizuální komponenty v dalších cvičeních, znovu použijete tento zdroj dat. V tomto kroku přidáte zdroj dat webové služby a datový objekt lightEmployeeRecordRows.

Svázání tabulky zaměstnanců:

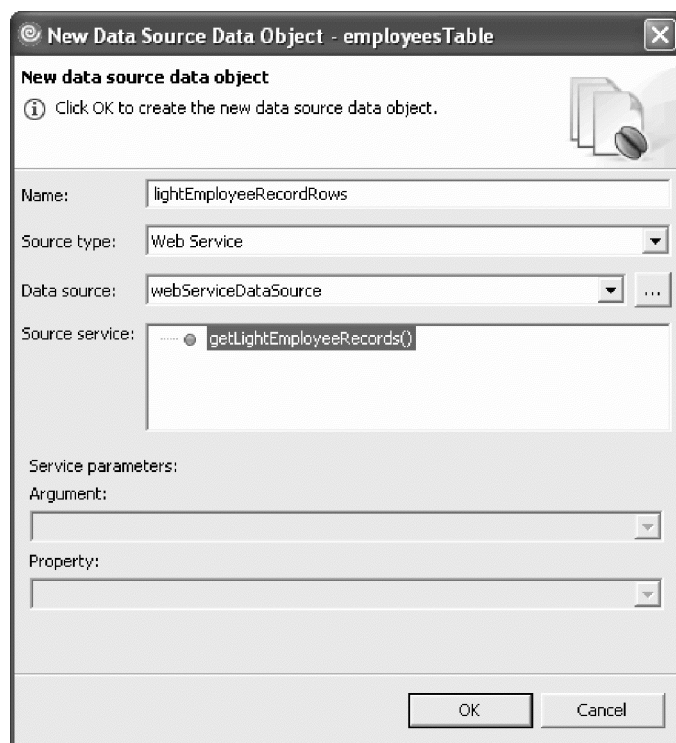
- V pohledu Java Beans nebo v pohledu návrhu vyberte employeesTable. (Ujistěte se, že jste nevybrali nadřazený JScrollPane.) V oblasti návrhu se nahoře od employeesTable zobrazí malá karta s názvem **Vazba**.

Filter...					[Last name]	
Bind...						
A	B	C	D	E		
0x0	0x1	0x2	0x3	0x4		New
1x0	1x1	1x2	1x3	1x4		Delete
2x0	2x1	2x2	2x3	2x4		
3x0	3x1	3x2	3x3	3x4		
4x0	4x1	4x2	4x3	4x4		

- Klepněte na kartu **Vazba** na employeesTable. Můžete případně klepnout pravým tlačítkem na tabulku employeesTable a vybrat **Vlastnosti vazeb**.
- Protože ve vaší aplikaci nejsou žádné datové objekty, musíte přidat nový objekt. Klepněte na **Nový datový objekt zdroje dat**.
- V poli **Typ zdroje** vyberte **Webová služba**.
- Protože jste ještě nepřidali zdroj dat webové služby do vaší aplikace, budete ho muset přidat nyní. Vedle pole **Zdroj dat** klepněte na tlačítko ... a otevřete dialogové okno Přidat zdroj dat webové služby, které vyhledá dostupné klienty webové služby nebo servery proxy ve vašem projektu.
- Vyberte webovou službu EmployeeDirectory a klepněte na Dokončit. Do souboru DirectoryApp.java se přidá nový zdroj dat.

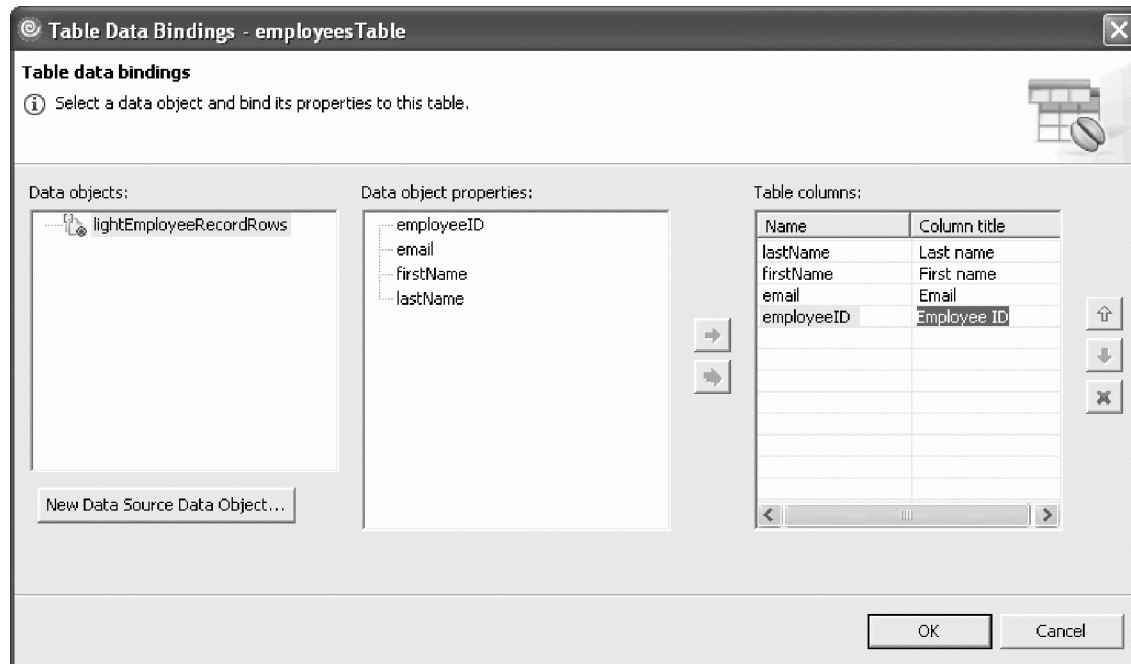


- V dialogovém okně Nový datový objekt zdroje dat vyberte getLightEmployeeRecords() v poli Služba zdroje a akceptujte předvolené jméno pro nový datový objekt: lightEmployeeRecordRows. Pro tuto metodu služby nejsou požadované žádné parametry. Klepněte na OK. Nový datový objekt se vytvoří a zobrazí na neformátované oblasti pohledu návrhu.



Tip: Protože svazujete tabulku, zobrazí dialogové okno Nový datový objekt zdroje dat pouze ty služby, které vracejí objekty dat řádku. V tomto případě je metoda `getLightEmployeeRecords()` jediná dostupná služba, která vrací pole objektů.

8. V dialogovém okně Vazby tabulkových dat vyberte datový objekt `lightEmployeeRecordRows`.
9. Nyní musíte vybrat vlastnosti datového objektu `lightEmployeeRecordRows`, které chcete zobrazit v tabulce `employeesTable`:



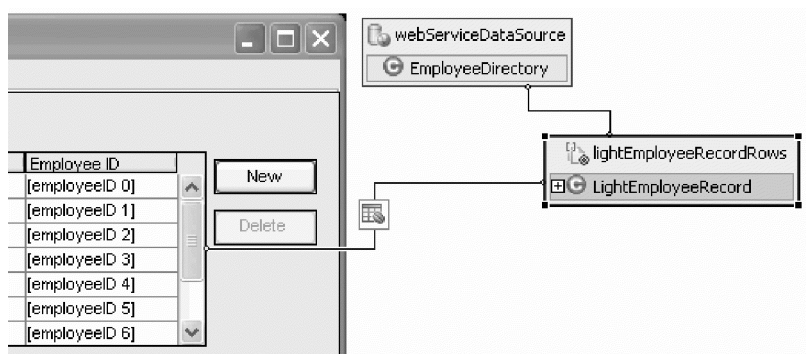
- a. Klepněte na tlačítko s obousměrnou šipkou  a přidejte všechny vlastnosti objektů do seznamu **Sloupce tabulky**.

