ADOBE FLEX 4 教程



© 2009 Adobe Systems Incorporated. All rights reserved.

Adobe[®] Flex[®] 4 教程。预发行版。

This prerelease version of the Software may not contain trademark and copyright notices that will appear in the commercially available version of the Software.

Adobe, the Adobe logo, Flash, Flash Builder, Flex, Flex Builder and LiveCycle are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. ActiveX and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Apple and Macintosh are trademarks of Apple Inc., registered in the United States and other countries. Linux is a registered trademarks of Linus Torvalds. Microsoft and Windows are either registered trademarks of Microsoft Corporation in the United States and/or other countries. Solaris is a registered trademark of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.

This Work is licensed under the Creative Commons Attribution Non-Commercial 3.0 License. To view a copy of this license, visit <u>http://creative-commons.org/licenses/by-nc-sa/3.0/</u>

Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA.

目录

第1章:导入服务并将数据绑定到应用程序组件

创建应用程序1
创建返回本地化字符串的 ColdFusion 服务
(可选) 测试 ColdFusion 服务
连接 ColdFusion 服务
检查生成的代码
第2章:安装 Flash Builder 教程数据库
下载并安装 Flash Builder 教程数据库
第3章:为客户端应用程序创建 ColdFusion 服务
创建访问数据库的 CFC
(可选)测试服务
将服务导入到 Flex 并将返回的数据绑定到应用程序组件17
第4章:为客户端应用程序创建 PHP 服务
创建实现服务的 PHP 类
(可选) 测试 PHP 服务
将服务导入到 Flex 并将返回的数据绑定到应用程序组件
第5章:通过分页管理对数据的访问
分页教程 (ColdFusion)
分页教程 (PHP)
第6章:使用数据管理同步服务器更新
数据管理教程 (ColdFusion)

| iii

第1章:导入服务并将数据绑定到应用程序组件

Flex 中客户端代码和服务器代码间的关系与服务器模板中的传统关系不同。

在传统的关系中,服务器模板将服务器代码与客户端代码混合在一起。当客户端查询数据库时,服务器模板会将返回的数据动态地嵌入 HTML 代码。

但是, Flex 能将客户端代码与服务器代码分开。远程服务仅返回数据。Flex 会将所返回的数据绑定到客户端应用程序中的用户 界面组件。

本教程将说明如何创建导入 ColdFusion 服务的客户端应用程序。首先,创建一个显示静态文本的 Label 项。接下来,创建返 回本地化文本字符串的 ColdFusion 服务。然后,将从 ColdFusion 服务返回的数据绑定到该 Label 项。最后,检查生成的代 码,以查看 Flex 如何访问从服务返回的数据。

创建应用程序

- 1 在 Flash Builder 中,选择"文件">"新建">"Flex 项目"。
- 2 将项目名称指定为"HelloCF",将"应用程序服务器类型"设置为 ColdFusion,并指定 ColdFusion Flash Remoting。

🖻 新建 Flex 项目
创建 Flex 項目。 选择项目的名称和位置,并配置项目将使用的服务器技术。
项目名 (2): PHF_Service 项目位置 ⑦使用默认位置 (2) 文件来: C:\Documents and Settings\genovese\Adobe Flash Builder 4\P] 浏览 (2) 应用程序类型 ④ ② Web(在 Adobe Flash Player 中运行) (2) ● ② 集面(在 Adobe AIR 中运行) (2) Plex SDK 版本 ④ 使用默认 SDK(当前为 "Flex 4.0") (2) ● 使用标定的 SDK(2): Flex 4.0 ♥ Flex 4.0 豪勞 Adobe Flash Player 10. 服务器技术 应用程序服务器类型 (2): PHF ♥ 使用远程对象访问服务 (2) □ LiveCycle Data Services ES (2) ● BlareDS (2) ○ ColdFusion Flash Remoting (2)
(上一步 ®) 下一步 @) 完成 ®) 取消

指定项目名称和服务器类型

- 3 单击"下一步"。验证 ColdFusion 服务器配置并单击"验证配置"。
- 4 单击"完成"。

Flash Builder 源代码编辑器将打开 Hello.mxml。

- 5 如果未选择设计模式,请选择"设计",以切换到编辑器的设计模式。
- 6 在"组件"视图中,请确保"控件"文件夹已打开。滚动查看控件,找到"Label"控件。



" 控件" 文件夹中的"Label" 控件

- 7 将"Label"组件从"组件"视图拖动到设计区域。
- 8 双击"Label"组件。将默认文本替换为 Hello。
- 9 在"属性"视图中,修改文本的外观。选中"Label"组件后,请指定下列选项:

字体大小	24
字体属性	粗体
字体属性	斜体

FLEX 教程 3 导入服务并将数据绑定到应用程序组件

▶ _■ 属性 🗙	🕄 外观			Az
A s:Label				
▼ 公共				^
ID:	label2			
所处状态:	所有状态			
文本:	Hello			
▼ 样式				
样式: ③	(认样式>		 ✓ A - 	
	(t+t+++			
	转换闪			
		<u></u>		
Arial	×	24 36		
TTT	Ŧ	E : [:		=
		背景		
颜色:		Alp:	100% 💌	
		颜色		
主色:				
选择:		6		
鼠标移入:			_	
选择字体属性	ŧ			

10 单击"字体颜色"框,然后指定 #ff0000,将文本更改为红色。

	🛌 屈性 🗙 🍪 外观	
	A s:Label	
	- ☆ ☆ 土	^
<u>新色选取器</u> 选择颜色:		▲ R: 255 G: 0 B: 0 # ff0000 ■ ■ ■

"字体颜色"框

11 保存文件, 然后单击"运行"按钮运行该应用程序。



"运行"按钮

应用程序将在 Web 浏览器中运行。

http://localhost:8888/HelloCF-debug/HelloCF.	.html - Microsoft I 🔳 🗖 🔀
文件(亚)编辑(亚)查看(V) 收藏(A) 工具(亚) 帮助(H)	22
🔇 后退 🔹 🐑 🔹 🛃 🏠 🔎 搜索 🌟 收藏夹	🚱 🔗 🍓 🖻 🚳
地址 ① 🚳 http://localhost:8888/HelloCF-debug/HelloCF.html	▶ 封到 链接 >>
Hello	
② 完毕	Section 1.

创建返回本地化字符串的 ColdFusion 服务

创建一个可以让 Flex 通过 Internet 作为服务访问的 ColdFusion 组件 (CFC)。

1 在 Web 根目录中, 创建以下目录:

<CF Web Root>/HelloCF/

2 编写以下 HelloService.cfc 代码,并将其放在 HelloCF 目录中:

```
<cfcomponent>
   <cffunction name="getMessage" access="remote" returntype="string">
        <cfset message = "Hello from CF!">
        <cfreturn message>
    </cffunction>
    <cffunction name="getLocalizedMessage" returnType="string" access="remote">
        <cfargument name="locale" required="yes"/>
        <cfswitch expression="#locale#">
            <cfcase value="en">
                <cfset message = "Hello from CF!"/>
            </cfcase>
            <cfcase value="es">
                <cfset message = "Hola de CF!"/>
            </cfcase>
            <cfcase value="ne">
                <cfset message = "Namaskar CF bata!"/>
            </cfcase>
            <cfdefaultcase>
                <cfthrow message="Unknown locale code"/>
            </cfdefaultcase>
        </cfswitch>
        <cfreturn message>
    </cffunction>
    <cffunction name="getLocales" returnType="array" access="remote">
        <cfset codes = listToArray("en,es,ne")/>
        <cfreturn codes>
    </cffunction>
</cfcomponent>
函数 getMessage() 是客户端应用程序访问的一个服务操作。
```

```
请确保将 remote 指定为该函数的访问类型。
```

(可选)测试 ColdFusion 服务

在从应用程序访问服务之前,可通过多种方法对服务进行测试。

1 创建 ColdFusion 脚本文件 helloservicetest.cfm,该文件将调用服务并以 HTML 格式显示返回的数据。

```
使用 cfinvoke 标签调用该服务。
```

使用 cfdump 标签查看结果。

```
Testing getLocales()...
<cfinvoke component="HelloService" method="getLocales" returnvariable="locales"/>
 Result: <cfdump var="#locales#"/>
```

```
Testing getLocalizedMessage("es")...
<cfinvoke component="HelloService" method="getLocalizedMessage" locale="es" returnvariable="message"/>
 Result: <cfdump var="#message#"/>
```

2 将 helloservicetest.cfm 保存在 HelloService.cfc 所在的文件夹中, 然后从 Web 浏览器调用脚本。

http://localhost:8500/HelloCF/helloservicetest.cfm

🗿 http://localhost:8500/HelloCF/helloservicetest.cfm - Microsof 🌅	
文件 (2) 编辑 (2) 查看 (Y) 收藏 (A) 工具 (1) 帮助 (H)	1
😋 后退 🔹 🕥 - 💌 😰 🚮 🔎 搜索 🌟 收藏夹 🧐 🔗 - 🌺 🖅 - 🛄	»
地址 (1) 🍓 http://localhost:8500/HelloCF/helloservicetest.cfm 🛛 💙 🔁 转到	链接 »
Testing getLocales() Result: array 1 en 2 es 3 ne	
Testing getLocalizedMessage("es")	
Result: Hola de CF!	
	V

使用 cfinvoke 和 cfdump

如果服务中有错误, ColdFusion 会显示关于该错误的详细信息。



您还可以启用 ColdFusion 服务器的"健壮异常信息"以获得有关调试输出的详细信息。

3 向服务中添加跟踪语句。

在 ColdFusion 管理器中, 启用调试和记录。

必要时添加 ColdFusion 跟踪语句,对服务进行调试。例如,在 helloservicetest.cfm 中添加以下行:

```
Testing getLocalizedMessage()...
<cfinvoke component="HelloService" method="getLocalizedMessage" locale="es" returnvariable="message"/>
 Result: <cfdump var="#message#"/> 
<cftrace category="getLocalizedMessage End" inline="yes" var="message" text="GetLocalizedMessage
call has completed">
```

从 Web 浏览器调用脚本可查看调试输出。您还可以检查 ColdFusion 日志文件。



直接通过 URL 测试服务

4 直接通过 URL 调用服务中的函数。

例如,调用 getLocalizedMessage()时指定参数 es:

http://localhost:8500/HelloCF/HelloService.cfc?method=getLocalizedMessage&locale=es

注:也可以在将服务导入 Flash Builder 之后再测试服务。可以使用"数据 / 服务"视图中的"测试操作"。

连接 ColdFusion 服务

在 Flash Builder 中,可以连接 ColdFusion 服务,并将从该服务返回的数据绑定到 Label 组件。

- 1 在MXML编辑器的"设计"模式下,确保 HelloCF.mxml 已打开。从 Flash Builder 的"数据"菜单中,选择"连接数据/服务"。
- 2 选择"ColdFusion"作为服务类型,然后单击"下一步"。