- b. Použijte šipky nahoru a dolů, abyste uspořádali sloupce do následujícího pořadí, odshora dolů: lastName, firstName, email, employeeID
- c. Přejmenujte názvy sloupců: Příjmení, Jméno, Email, ID zaměstnance

Tip: Poté, co dokončíte svázání tabulky, můžete se kdykoliv vrátit do vlastností vazeb a přejmenovat sloupce nebo změnit jejich pořadí.

- d. Klepněte na **OK**.

Tabulka employeesTable je nyní svázána s datovým objektem lightEmployeeRecordRows pomocí JRowTableBinder. Když klepnete na datový objekt lightEmployeeRecordRows na neformátované oblasti, vizuální editor nakreslí čáru od datového objektu k tabulce. Na čáře je JRowTableBinder znázorněn ikonou vázacího programu tabulky . Další čára indikuje, že datový objekt používá jako zdroj dat webServiceDataSource.



Kontrolní bod lekce

Povšimněte si změn ve vašem projektu a aplikaci. Během této lekce jste přidali zdroj dat webové služby, objekt dat řádku a vázací program, který svazuje employeesTable s řádkovým datovým objektem.

Prozkoumejte nový balíček (jve.generated), který se vytvořil ve vašem projektu pro třídy vázacího programu vytvořené vizuálním editorem Javy. Také si všimněte nového balíčku (directory.service), který obsahuje proxy prostředí Java pro Web service.Describe a shrňte, co jste se naučili v této lekci.



Nyní, když spustíte aplikaci My Company Directory, obsadila webová služba tabulku zaměstnanců stávajícími záznamy zaměstnanců.

Lekce 2.3: Svázání polí s podrobnostmi s výběrem tabulky

V předchozím cvičení jste svázali tabulku employeesTable s datovým objektem lightEmployeeRecordRows vráceným službou getLightEmployeeRecords() ve webové službě. Nyní musíte obsadit pole s podrobnostmi na základě zaměstnance, který je v tabulce vybrán.

Pro další podrobnosti o každém ze zvolených zaměstnanců se používá další datový objekt. Datový objekt selectedEmployeeRecord, který budete přidávat, je vrácen službou getFullEmployeeRecord(). Tato služba bere ID vybraného uživatele v tabulce jako parametr a přináší další podrobnosti o zaměstnanci, včetně telefonního čísla a pracoviště.

Vázací program JRowTableBinder použitý pro svázání tabulky s řádkovým datovým objektem tento krok usnadňuje. JRowTableBinder ukazuje vybraný prvek v tabulce jako samostatný datový objekt, který je možné použít jako parametr pro metodu `getFullEmployeeRecord(java.lang.Integer)`. Můžete pak snadno svázat každé z textových polí s příslušnou vlastností v datovém objektu `selectedEmployeeRecord`.

Zjistit více o této webové službě: Tato webová služba obsahuje dvě služby pro získání všech podrobností o každém ze zaměstnanců. Tabulka uvádí seznam všech zaměstnanců, v tabulce se zobrazí pouze část. Poté, když je vybrán jediný zaměstnanec, můžete načíst zbytek informací pouze pro vybraného zaměstnance. Pokud by webová služba odesílala všechna data pro každého zaměstnance, když tabulka požaduje data, přenos by byl příliš náročný a způsobil by pomalejší výkon aplikace.

Například, pokud záznam zaměstnance zahrnuje fotografii nebo přílohu, pravděpodobně byste nechtěli načítat všechny fotografie, když chcete pouze získat seznam zaměstnanců. Proto se služba `getLightEmployeeRecord` používá pro obsazení tabulek a služba `getFullEmployeeRecord` získává plný záznam pro zaměstnance vybraného v tabulce.

Svázání pole Příjmení

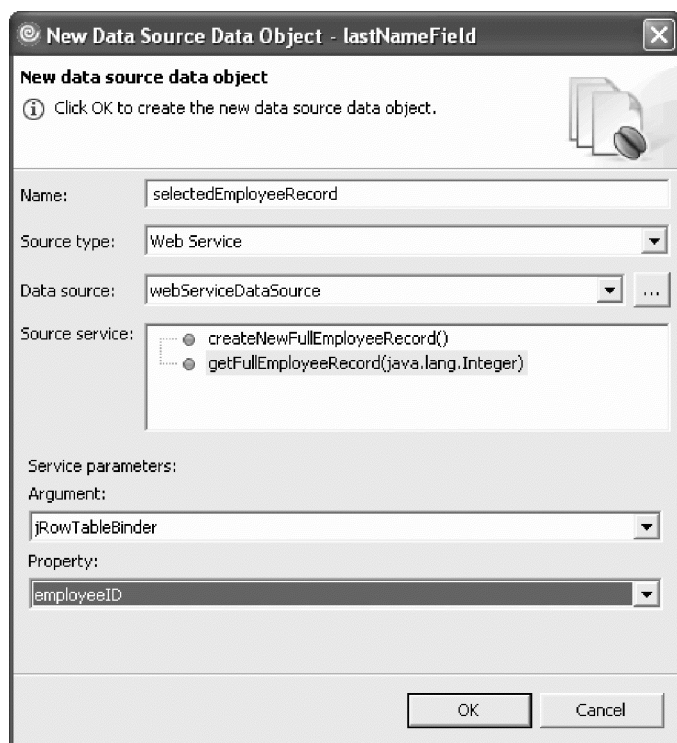
V tomto kroku svážete pole **Příjmení** s vlastností `lastName` v datovém objektu `selectedEmployeeRecord`:

1. V pohledu Java Beans nebo v pohledu Návrh vyberte `JTextField` jako příjmení (`lastNameField`). Oblast návrhu zobrazuje v textovém poli kartu **Vazba**.

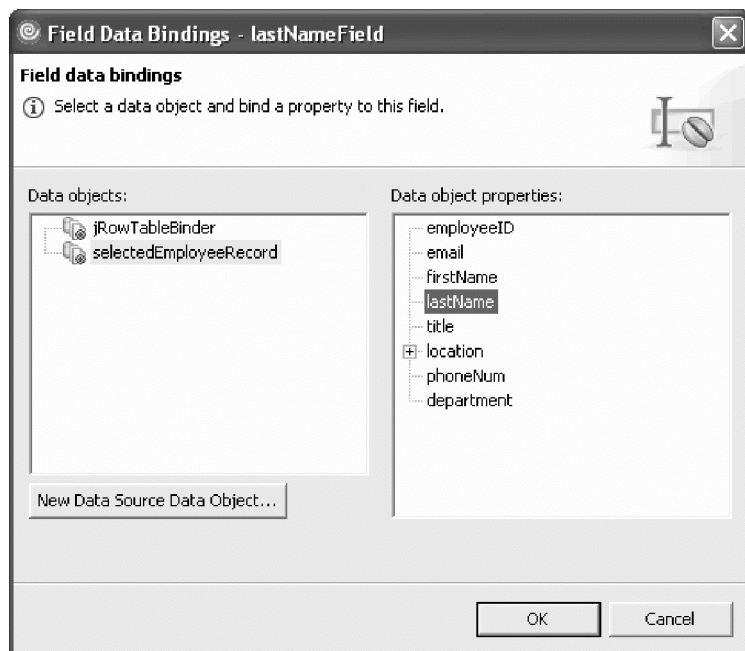


2. Klepnutím na kartu **Vazba** otevřete dialogové okno Datové vazby pole.
3. Klepněte na **Nový datový objekt zdroje dat**. Přestože stávající datový objekt `JRowTableBinder` vrací správné příjmení, nezahrnuje plný záznam zaměstnance. Musíte vytvořit nový datový objekt, který bude představovat plný záznam zaměstnance.
4. Ujistěte se, že je v poli **Typ zdroje** vybráno **Webová služba** a že je pro **Zdroj dat** vybráno **webServiceDataSource**.
5. V seznamu **Služba zdroje** vyberte `getFullEmployeeRecord(java.lang.Integer)`. Dialogové okno Nový datový objekt zdroje dat uvádí seznam služeb, které vracejí datové objekty, které jsou kompatibilní s textovým polem.
6. V poli **Název** zadejte `selectedEmployeeRecord`.
7. V poli **Argument** vyberte `JRowTableBinder` a v poli **Vlastnost** vyberte `employeeID`. ID zaměstnance ve zvoleném řádku je nyní nastaveno tak, aby bylo argumentem pro metodu služby `getFullEmployeeRecord()`.