□ 连接数据/展 选择服务类型	务HelloCF 的			
选择要连接的服务	类型。			-
服务是指『lex 说 查看有关如何创建	问服务器数据的; 和使用服务的更;	方式。 多 <u>信息。</u>		
CF	HTTP	WSDL	XML	
ColdFusion	HTTP	Web 服务	XML	

3 单击"浏览",然后导航到先前创建的 HelloService.cfc。 Flash Builder 将服务命名为 HelloService。

匝 连接数据/	服务HelloCF 的	
配置 ColdFus 指定 ColdFusi	ion 服务 on CFC 文件的位置和服务名称。	* *
CFC 位置 (L): 服务详细信息 服务名称: 服务包: 数据类型包:	C:\ColdFusion8\wwwroot\HelloCF\HelloService.cfc 如果您没有现有 CFC,這单击此处生成一个示例。 HelloService services.helloservice valueObjects	洲荒 @))
?	(上一步復))下一步(虹)) 完成(2)	取消

导入 ColdFusion 服务

4 单击"完成"。提供您的凭据以登录到 ColdFusion 数据库。

"数据/服务"视图现在将显示 ColdFusion 服务。

🛃 问题 🧠 数据/服务 🖂							
	<u>66</u>	°0	∵∷ °☆	જે જૈ	🐝 🚮	6	\bigtriangledown
 ● HelloService 数据类型 ● getLocales(): ▲ getMessage(): 	Object ssage(] String	:[] .ocal:	≗ : OЪj	ect) : St:	ring		

"数据/服务"视图

◎ "数据/服务"视图会提供代表各种远程服务的图片。在为服务调用返回的数据配置类型时,该视图会非常有用。

- 5 选择设计区域中的 Label 组件。从 Label 组件的上下文菜单中,选择"绑定到数据"按钮。
- 6 选择"确定",将当前文本替换为从服务操作返回的文本。

♀ Flash Builder 提供了各种方法来将数据绑定到所选组件。除"属性"视图以外,还可以从上下文菜单绑定到数据。此外,也可以从 Flash Builder 的"数据"菜单中选择"绑定到数据"。

7 在"绑定到数据"对话框中,请确保选择了"新服务调用"。选择 getLocalizedMessage 作为"操作"。单击"确定"。

Fe 绑定到数据	
要调用的操作 ③新服务调用	
服务:	HelloService 💌
操作:	getLocalizedMessage(locale : Objec 💌
○ 现有调用结果:	更改派回类型
文本提供程序:	
	确定 取消

选择用于绑定数据的操作

Flash Builder 将编辑器切换到源代码模式,并加亮 getLocalizedMessage()的参数。

8 键入 "es" 作为 getLocalizedMessage() 的参数值。切换到编辑器的设计模式。



9 保存并运行该应用程序。

Label 组件显示从 HelloService 检索的消息。

http://localhost:8500/HelloCF-debug/HelloCF.html	- Microsoft Internet Exp 🔳 🗖 🗙
文件 (2) 编辑 (2) 查看 (2) 收藏 (4) 工具 (2) 帮助 (4)	
🔇 后退 🔹 🕥 🕘 🔀 💋 換索 🌪 收藏夹 🊱	🔗 · 🌺 📧 · 🔜 👰 🏭 🦓
地址 ① 💰 http://localhost:8500/HelloCF-debug/HelloCF.html	▶ 转到 链接 》
Hola de CF!	
	<u>×</u>
el 完毕	🧾 😼 本地 Intranet 🛒

检查生成的代码

将 ColdFusion 服务导入到客户端应用程序时, Flex 生成了可访问 ColdFusion 服务的客户端代码。将操作绑定到应用程序组件时, Flex 也会生成代码。

生成的代码包括三个部分:

- 事件处理函数
- 服务调用和响应程序
- 数据绑定

本部分教程将带您了解前述教程中生成的 Flex 代码。在本教程中,本部分没有"步骤",仅仅是对代码生成的描述。

事件处理函数

Flex 使用事件处理函数来触发服务调用。

Flex 代码不是像在服务器模板中那样被"自上向下"处理的,而是由 Flex 构建的应用程序对事件进行侦听,并触发响应。

默认情况下, Flex 使用组件上的 creationComplete 事件触发服务调用。

在 HelloCF.mxml 源代码中检查 Label 组件:

```
<s:Label x="70" y="60"
    text="{getLocalizedMessageResult.lastResult}"
    fontWeight="bold" fontSize="24" fontStyle="italic"
    color="#FF0000" id="label"
    creationComplete="label_creationCompleteHandler(event)"/>
```

应用程序创建组件后,将触发 creationComplete 事件。对于该 Label 组件, creationComplete 调用 Flash Builder 为该事件生成 的事件处理函数。在脚本块中检查该事件处理函数的代码:

```
<fr:Script>

<![CDATA[

import mx.events.FlexEvent;

import mx.controls.Alert;

protected function label_creationCompleteHandler(event:FlexEvent):void

{

getLocalizedMessageResult.token = helloService.getLocalizedMessage("es");

}

]]>

</fx:Script>
```

creationComplete 事件处理函数从 ColdFusion 服务调用 getLocalizedMessage() 操作。它传递先前在源代码中指定的参数 "es"。

可以使用其它事件 (如按钮上的 Click 事件或列表的 selectionChange 事件) 触发服务调用。

服务调用和响应程序

添加服务时, Flex 会创建一个包含调用服务操作的方法的组件。调用其中一个方法时,该组件会通过使用适当协议的网络调用 相应的服务操作。

服务调用与 CallResponder 相关联。从服务调用返回的数据会放到 CallResponder 的 lastResult 属性中。通过 CallResponder,可以异步操作服务调用。处理服务调用的过程中,客户端应用程序能继续作出响应。

检查生成的服务调用和响应程序:

```
<fx:Declarations>
<fx:Declarations>
<fs:CallResponder id="getLocalizedMessageResult"/>
<helloservice:HelloService id="helloService"
fault="Alert.show(event.fault.faultString + '\n' + event.fault.faultDetail)"
showBusyCursor="true"/>
</fw.Declarations;
```

</fx:Declarations>

导入 HelloService 服务时, Flex 创建了 HelloService 类。该类包含调用该服务的 getLocales() 和 getLocalizedMessage 操作的 方法。

通过 id (设置为 getLocalizedMessageResult)标识的 CallResponder,可以访问返回的数据。

数据绑定

通过 Flex 数据绑定,可以将 CallResponder 的 lastResult 属性的值指定给某个用户界面组件。当最后的结果更新时, Flex 会 自动更新该组件。

Flex 使用花括号 {} 将数据绑定到组件属性。在以下示例中, Flex 将 lastResult 值绑定到 Label 组件的 text 属性。

检查 Label 组件的数据绑定:

```
<s:Label x="70" y="60"
```

```
text="{getLocalizedMessageResult.lastResult}"
fontWeight="bold" fontSize="24" fontStyle="italic"
color="red" id="richEditableText"
creationComplete="label_creationCompleteHandler(event)"/>
```

返回到 CallResponder 的数据 getLocalizedMessageResult.lastResult 被自动指定为 Label 组件的 text 属性。

代码生成概述

总之, Flash Builder 能够生成访问在客户端应用程序中导入的服务的代码:

- Flex 在运行中的应用程序中创建 Label 组件之后,将触发 creationComplete 事件并调用服务操作。
- 从该服务调用返回的数据将放在相关联的 CallResponder 的 lastResult 属性中。
- 绑定到 lastResult 属性的 Label 组件的 text 属性将用返回的数据更新。

第2章:安装 Flash Builder 教程数据库

有几个教程说明了如何访问从数据库返回的数据。这些教程使用 MySQL 数据库,您可以在 PHP 或 ColdFusion 安装中访问 该数据库。

下载并安装 Flash Builder 教程数据库

1 下载 Flash Builder 教程数据库,并将其解压到计算机上的临时目录中。

下载的 ZIP 文件包含两个 SQL 文件:

fb_tutorial_db_full.sql

该文件用于将整个数据库导入到 MySQL 安装中。它创建 fb_tutorial_db 数据库,其中包含三个表,即, departments、 employees 和 dept_emp。

fb_tutorial_db_tables.sql

该文件用于只将三个表导入到现有数据库中。在您无权在 MySQL 安装中创建数据库的情况下,可以使用该文件。有关导入该文件的信息,请参阅自述文件。

可以从命令行安装数据库,也可以通过 MySQL 数据库的 phpMyAdmin 实用程序进行安装。

2 (命令行)在操作系统的终端窗口上执行以下命令。

<MySQL_Install_Dir>/bin/mysql -u root -p < fb_tutorial_db_full.sql

- 3 (phpMyAdmin)从 phpMyAdmin 主页导航到"导入"页面。浏览到下载的 SDL 文件,并导入数据库。
- 4 (ColdFusion) 在 ColdFusion 管理器中, 将 fb_tutorial_db 添加为数据源。

下表列出了使用 ColdFusion 8 默认配置的情况下数据源的设置。

配置字段	值
ColdFusion 数据源名称	fb_tutorial_db
数据库	fb_tutorial_db
服务器	127.0.0.1
端口	3306

第3章:为客户端应用程序创建 ColdFusion 服务

访问远程服务时,通常会从数据库检索记录,然后将它们显示在客户端应用程序中。本教程将说明如何访问远程数据库,并用 DataGrid 显示检索的记录。

🕘 ht t	p://localhost:8500/books/Books	s.html - Microsoft Internet H	ixplorer	
文件 @) 编辑(E) 查看(Y) 收藏(A) 工具(T)) 帮助任)		1
<u>ج</u>	腿 🔹 🛞 🔹 🚺 💋 攖	索 🥎 收藏夹 🚱 🔗 - 嫨 🚾	- 📃 👰 🛍	28
地址①	🕘 🚳 http://localhost:8500/books/Books. 1	html	💙 🔁 转	到链接»
				<u>~</u>
	Genre	Title		
	Fiction	The Road	A	
	Non-fiction	Shopping Mart Mania		
	Non-fiction	Beer Googles		
	Non-fiction	The Living Portrait		
	Non-fiction	All American		
	Fiction	The Look of Innovation	v	
ľ í				
L				~
ど 完毕			🧐 本地 Intrar	et 📑

本教程将阐述下列概念:

- 使用 ColdFusion 组件 (CFC) 实现查询数据库的服务操作。
- 将 CFC 服务操作绑定到 Flex 数据控件,如 DataGrid。
- 对从服务操作返回的数据进行取样,以便为返回的数据配置数据类型。
- 运行应用程序,并在 DataGrid 中查看返回的数据。

创建访问数据库的 CFC

本教程将访问随 ColdFusion 安装一起提供的 cfbookclub 数据库。它列出了该数据库中 Books 表的内容。但是,您可以将该 教程修改为从您可以访问的任何数据库选择记录。

本教程假定您在本地安装了 ColdFusion 8 或 ColdFusion 9, 且 ColdFusion 服务器正在运行。

- 1 在 Web 根文件夹中, 创建一个名为 books 的文件夹。
- 2 在您常用的 ColdFusion 编辑器中,创建以下 CFC。将该 CFC 命名为 BookService.cfc, 然后将其放在 Web 根文件夹的 books 文件夹中。