Poznámka: `getFullEmployeeRecord(java.lang.Integer)` vyžaduje jako argument celé číslo. Chcete použít ID zaměstnance z aktuálního výběru v tabulce zaměstnanců pro načtení plného záznamu. Když svážete tabulku, vizuální editor automaticky generuje vázací program `JRowTableBinder`, který naslouchá aktuálnímu výběru v tabulce zaměstnanců. Jako celočíselný parametr použijte `employeeID` z vybraného řádku v `JRowTableBinder`.



8. Klepněte na **OK**.
9. V dialogovém okně **Datové vazby** pole se ujistěte, že je v seznamu **Datové objekty** vybráno `selectedEmployeeRecord`. Všimněte si, že existuje více dostupných vlastností pro datový objekt `selectedEmployeeRecord` než pro datový objekt `jRowTableBinder`.
10. V seznamu **Vlastnosti datových objektů** vyberte vlastnost `lastName`.



11. Klepněte na **OK**. Pole příjmení ve vaší aplikaci je nyní svázané s vlastností `lastName` vybraného datového objektu `selectedEmployeeRecord`, který je vrácen od `getFullEmployeeRecord()`.

Vytvoří se nový datový objekt jménem `selectedEmployeeRecord` a přidá se do vaší aplikace. Vizualní znázornění datového objektu se přidá do neformátované oblasti pohledu **Návrh**, tak jak uvádí následující obrázek:



Nyní, když vyberete v oblasti návrhu pole lastName, indikuje čára, že je svázáno se selectedEmployeeRecord. Uprostřed čáry ikona textového vázacího programu představuje SwingTextComponentBinder, který se pro tuto vazbu používá. Pokud vyberete čáru nebo ikonu představující vázací program v oblasti návrhu, můžete prozkoumat vlastnosti vázacího programu v pohledu Vlastnosti.

Vazba zbývajících polí s podrobnostmi

Abyste svázali každé ze zbývajících polí s podrobnostmi o zaměstnancích, zopakujete podobný proces jako s polem příjmení, ale nemusíte přidávat datový objekt. Protože jste již přidali datový objekt selectedEmployeeRecord, můžete prostě svázat každé pole s jemu odpovídající vlastností v datovém objektu selectedEmployeeRecord.

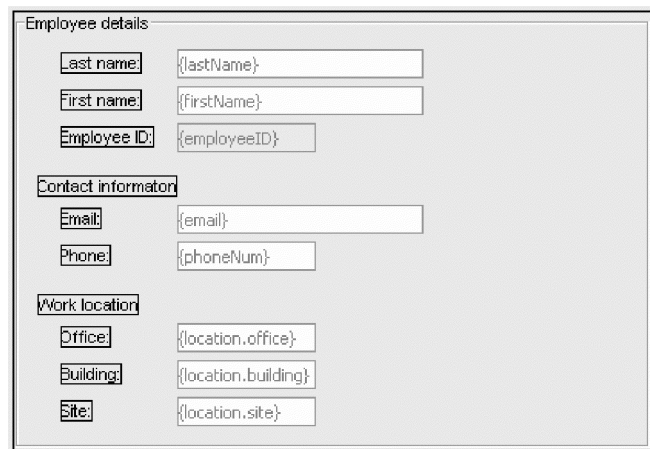
Abyste vytvořili vazby polí, proveďte následující kroky pro každé z polí v sekci aplikace Podrobnosti o zaměstnancích:

1. Vyberte pole v pohledu Návrh a klepněte na kartu **Vazba**.
2. V dialogovém okně Datové vazby polí vyberte selectedEmployeeRecord ze seznamu **Datové objekty**.
3. V seznamu **Vlastnosti datových objektů** vyberte odpovídající vlastnost pro pole, které svazujete. Následující tabulka uvádí vlastnost, se kterou je třeba každé textové pole svázat:

Pole	Vlastnost v datovém objektu selectedEmployeeRecord
lastNameField	lastName
firstNameField	firstName
idField	employeeID
emailField	email
phoneField	phoneNum
officeField	location.office
buildingField	location.building
siteField	location.site

4. Klepněte na **OK**.

Až dokončíte svázání textových polí, měla by oblast návrhu vypadat takto:



Nastavení pole ID zaměstnance pouze pro čtení

Pole ID zaměstnance je znepřístupněné, protože je vlastnost pro úpravu pole nastavena na false. Nicméně, předvolené chování vázacího programu textového pole změni zpřístupněný stav na poli, když datový objekt obsahuje nějakou hodnotu. Toto chování vázacího programu můžete vypnout, aby pole zůstalo ve výchozím stavu pouze pro čtení.

Abyste vázacímu programu zabránili automaticky přepínat možnost upravit vlastnost:

1. Vyberte pole ID zaměstnance. V oblasti návrhu se zobrazí čára s ikonou , která představuje vázací program pro dané pole.
2. Klepněte na ikonu vázacího programu  pro pole ID zaměstnance.
3. V pohledu Vlastnosti změňte vlastnost autoEditable na **false**. Stiskněte klávesu **Enter**.

Kontrolní bod lekce

Nyní, když spustíte aplikaci a vyberete zaměstnance z tabulky, zobrazí se v polích s podrobnostmi podrobnosti ze záznamu tohoto zaměstnance.

Lekce 2.4: Svázání tlačítka Aktualizovat s vázacím programem akce

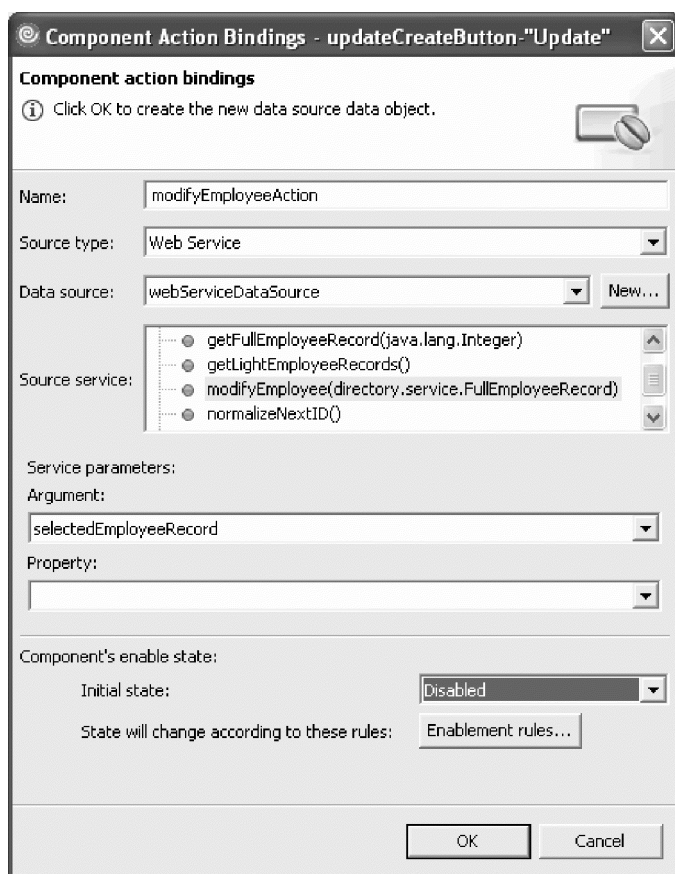
Vizuální editor Javy poskytuje vázací programy pro volání služby na zdroji dat, když se klepne na tlačítko. Například, když se klepne na tlačítko Aktualizovat, aplikace by měla spustit metodu modifyEmployee() na webové službě se změnami zadanými v polích s podrobnostmi. V této lekci svážete tlačítko Aktualizovat s vázacím programem akce.