BookService.cfc 将选择 cfbookclub 数据库中 Books 表的所有记录。

```
<!---

This sample service contains functions that illustrate typical service operations.

This code is for prototyping only.

Authenticate the user prior to allowing them to call these methods. You can find more

information at http://www.adobe.com/go/cf9_usersecurity

---->

<cffunction name="getBooks" output="false" access="remote" returntype="any" >

        <!--- Retrieve set of records and return them as a query or array. ---->

        <!--- Add authorization or any logical checks for secure access to your data ---->

        <cfset var qAllBooks="">

        <cfquery name="qAllBooks" datasource="cfbookclub">

        SELECT * FROM Books

        </cfquery>

        <cfreturn qAllBooks>

</cffunction>
```

3 在 BookService.cfc 中,检查调用数据库的代码:

```
Flex 期望该操作返回的数据为对象或对象数组。数组中的每个对象都对应于从数据库中检索的一个记录,并包含记录中每一列的值。
```

名为 qAllBooks 的 cfquery 访问 cfbookclub 数据库中的 Books 表。 ColdFusion 从数据库返回记录数组。

(可选) 测试服务

最好从应用程序访问服务之前,先对该服务进行测试。

1 创建 bookservicetest.cfm 以测试 CFC。该脚本使用 cfdump 查看服务操作的结果。

```
Testing getBooks()...
<cfinvoke component="BookService" method="getBooks" returnvariable="books"/>
 <cfdump var="#books#"/>
```

- **2** 将 bookservicetest.cfm 保存在 BookService.cfc 所在的 books 文件夹中。
- 3 从 Web 浏览器运行该脚本, 以对服务进行测试:

http://localhost:8500/books/bookservicetest.cfm

🕘 http://local	ho	st:8500/B	ooks/bookservicetest.cfm - Microsoft Internet Expl	orer								
文件(正) 编辑(正)	文件 (2) 编辑 (2) 查看 (2) 收藏 (4) 工具 (1) 帮助 (1)											
🔇 后退 🔹 🕥	🔾 后退 🔹 📀 - 💌 🛃 🏠 🔎 搜索 🧙 收藏夹 🤣 🔗 - 🌽 🖻 - 🛄 🌌 🏭 🦀											
地址 @) 🙆 http://	地址 (1) 🧉 http://localhost:8500/Books/bookservicetest.cfm											
Testing getBooks	Testing getBooks()											
que ry												
RESULTSET	qu	ery										
		AUTHORID	BOOKDESCRIPTION	BOOKID	BOOKIMAGE	GENRE	ISSPO					
	1	1	A trucker's memoir. Poignant scenes showing a man who changes for better and worse while driving in every state of the US.	1	book01.jpg	Fiction	N					
	2	2	The seductions of shopping. The true confessions of a shopaholic, with two how-to guides: one on maximizing your shopping experiences, and other a self-help guide on controlling your shopping urges.	2	book02.jpg	Non- fiction	Ŷ					
	3	3	A falling-off-your-chair funny book about the dangers of mixing beer with a Web 2.0 lifestyle. Rob Roberts makes all the mistakes a man can make in love, work, the universe, and everything, yet comes out, well, find out for yourself.	3	book14.jpg	Non- fiction	N					
	4	4	How to draw portraits from life and make them lifelike! Covers all traditional art media, but focuses on oil and pastels.	4	book03.jpg	Non- fiction	N					
	5	5	The Justin Thyme story. How a man with limited intelligence can make it big in America by obsessively following a narrow path.	5	book04.jpg	Non- fiction	Y					
	6	6	How one woman struggles to make artistic designs that effectively sell new products. Is it fiction, or is it fact? You be the judge.	6	book05.jpg	Fiction	Y					
7 7 Chick-lit at its best. The story of Wanda Lyfe, an ugly duckling on the joutside, as wan on the inside, and how she learns to let her inner being make the best of the traumas of being a teenager in the big, heartless, city. 7 8 8 A ghost story for the ages. Harrowing events and eerie doings galore of start when the Jones-Smith family moves into the gothic mansion left them by their barely remembered great uncle Ebenezer. 8 8 book06.jpg							N					
							Y					
	9	9	Amateur softball's greatest. Stories of all the great amateurs of today and yesteryear.	9	book08.jpg	Non- fiction	Y					
2 完毕	140	4.0		140	1.001	1 er 11						

○ 也可以在将服务导入 Flash Builder 之后再测试此服务。可以使用"数据 / 服务"视图中的"测试操作"。

将服务导入到 Flex 并将返回的数据绑定到应用程序组件

Flex 中客户端代码和服务器代码间的关系与服务器模板中的传统关系不同。

在传统的关系中,服务器模板将服务器代码与客户端代码混合在一起。当客户端查询数据库时,服务器模板会将返回的数据动态地嵌入 HTML 代码。

但是, Flex 能将客户端代码与服务器代码分开。远程服务仅返回数据。Flex 会将所返回的数据绑定到客户端应用程序中的用户 界面组件。

1 在 Flash Builder 中,选择"新建">"Flex 项目",创建一个 ColdFusion 服务器项目。将该项目命名为 Books。

将"应用程序服务器类型"设置为"ColdFusion"。启用"使用远程对象访问服务"并选择"ColdFusion Flash Remoting"。

🖻 新建 Flex 项目 📃 🗖 🔀
创建 Flex 項目。 选择项目的名称和位置,并配置项目将使用的服务器技术。
「項目名 @): Pooks 「項目位置 「使用默认位置 @) 文件来: [:\Documents and Settings\genovese\Adobe Flash Builder 4\Br, 浏览 @) 应用程序类型 ④ @ Web(在 Adobe Flash Player 中运行) @) ⑥ @ Web(在 Adobe Flash Player 中运行) @) ⑦ @ 桌面(在 Adobe AIR 中运行) @) ⑦ @ ↓ @ ↓ @ ↓ @ ↓ ↓ @ ↓ ↓ ↓ @ ↓ ↓ ↓ ↓
(上一步 18) 下一步 00) 完成 (2) 取消

- 2 单击"下一步"。验证配置。
- 3 将输出文件夹设置为包含服务的 books 文件夹。单击"完成"。

Flash Builder 将创建 Books.mxml,并在 MXML 编辑器中打开该文件。选择"设计"以在设计模式下打开该编辑器。

4 在设计模式下,将 DataGrid 控件拖动到设计区域。

DataGrid 控件列在"组件"视图中的"控件"下。

- 5 从 DataGrid 的上下文菜单中,选择"绑定到数据"。
- 6 在"未定义服务"对话框中,单击"是"以连接一个服务。
- 7 在"连接数据 / 服务"向导中,选择"ColdFusion"。单击"下一步"。

Flash Builder 提供了多种连接数据服务的方法。在本案例中,首先创建用户界面,然后从用户界面组件连接服务,并 指定远程操作。

8 单击"浏览",然后导航到先前创建的 BookService.cfc。单击"完成"。

根据需要为系统提供授权凭据。

"数据/服务"视图现在会显示"BookService"。

9 再次从 DataGrid 的上下文菜单中,选择"绑定到数据"。 此时将打开"绑定到数据"对话框,且"新服务调用"处于选中状态。 BookService 是该 Flex 项目中唯一可用的服务。

getBooks()是该服务中唯一可用的操作。

匝 绑定到数据	
要调用的操作 ③新服务调用	
服务:	BookService 💌
操作:	getBooks() : Object 💌
	配置返回类型 (必需)
○现有调用结果:	
数据提供程序:	
🙆 诸配置有效的返	回类型并继续。
	确定 取消

10 在"绑定到数据"对话框中,选择"配置返回类型",为返回的数据定义数据类型。

Flex 使用返回数据类型访问服务操作。BookService 服务不为返回的数据定义数据类型。Flash Builder 使用客户端键入为 返回的数据定义自定义数据类型。

♀ 某些服务 (如 LiveCycle Data Services 提供的服务)提供服务器端键入。使用服务器端键入, Flash Builder 不必为 返回的数据配置数据类型。

11 在"配置返回类型"对话框中,默认情况下选中"自动检测返回类型"。单击"下一步"。

Flash Builder 内部检查服务,以确定返回的数据和数据类型。

12 在"以下类型的数组"字段中指定 Book,为返回的数据定义自定义类型。

BookService 从数据服务返回记录数组。每个记录都是一个复杂的数据类型,表示书籍的数据库记录。通过自定义类型 Book,可以访问记录的每个字段。

"配置返回类型"对话框显示服务返回的数据类型的属性。单击"完成"。

♀ 当 Flash Builder 配置返回类型时,它会访问数据库以创建一个值对象。自定义数据类型的属性将从该值对象派生。可以在继续操作之前查看数据类型的属性。

13 在"绑定到数据"对话框中,单击"确定"。

Flash Builder 将从服务调用返回的数据绑定到 DataGrid 组件。它可修改 DataGrid 的列,将为每个 Book 属性返回的值 绑定到 DataGrid 中的一个列。

🐻 *Books	Books. mxml 🔀									
☑ 源代码	i 🔚 igit 🛛 🜔 🛛	🖥 状态: 〈Stat	e1> 🔽		k	🖑 🔍 100% 🔽				
	Ø									
	THUMBNAILIMA(ISSPOTLIGHT	BOOKID	BOOKDESCRIP	GENRE	AUTHORID				
	9									
	L			l(J	=				

14 请确保 DataGrid 仍处于选中状态。在"属性"视图中,单击"配置列",然后执行以下步骤:

- a 选择 ISSPOTLIGHT 列。单击"删除"以删除该列。
- **b** 删除 TITLE 和 GENRE 以外的所有列。
- c 选择 TITLE 列。编辑"头文本"字段,将该列重命名为 Title。
- d 将 GENRE 列重命名为 Genre。
- e 选中 GENRE 列, 然后单击"向上"以对列重新排序。
- **f** 单击"确定"。

■ 配置列		
Genre Title	添加 删除 向上 向下	新建調整 ● 标准调整 ● 标准调整 ● 标准调整 ● 「「「小小小小小小小小小小小小小小小小小小小小小小小小小小小小小小小小小小
		文本格式 Arial ▼ 12 T T T T T 目动換行
•		确定 取消

15 在设计视图中, 捕捉 DataGrid 最左侧的操作柄, 然后将 DataGrid 大小调整为更标准的形状。保存并运行该应用程序。



第4章:为客户端应用程序创建 PHP 服务

您可以使用 Flex 创建可通过各种技术访问服务的应用程序。这些技术包括:

- ColdFusion
- PHP
- HTML (REST 样式) 服务
- Web 服务 (SOAP)