Svázání tlačítka Aktualizovat:

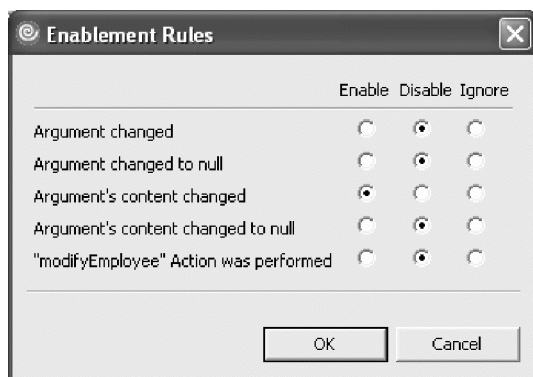
1. V oblasti návrhu vyberte tlačítko **Aktualizovat** a klepnutím na kartu **Vazba** otevřete dialogové okno Vazby akcí komponenty.



2. V poli **Typ zdroje** vyberte **Webová služba**.
3. V poli **Zdroj dat** vyberte **webServiceDataSource**.
4. Ze seznamu **Služba zdroje** vyberte **modifyEmployee(directory.service.FullEmployeeRecord)**.
5. Pole **Název** se automaticky změni na **modifyEmployeeAction**. Potvrďte tuto předvolbu.
6. V poli **Argument** vyberte **selectedEmployeeRecord**.
7. Protože metoda modifyEmployee() má jako argument plný záznam zaměstnance, musíte nechat pole **Vlastnost** prázdné.
8. Nastavte **Výchozí stav** tlačítka na **Znepřístupněno**.



9. Abyste definovali, jak tlačítko mění stav, klepněte na **Pravidla zpřístupnění**. Uveďte, že toto tlačítko se zpřístupní pouze tehdy, když se změní obsah argumentu a je znepřístupněné pro všechny ostatní instance. Klepněte na **OK**.



To znamená, že tlačítko **Aktualizovat** nebude zpřístupněné, dokud se obsah `selectedEmployeeRecord` nezmění. Jinými slovy, když do jednoho z polí s podrobnostmi svázaných se `selectedEmployeeRecord` zapíšete novou hodnotu, vázací program zpřístupní toto tlačítko. Pokud vyberete nový záznam nebo klepnete na **Aktualizovat**, tlačítko se opět znepřístupní.

10. Klepněte na **OK**.

Pro tlačítko **Aktualizovat** se přidá nový vázací program `SwingDataServiceAction`. Pokud v oblasti návrhu vyberete toto tlačítko, vizuální editor nakreslí čáru, která indikuje, že toto tlačítko je svázané se zdrojem dat webové služby. Růžová tečkovaná šipka ukazuje od objektu `selectedEmployeeRecord` k této čáře. Tato šipka indikuje, že `selectedEmployeeRecord` je argumentem pro volání této služby.

Kontrolní bod lekce

Nyní, když spustíte aplikaci, můžete aktualizovat záznam zaměstnanců.

Vyberte zaměstnance v tabulce a změňte příjmení. Když změníte příjmení, zpřístupní se tlačítko **Aktualizovat**. Když klepnete na **Aktualizovat**, volá se služba `modifyEmployee` a záznam zaměstnance se aktualizuje. V tabulce bude uvedeno nové příjmení.

Lekce 2.5: Zpřístupnění tlačítka Odstranit a dialogového okna Potvrdit odstranění

V tomto cvičení naprogramujete aplikaci `My Company Directory` pro odstranění záznamu zaměstnance.

Následující seznam popisuje chování, které chcete, aby aplikace používala:

- Když vyberete zaměstnance v tabulce, zpřístupní se tlačítko **Odstranit**.
- Když klepnete na tlačítko **Odstranit**, otevře se dialogové okno a vyzve vás, abyste odstranění potvrdili.
- Když klepnete na tlačítko **Ano** v dialogovém okně **Potvrdit odstranění**, záznam zaměstnance se odstraní, okno **Potvrdit odstranění** se zavře a seznam zaměstnanců se obnoví.
- Když klepnete na **Ne**, výmaz se zruší a dialogové okno **Potvrdit odstranění** se zavře.

Programování zpřístupnění tlačítka Odstranit podle toho, zda je v tabulce vybraný řádek nebo ne

Abyste naprogramovali zpřístupnění tlačítka **Odstranit**, přidejte k tabulce posluchač, který tlačítko zpřístupní, když je vybraný řádek.

1. V pohledu `Java Beans` vyberte `employeesTable`. Pohled zdroje zvýrazňuje následující řádek:

```
employeesTable = new JTable();
```
2. Těsně za tento řádek přidejte nový posluchač `ListSelectionListener` a událost `valueChanged` pro tabulku `employeesTable`:

```
employeesTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {  
    public void valueChanged(ListSelectionEvent e) {  
        getDeleteButton().setEnabled(getEmployeesTable().getSelectedRowCount() != 0);  
    }  
});
```
3. Poté, co ke kódu přidáte tyto řádky, editor zdroje je označí jako chyby, dokud neimportujete `ListSelectListener` a `ListSelectionEvent`. Pro přidání požadovaných importů klepněte na **Zdroj** → **Organizovat import** v hlavní nabídce. Do sekce importů ve třídě se přidají následující řádky:

```
import javax.swing.event.ListSelectionEvent;  
import javax.swing.event.ListSelectionListener;
```

Nyní, když vyberte řádek v tabulce, zpřístupní se tlačítko **Ostranit**.

Programovat otevření dialogového okna Potvrdit odstranění po klepnutí na tlačítko Odstranit.

Přidejte událost `actionPerformed` k tlačítku **Odstranit** a naprogramujte událost tak, aby otevřela dialogové okno **Potvrdit odstranění**.

1. Klepněte pravým tlačítkem na tlačítko **Odstranit** a vyberte **Události** → **actionPerformed**. K metodě `getDeleteButton()` se přidá následující stub:

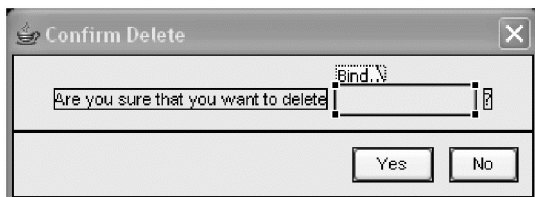
```
deleteButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent e) {  
        System.out.println("actionPerformed()");  
        // TODO Automaticky generovaný stub události actionPerformed()  
    }  
});
```
2. Nahraďte tento generovaný stub následujícím kódem, který nastaví, že se po klepnutí na tlačítko zobrazí dialogové okno **Potvrdit odstranění**:

```
deleteButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent e) {  
        getConfirmDialog().setVisible(true);  
    }  
});
```

Svázání textového pole v dialogovém okně Potvrdit odstranění

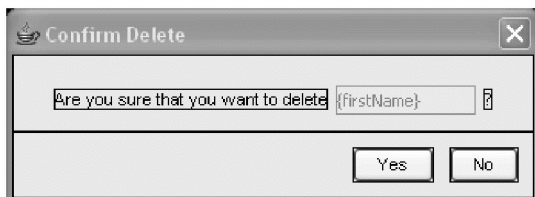
Svažte textové pole v dialogovém okně Potvrdit odstranění, aby zobrazovalo jméno zaměstnance, který se má odstranit.

1. V pohledu Java Beans nebo v oblasti návrhu vyberte textové pole `employeeToDeleteField` a klepněte na kartu **Vazba**.



2. V dialogovém okně Datové vazby polí vyberte datový objekt `selectedEmployeeRecord` a pole `firstName` a pak klepněte na **OK**.

Textové pole je nyní svázáno se sloupcem `firstName` ve vybraném řádku v `employeesTable`.



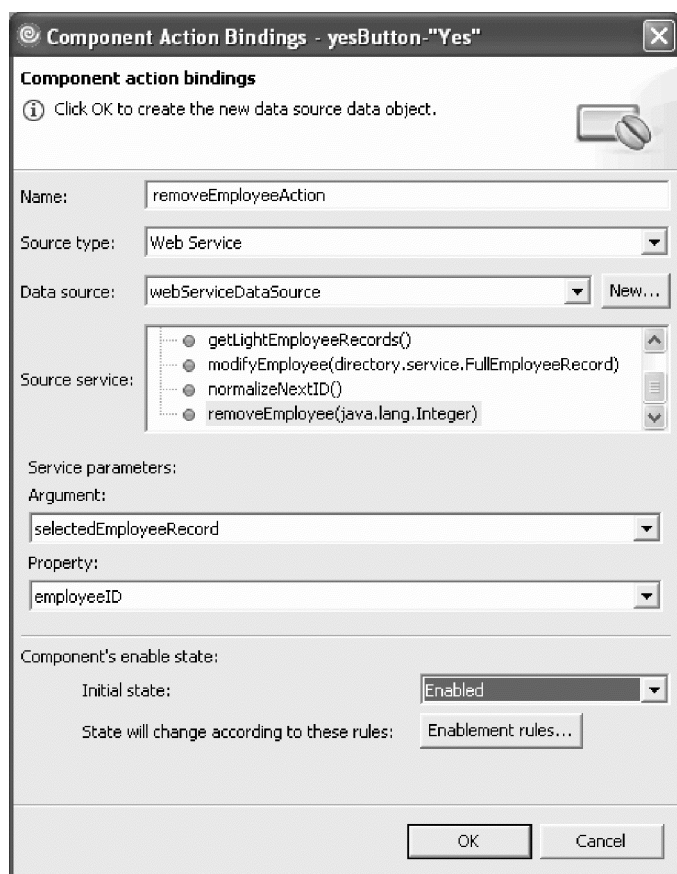
3. Abyste zajistili, že toto pole bude pouze pro čtení, nastavte vlastnost **autoEditable** pro vázací program pole na **false**.

Vazba tlačítka Ano pro provedení odstranění

Svazání tlačítka **Ano**, aby volalo metodu `removeEmployee(java.lang.Integer)` na webové službě.

1. Vyberte tlačítko **Ano** a klepnutím na kartu **Vazba** otevřete dialogové okno Vazby akcí komponenty.
2. V poli **Typ zdroje** vyberte **Webová služba**.
3. V poli **Zdroj dat** vyberte **webServiceDataSource**.
4. Ze seznamu **Služba zdroje** vyberte **removeEmployee(java.lang.Integer)**.
5. Pole **Název** se automaticky změní na **removeEmployeeAction**. Potvrďte tuto předvolbu.
6. V poli **Argument** vyberte **selectedEmployeeRecord**.
7. V poli **Vlastnost** vyberte **employeeID**. Protože metoda `removeEmployee()` má jako argument celé číslo, použijte ID zaměstnance z `selectedEmployeeRecord`.
8. Nastavte **Výchozí stav** tlačítka na **Zpřístupněno**.
9. Pro **Pravidla zpřístupnění** vyberte **Ignorovat** pro každou ze situací.

Tento stav komponenty znamená, že tlačítko **Ano** bude vždy zpřístupněné, protože není třeba, aby měnilo stav.



10. Klepněte na **OK**.

Přidání události pro skrytí dialogového okna Potvrdit odstranění po odstranění zaměstnance

V tomto kroku přidáte událost k *vázacímu programu* tlačítka **Ano** (nikoliv k samotnému tlačítku **Ano**). Chcete, aby se okno Potvrdit odstranění zavřelo poté, se je zaměstnanec odstraněn, což znamená poté, co vázací program úspěšně volal službu na zdroji dat.

Přidejte následující kód do metody `getRemoveEmployeeAction()`:

```
removeEmployeeAction.addActionBinderListener(new jve.generated.IActionBinder.ActionBinderListener() {
    public void afterActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {
        getConfirmDialog().setVisible(false);
    }
    public void beforeActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {}
});
```

Tento kód události skryje dialogové okno Potvrdit odstranění poté, co je provedena akce vázacího programu.

Kontrolní bod lekce

Nyní, když spustíte aplikaci My Company Directory, můžete vybrat zaměstnance z tabulky, klepnout na tlačítko **Odstranit** a klepnout na **Ano** pro potvrzení odstranění. Záznam zaměstnance bude odstraněn z adresáře a seznam zaměstnanců bude odrážet toto odstranění.

Lekce 2.6: Nastavení akcí a vazeb pro přidání nového zaměstnance

V této lekci povolíte aplikaci My Company Directory přidat nový záznam o zaměstnanci.

Protože je chování aplikace při přidávání nového zaměstnance komplikované a dynamické, je i toto cvičení nezbytně velmi komplexní a vyžaduje, abyste manuálně provedli některé změny ve zdrojovém kódu. Toto cvičení také demonstruje některé rozšířené schopnosti datových objektů a dává vám kreativní příklad, jak využít vazací programy a datové objekty tak, aby odpovídaly vašim potřebám.

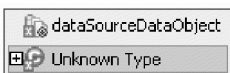
Následující seznam popisuje požadované chování aplikace:

- Když klepnete na tlačítko **Nový**, dojde k následujícímu chování:
 - Vymaže se výběr v tabulce zaměstnanců a tabulka bude znepřístupněná.
 - Vymazání výběru v tabulce způsobí, že se znepřístupní tlačítko **Odstranit**.
 - Znepřístupní se pole **Filtrovat**.
 - Z polí s podrobnostmi se vymažou veškeré hodnoty s výjimkou ID nového zaměstnance.
 - Text na tlačítku **Aktualizovat** se přepne na **Přidat**.
- Když klepnete na tlačítko **Přidat**, dojde k následujícímu chování:
 - Hodnoty zadané do polí s podrobnostmi se přidají do adresáře jako záznam nového zaměstnance.
 - Tabulka se zpřístupní a hodnoty se obnoví.
 - Zpřístupní se pole **Filtr**.
 - Text na tlačítku **Přidat** se přepne zpátky na **Aktualizovat**.