创建服务后,最好先对其进行测试,然后再从客户端应用程序访问。

PHP 服务的创建和测试原理可适用于其它服务技术。

🥙 h	ttp://localh	ost:8888/PHF	Service/P	HP_Service	e. html	L – I	icros	oft	Intern	et l	ix			K
文件	・(F) 编辑(E)	查看(V) 收藏(A) 工具(1)	帮助(出)										1
G	后退 - 🕥 -	💌 🖻 🎸) 🔎 搜索	🥎 收藏夹	Ø	8.	5	IW	• 📃	M	í۵,	28		
地址	🛈 🙆 http://le	ocalhost:8888/PH	PService/PHP	_Service.html						~	🔁 转	到	链接	»
													1	
	ID	First Name	Last Name											
	10001	Georgi	Facello	A										
	10002	Bezalel	Simmel											
	10003	Parto	Bamford											
	10004	Chirstian	Koblick											
	10005	Kyoichi	Maliniak											
	10006	Anneke	Preusig	v										
													1	e.
2 完	毕								9	本地	Intra	net		.::

创建实现服务的 PHP 类

在本教程中创建的服务可访问 fb_tutorial_db 数据库。有关安装该数据库的信息,请参阅第 14 页的"安装 Flash Builder 教程数据库"。

- 1 在Web根文件夹中,创建一个名为PHPService的文件夹。在该文件夹内,再创建一个名为 services 的文件夹。
- 2 在您常用的 PHP 编辑器中,创建以下 PHP 文件以实现 EmployeeService。将该文件命名为 EmployeeService.php,并将 其放在 PHPService/services 文件夹中。

```
<?php
/**
* This sample service contains functions that illustrate typical service operations.
* This code is for prototyping only.
 \star\, Authenticate users before allowing them to call these methods.
*/
/**
* The constructor initializes the connection to database. Everytime a request is
 * received by Zend AMF, an instance of the service class is created and then the
 * requested method is called.
 */
class EmployeeService {
 var $username = "USERNAME";
 var $password = "PASSWORD";
 var $server = "localhost";
 var $port = "3306";
 var $databasename = "fb_tutorial_db";
 var $tablename = "employees";
 var $connection;
 public function construct() {
   $this->connection = mysqli_connect(
                       $this->server,
                       $this->username,
                       $this->password,
                       $this->databasename,
                       $this->port
                       );
   $this->throwExceptionOnError($this->connection);
  }
 public function getEmployees() {
    $stmt = mysqli_prepare($this->connection, "SELECT * FROM $this->tablename");
   $this->throwExceptionOnError();
   mysqli stmt execute($stmt);
   $this->throwExceptionOnError();
   $rows = array();
   mysqli_stmt_bind_result($stmt, $row->emp_no, $row->birth_date,
                $row->first_name, $row->last_name, $row->gender, $row->hire_date);
   while (mysqli stmt fetch($stmt)) {
     $rows[] = $row;
     $row = new stdClass();
     mysqli_stmt_bind_result($stmt, $row->emp_no, $row->birth_date,
                  $row->first_name, $row->last_name, $row->gender, $row->hire_date);
    }
   mysqli_stmt_free_result($stmt);
```

```
mysqli_close($this->connection);
   return $rows;
  }
  /**
  * Utitity function to throw an exception if an error occurs
  * while running a mysql command.
  */
 private function throwExceptionOnError($link = null) {
   if($link == null) {
     $link = $this->connection;
   if(mysqli_error($link)) {
     $msg = mysqli errno($link) . ": " . mysqli error($link);
     throw new Exception ('MySQL Error - '. $msq);
   }
  }
}
?>
```

该 PHP 类包含实现数据服务操作的函数。Flash Builder 内部检查 PHP 类中的服务操作,以在客户端应用程序中创建 ActionScript 类。客户端应用程序使用这些 ActionScript 类访问数据服务。

3 将 EmployeeService.php 保存在以下位置:

<Web Root>/PHPService/services/EmployeeService.php

注:请确保文件名对应于实现 PHP 服务的类的名称。

- 4 在 EmployeeService.php 中,检查调用数据库的代码。
 - 连接数据:

代码使用全局变量和连接数据库服务器的构造函数。

使用连接设置替代这些全局变量。

• 调用数据库:

```
$stmt = mysqli_prepare($this->connection, "SELECT * FROM $this->tablename");
mysqli_stmt_execute($stmt);
```

此代码用于实现对数据库的基本 PHP 调用,可返回结果集。该调用使用通过 MySQLi 扩展提供的预定义语句。预定义语句比基本数据库查询具有更高的安全性。

• 将数据返回为行数组:

Flex 期望该操作返回的数据为对象或对象数组。处理返回的结果集的 PHP 代码循环访问结果,并返回对象数组。数组 中的每个对象都对应于从数据库中检索的一个记录,并包含记录中每一列的值。

该代码将绑定结果用于预定义语句。通过此调用,结果值可以正确绑定到数据的期望值。

• 异常处理

代码创建实用程序函数以捕获异常。

(可选)测试 PHP 服务

最好从应用程序访问服务之前,先对该服务进行测试。

1 创建 PHP 脚本文件 tester.php,该文件将调用服务并显示返回的数据。

```
<?php
include('EmployeeService.php');
$0 = new EmployeeService();
var_dump($0->getemployees());
?>
```

2 将 tester.php 保存在 EmployeeService.php 所在的文件夹中, 然后从 Web 浏览器调用该服务。

```
如果调用成功, tester 转储数据库中的记录:
```

```
array(999) {
[0]=>
   object(stdClass)#4 (6) {
       ["emp_no"]=>
       string(5) "10002"
       ["birth date"]=>
       string(10) "1964-06-02"
       ["first_name"]=>
       string(7) "Bezalel"
       ["last_name"]=>
       string(6) "Simmel"
       ["gender"]=>
       string(1) "F"
       ["hire_date"]=>
       string(10) "1985-11-21"
   }
[1]=>
   object(stdClass)#5 (6) {
       ["emp no"]=>
       string(5) "10003"
       ["birth_date"]=>
       string(10) "1959-12-03"
       ["first name"]=>
       string(5) "Parto"
       ["last name"]=>
       string(7) "Bamford"
       ["gender"]=>
       string(1) "M"
       ["hire_date"]=>
       string(10) "1986-08-28"
   }
[2]=>
   object(stdClass)#6 (6) {
       ["emp_no"]=>
. . .
如果服务存在错误, PHP 会显示信息以帮助确定该错误的位置。
```