Přidání nového datového objektu zdroje dat, který volá službu createNewFullEmployeeRecord()

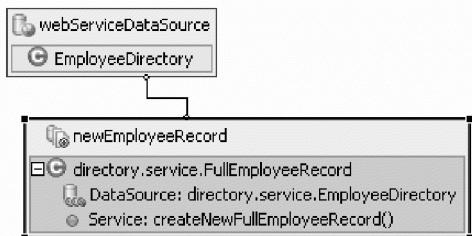
Tato ukázka webové služby poskytuje službu createNewFullEmployeeRecord, která poskytuje nový, prázdný záznam o zaměstnanci, který je obsazen dalším dostupným číslem ID zaměstnance. Prázdný záznam se pak obsadí informacemi o novém zaměstnanci a zadá zpátky webové službě.

1. Na paletě vizuálního editoru Javy rozbalte zásuvku Datové objekty a vyberte **Datový objekt zdroje dat**.
2. Přesuňte ukazatel myši nad prázdnou oblast pohledu Návrh nebo neformátovanou oblast a klepněte levým tlačítkem na Datový objekt zdroje dat. Nový datový objekt zdroje dat se přidá a zobrazí v neformátované oblasti:



3. Klepněte pravým tlačítkem na Datový objekt zdroje dat a vyberte **Přejmenovat pole**. Přejmenujte datový objekt na newEmployeeRecord.
4. Klepněte pravým tlačítkem myši na datový objekt newEmployeeRecord a vyberte **Vlastnosti vazeb**. Otevře se dialog Datové vazby.
5. V poli **Zdroj dat** vyberte webServiceDataSource
6. V poli **Služby** vyberte createNewFullEmployeeRecord()
7. Klepněte na **OK**.

V neformátované oblasti je vidět, že datový objekt zdroje dat newEmployeeRecord je svázán s webovou službou.



Přidat základní datový objekt pro usnadnění přepínání mezi datovými objekty

Protože pole s podrobnostmi a tlačítko aktualizovat se musí přepínat mezi jednotlivými režimy (provedení aktualizace a vytvoření nového zaměstnance), musí být svázány se dvěma datovými objekty v různých časech. Abyste tento krok

usnadnili, přidejte základní datový objekt jménem `switchingDataObject`. Tento základní datový objekt použijete pro přepínání vazby pro textová pole mezi `selectedEmployeeRecord` a `newEmployeeRecord`.

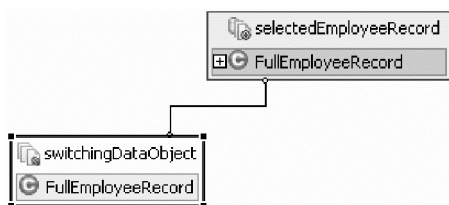
Základní datový objekt prostě odkazuje na jiný datový objekt (`selectedEmployeeRecord`), který jste definovali v předchozím cvičení. Tento nový datový objekt bude užitečný při vytváření metody, která vyzývá základní datový objekt k použití `newEmployeeRecord`, který jste definovali dříve. Jinými slovy, tento základní datový objekt bude fungovat jako zprostředkující datový objekt, který přepíná mezi datovým objektem `selectedEmployeeRecord` a datovým objektem `newEmployeeRecord` a umožňuje tak vizuálním komponentám ve vaší aplikaci pracovat se dvěma různými datovými objekty.

1. Na paletě vizuálního editoru vyberte **Základní datový objekt** a umístěte ho do neformátované oblasti. Přidá se `basicDataObject`.



2. Přejmenujte datový objekt na `switchingDataObject`
3. V zobrazení vlastností pro `switchingDataObject` nastavte vlastnost **sourceObject** na **selectedEmployeeRecord**. Můžete vybrat `selectedEmployeeRecord` z rozbalovací nabídky ve sloupci Hodnota pro danou vlastnost.

Nyní `switchingDataObject` odkazuje na `selectedEmployeeRecord` a obsahuje stejné hodnoty:

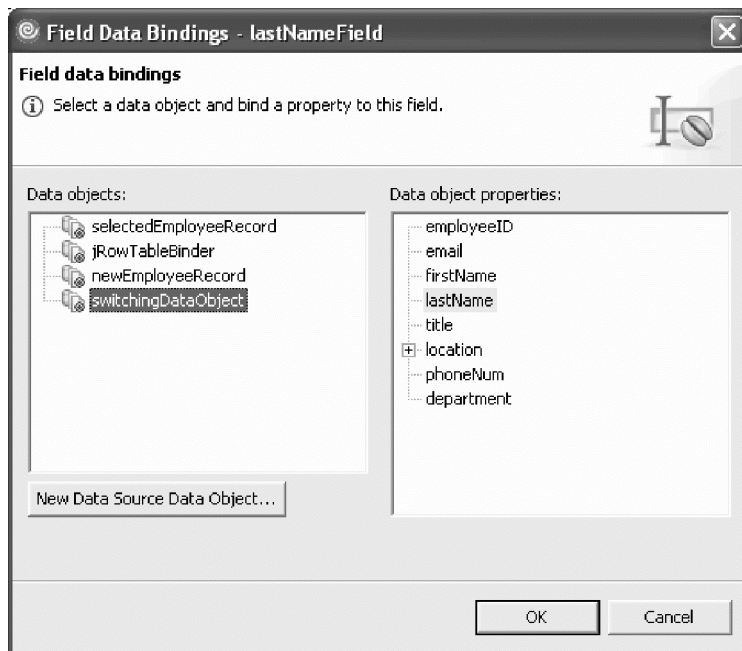


Znovu svázat každé pole zaměstnance s `switchingDataObject`

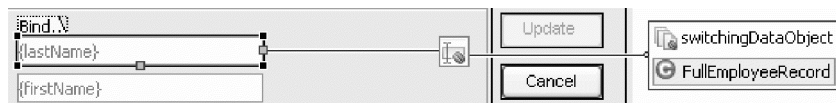
Přestože každé z polí s podrobnostmi o zaměstnancích je již svázané se `selectedEmployeeRecord`, svážete je nyní s `switchingDataObject`. Poté, co svážete pole, můžete dynamicky přepínat mezi datovými objekty pro pole v závislosti na tom, zda upravujete existující záznam zaměstnance nebo chcete přidat nový záznam zaměstnance.

Pro každé z polí v sekci Podrobnosti o zaměstnancích proveďte následující kroky:

1. Vyberte pole a klepněte na kartu **Vazba**.
2. V dialogovém okně Datové vazby vyberte `switchingDataObject`. Již dříve jste svázali tato pole s `selectedEmployeeRecord`.



- Ujistěte se, že je pole stále svázané se správnou vlastností datového objektu a klepněte na **OK**. Pokud vyberete pole v pohledu Návrh, uvidíte, že čáry vázacího programu nyní ukazují na switchingDataObject.



Definování příznaku a metody pro aktualizaci a režimy přepínání

Následující metoda updateMode() kontroluje, zda je příznak režimu nastaven na nový, a potom příslušně změní chování aplikace. Podle předvolby je booleovský příznak isNewMode nastaven na false, a metoda updateMode() zpřístupňuje tabulku zaměstnanců a pole filtrování a nastavuje text na tlačítku Aktualizovat na "Aktualizovat". Když je isNewMode nastaven na true, je tabulka zaměstnanců znepřístupněná a libovolná volba je vymazaná, pole filtrování je znepřístupněné a text na tlačítku Aktualizovat se nastaví na "Přidat".

Přidejte následující kód do třídy DirectoryApp.java těsně před poslední pravou složenou závorkou:

```
private boolean isNewMode = false;
private void updateMode() {
    if (isNewMode) {
        getEmployeesTable().clearSelection();
        getEmployeesTable().setEnabled(false);
        getFilterField().setEditable(false);
        getUpdateCreateButton().setText("Přidat");
    } else {
        getEmployeesTable().setEnabled(true);
        getFilterField().setEditable(true);
        getUpdateCreateButton().setText("Aktualizovat");
    }
}
```

Přidání události actionPerformed k tlačítku Nový

V tomto kroku přidáte kód události pro klepnutí na tlačítko **Nový**. Událost příkáže objektu switchingDataObject použít datový objekt newEmployeeRecord, nastaví příznak události na "new" a spustí metodu updateMode(), kterou jste přidali v předchozím kroku.

- V pohledu Návrh klepněte pravým tlačítkem myši na tlačítko **Nový** a vyberte **Události** → **actionPerformed**. Následující kód je generován v metodě getNewButton():

```
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()"); // TODO Automaticky generovaný stub události actionPerformed()
    }
});
```

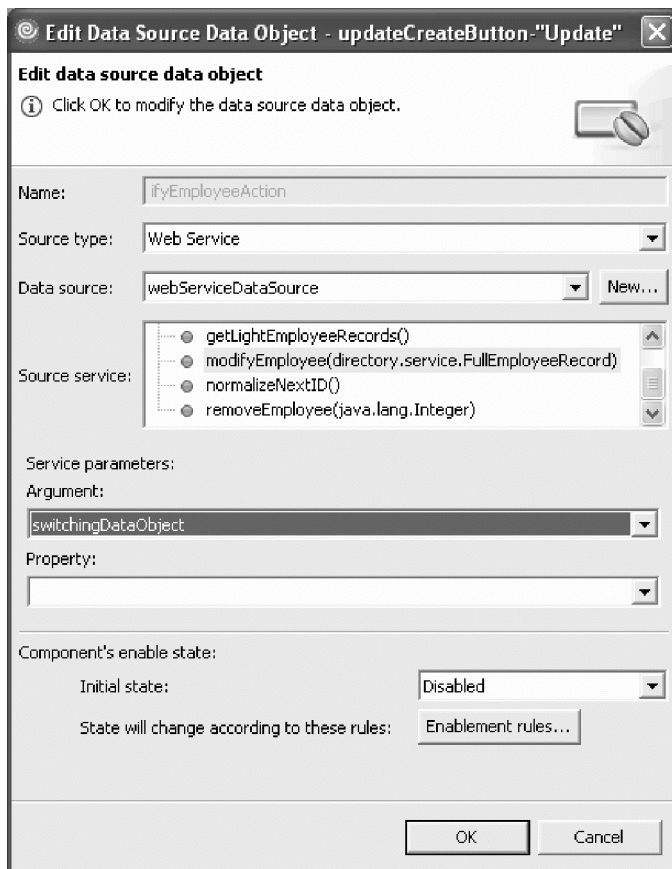
2. Nahraďte tento generovaný stub následujícím kódem:

```
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        getSwitchingDataObject().setSourceObject(getNewEmployeeRecord());
        getNewEmployeeRecord().refresh();
        isNewMode = true; //nastaví aplikaci do nového režimu
        updateMode(); //změní UI podle nového režimu
        getLastNameField().grabFocus();
    }
});
```

Znovu svázat tlačítko Aktualizovat

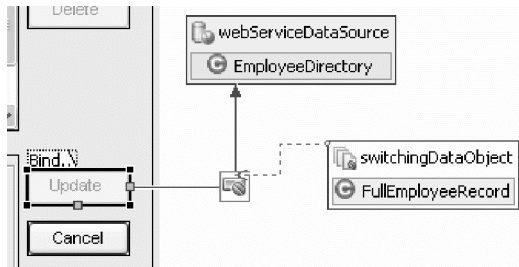
V předchozí lekci jste naprogramovali tlačítko **Aktualizovat**, aby používalo metodu `modifyEmployee` na webové službě. Tato akce se implementuje jako `SwingDataServiceAction`. Jedna z vlastností `SwingDataServiceAction` je zdrojový objekt, který funguje jako argument pro službu. Zdrojový objekt pro akci změny je momentálně nastaven na `selectedEmployeeRecord`. Abyste tlačítko naprogramovali tak, aby řídilo jak aktualizaci, tak přidání, překonfigurujte akci tlačítka tak, že bude používat `switchingDataObject` jako argument pro službu `modifyEmployee`.

1. V pohledu **Návrh** vyberte tlačítko **Aktualizovat**. Všimněte si růžové tečkované šipky, která ukazuje, že `selectedEmployeeRecord` je argumentem pro volání služby.
2. Klepněte na kartu **Vazba** na tlačítku **Aktualizovat**.
3. V poli **Argument** vyberte `switchingDataObject`.



4. Klepněte na **OK**.

Všimněte si, že akce tlačítka je nyní konfigurovaná tak, aby používala `switchingDataObject` jako jeho argument pro metodu `modifyEmployee`:



Přidáním události do vázacího programu tlačítka obnovíte režim

Po klepnutí na tlačítko **Aktualizovat** a dokončení akce na webové službě budete chtít, aby se aplikace vrátila do předvoleného režimu a chování. Můžete přidat posluchač událostí na vázací program akcí tlačítka, který bude aktualizovat režim a obnovovat tabulku po provedení aktualizace nebo přidání.

Přidejte následující kód do metody `getModifyEmployeeAction()` pro tlačítko **Aktualizovat**:

```
modifyEmployeeAction.addActionBinderListener(new jve.generated.IActionBinder.ActionBinderListener() {
    public void afterActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {
        if (isNewMode) {
            //Go back to using the selectedEmployeeRecord
            getSwitchingDataObject().setSourceObject(getSelectedEmployeeRecord());
            //Revert out of new mode
            isNewMode = false;
            updateMode();
        }
        // Refresh the table's data object
        getLightEmployeeRecordRows().refresh();
    }
    public void beforeActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {}
});
```

Kontrolní bod lekce

Když nyní spustíte aplikaci **My Company Directory**, můžete klepnout na tlačítko **Nový** a přidat nový záznam uživatele.

Lekce 2.7: Programování chování tlačítka **Zrušit**

Když používáte svou aplikaci, chcete mít možnost snadno vrátit libovolné změny, které začnete provádět na záznamu zaměstnance, pokud se je rozhodnete nezadávat. Jinými slovy, potřebujete mít možnost zrušit a vyčistit pole, abyste mohli začít znovu. Abyste přidali tuto funkci, nastavte události `actionPerformed` na tlačítku **Zrušit**.

Následující seznam popisuje požadované chování tlačítka **Zrušit**:

- Když klepnete na **Zrušit**, když jste v novém režimu, aplikace se z nového režimu vrátí.
- Když klepnete na tlačítko **Zrušit** během úpravy záznamu o zaměstnanci, libovolné hodnoty, které jste změnili, se vrátí na původní hodnoty.

Přidání události `actionPerformed` k tlačítku **Zrušit**, aby mělo požadované funkce:

1. V pohledu **Návrh** klepněte pravým tlačítkem na tlačítko **Storno** a vyberte **Události → actionPerformed**. V metodě `getCancelButton()` se generuje následující kód:

```
cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()"); // TODO Automaticky generovaný stub události actionPerformed()
    }
});
```

2. Nahradejte generovaný stub události následujícím kódem:

```
cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        if (isNewMode) {
            getSwitchingDataObject().setSourceObject(getSelectedEmployeeRecord());
            isNewMode = false;
        }
    }
});
```



```

        updateMode();
    } else {
        getSelectedEmployeeRecord().refresh();
    }
}
});

```

Kontrolní bod lekce

V této lekci jste se naučili, jak programovat tlačítko **Zrušit** pro události actionPerformed.

Lekce 2.8: Nastavení filtru na tabulce zaměstnanců

Můžete použít vázací program textových filtrů pro filtrování obsahu tabulky zaměstnanců. Filtr získává vstup z textového pole a filtruje tabulku v závislosti na určité vlastnosti nebo sloupci v tabulce.

V aplikaci použijete znaky zadané v poli **Filtr** pro filtrování příjmení zaměstnance. Pokud je přesná hodnota zadaná v poli **Filtr** přítomná mezi záznamy o příjmení zaměstnanců, zobrazí se v tabulce záznam tohoto zaměstnance.

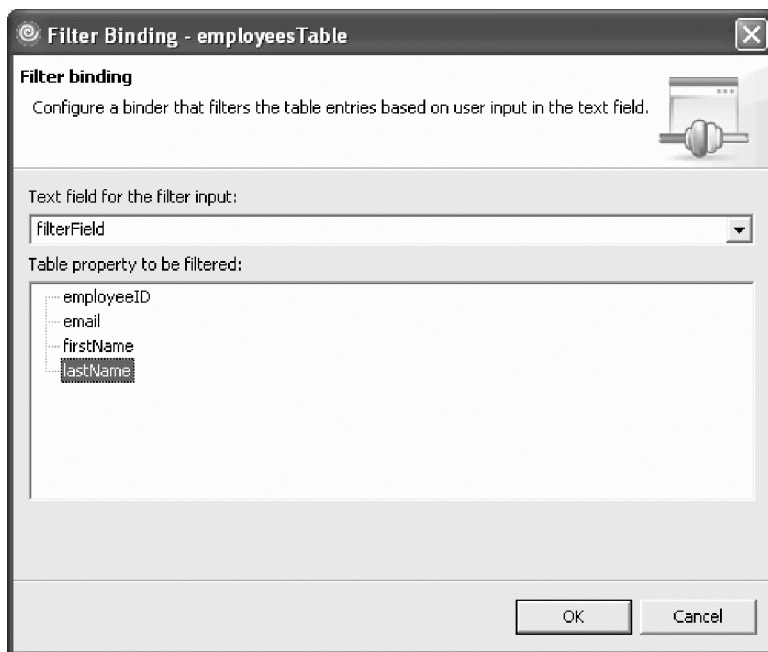
Filter: (Last name)

Last name	First name	Email	Employee ID
Maxwell	Seth	seth.maxwell@my...	24561
Maxwell	Aubrey	aubrey.maxwell@...	30089
Martinez	James	james.martinez@m...	31780

Newv
Delete

Vytvoření filtru pro tabulku:

1. Vyberte ikonu vázacího programu pro tabulku employeesTable a vyberte **Vlastnosti vazeb filtrů**. Otevře se dialogové okno Vazby filtrů.
2. V seznamu **Textové pole pro vstup filtru** vyberte **filterField**.
3. V seznamu **Vlastnost tabulky, která se má filtrovat** vyberte **lastName**.



4. Klepněte na **OK**.

Bude vytvořen nový filtr SwingPropertyFilter. Vlastnost filtru na vázacím programu tabulky se nastaví pro použití nového filtru. Nový filtr je konfigurován pro použití pole **Filtr** pro vstup a filtruje podle vlastnosti lastName v tabulce.

Kontrolní bod lekce

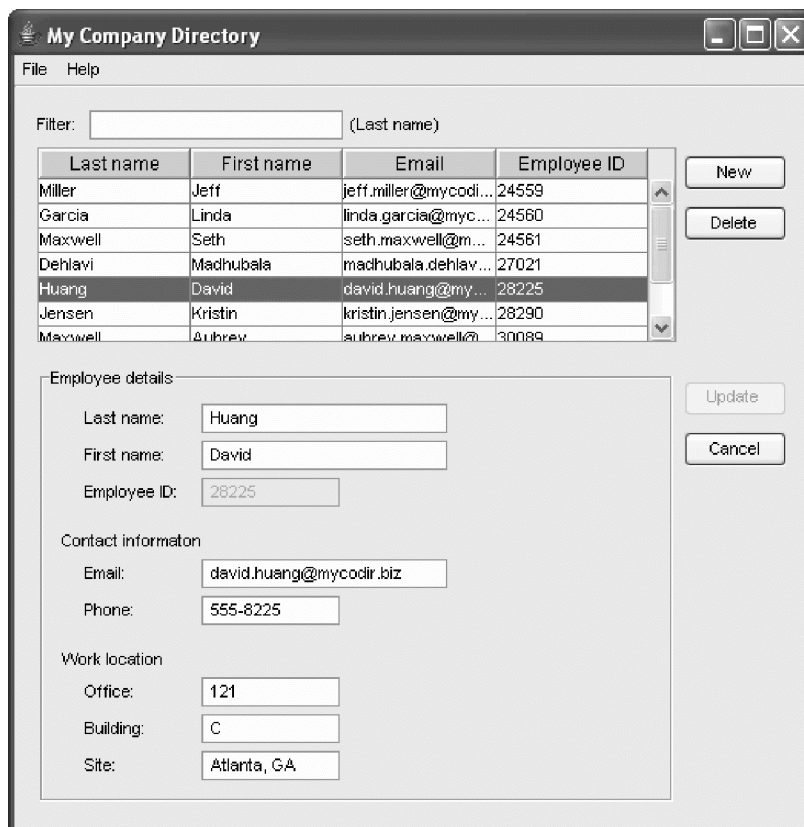
V této lekci jste se naučili, jak pro tabulku nastavit filtr.

Nyní, když spustíte aplikaci My Company Directory, můžete zadat znaky do pole **Filtr** a tabulka se bude filtrovat a zobrazí pouze ty řádky, kde příjmení obsahuje uvedené znaky.

Blahopřejeme! Aplikace My Company Directory je dokončena.

Souhrn: Vystavění klienta Rich Java Client, který používá webovou službu

Blahopřejeme! Naučili jste se, jak používat vizuální editor Javy pro vystavění aplikace My Company Directory, klienta Rich Java Client, který se připojuje k ukázkové webové službě a spravuje adresář zaměstnanců.



Viz obrázek dokončeného produktu:

Probrané lekce

Použili jste vizuální editor pro úpravu grafického uživatelského rozhraní a GridBagLayout pro uspořádání tabulky zaměstnanců. Poté jste svázali tabulku, pole a tlačítka s příslušnými datovými objekty a zdroji dat, aby aplikace pracovala s proxy prostředím Java webové služby, které jste vytvořili. Také jste provedli komplexní kódování, aby aplikace řádně fungovala a její použití bylo snadné a intuitivní. Poté jste se naučili, jak instalovat podnikovou aplikaci na server WebSphere Application Server v6.0 a jak implementovat webovou službu.

Především jste se naučili vše o výkonné třídě vázacího programu poskytované vizuálním editorem Javy pro práci s daty. Nyní jste připraveni začít sami experimentovat a umísťovat vázací programy pro nová a zajímavá využití.

Měli byste nyní být schopni provést následující úlohy:

- Použít vizuální editor Javy pro návrh komponent v GridBagLayout.
- Spustit vizuální třídu jako objekt bean Javy.

- Svázat vizuální komponenty Javy s metodami a datovými objekty vrácenými webovou službou.
- Přidat události do vizuálních komponent.

Další prostředky

Import dokončené verze do aplikace My Directory

Tento projekt zahrnuje dokončenou aplikaci, balík jve.generated s třídami vázacího programu a klienta Java webových služeb konfigurovaného pro server WebSphere Application Server v6.1. Pokud nainportujete tento dokončený projekt, aniž byste prošli výukový program, budete muset nakonfigurovat proměnnou cesty sestavení Java. Musí ukazovat na soubor JAR klienta Thin Client webových služeb serveru WebSphere v6.1.

Tip: Pokud neuvedete jiný název projektu během importu, tato akce přepíše obsah projektu MyDirectory.