Warning: mysqli_connect() [function.mysqli-connect]: (28000/1045): Access denied for user 'admin'@'localhost' (using password: YES) in C:\wamp2\www\PHP_Service\services\PHPservice.php on line 6

Access denied for user 'admin'@'localhost' (using password: YES)

也可以在将服务导入 Flash Builder 之后再测试服务。可以使用"数据 / 服务"视图中的"测试操作"。

将服务导入到 Flex 并将返回的数据绑定到应用程序组件

Flex 中客户端代码和服务器代码间的关系与服务器模板中的传统关系不同。

在传统的关系中,服务器模板将服务器代码与客户端代码混合在一起。当客户端查询数据库时,服务器模板会将返回的数据动态地嵌入HTML代码。

但是, Flex 能将客户端代码与服务器代码分开。远程服务仅返回数据。Flex 会将所返回的数据绑定到客户端应用程序中的用户 界面组件。

- 1 在 Flash Builder 中,选择"文件">"新建">"Flex 项目"。
- 2 将项目名称指定为 "PHP_Service" 并将 "应用程序服务器类型" 设置为 "PHP"。

🅫 新建 Flex 项目
创建 Flex 项目。 选择项目的名称和位置,并配置项目将使用的服务器技术。
项目名(g): PHF_Service 项目位置 ● 使用默认位置(g) 文件来: [:\Documents and Settings\genovese\Adobe Flash Builder 4\P] 浏览(g) 应用程序类型 ● 读 Web(在 Adobe Flash Player 中运行)(g) ● 读 集面(在 Adobe AIR 中运行)(g) ● 读 集面(在 Adobe AIR 中运行)(g) ■ flex SDK 版本 ● 使用新认 SDK(当前力 "Flex 4.0")(g) ● 使用特定的 SDK(g): Flex 4.0 ♥ Flex 4.0 漸要 Adobe Flash Player 10. 服务器技术 应用程序服务器类型(g): Flex ● ● (d用指对象访问服务(g) ● (liveCycle Data Services ES(g) ● (liveCycle Data Services ES(g) ● (coldFusion Flash Remoting(g)
⑦ 〈上→步 ⑧ 下→步 ⑭〉〉 完成 ⑫ 取消

指定项目名称和服务器类型

注: 如果创建项目时忘记设置应用程序服务器类型,可以以后从项目属性页指定服务器类型。

- 3 单击"下一步"。验证 PHP 配置并单击"验证配置"。
- 4 将先前创建的 PHPService 文件夹指定为"输出文件夹"。

Flash Builder 会建议输出文件夹的默认位置。使用已包含 EmployeeService.php 的位置。以下是先前指定的位置: <Web Root>/PHPService/

5 单击"完成"。

Flash Builder 源代码编辑器将打开 PHP_Service.mxml。

6 选择"设计"以在设计模式下打开该编辑器。将 DataGrid 控件添加到应用程序:

DataGrid 组件列在"组件"视图中的"控件"下。将 DataGrid 控件拖动到设计区域。

- 7 从 DataGrid 的上下文菜单中,选择"绑定到数据"。
- 8 在"未定义服务"对话框中,单击"是"以连接一个服务。
- 9 在"新建 Flex 服务"向导中,选择"PHP"。单击"下一步"。

Flash Builder 提供了多种连接数据服务的方法。在本案例中,首先创建用户界面,然后可以从用户界面组件连接服务,并指定远程操作。

- 10 单击"浏览",然后导航到先前创建的 PHPservice.php 文件。
 - 该文件的路径为:

<Web Root>/PHP_Service/services/EmployeeService.php

单击"完成"。Flash Builder 将出现提示,您可以根据需要安装或更新 Zend Framework。

"数据/服务"视图此时显示 EmployeeService。

11 选中 DataGrid 后,再次单击"属性"视图中的"数据提供程序"按钮。

将打开"绑定到数据"对话框,且已选中了"新服务调用"。

EmployeeService 是该 Flex 项目中唯一可用的服务。

getEmployeess()是该服务中唯一可用的操作。

🖬 绑定到数据	
要调用的操作 新服务调用 服务: 操作: 	EmployeeService getEmployees() : Object 配置返回类型 (必需)
○现有调用结果:	
数据提供程序:	
🙆 诸配置有效的返	回类型并继续。
	确定 取消

12 在"绑定到数据"对话框中,选择"配置返回类型",为返回的数据定义数据类型。

Flex 需要使用返回数据类型访问服务操作。EmployeeService 服务不为返回的数据定义数据类型。Flash Builder 使用客户端键入为返回的数据定义自定义数据类型。

13 在"配置返回类型"对话框中,默认情况下选中"自动检测返回类型"。单击"下一步"。

14 指定 Employee 作为类型的名称。单击"完成"。

EmployeeService 将返回一个表示员工数据库记录的复杂数据类型。通过自定义类型 Employee,可以访问记录的每个字段。

查看服务返回的 Employee 数据类型的属性。单击"完成"。

♀ 当 Flash Builder 配置返回类型时,它会访问数据库以创建一个值对象。自定义数据类型的属性将从该值对象派生。您可以查看和修改该数据类型的属性,然后继续。

15 在"绑定到数据"对话框中,单击"确定"。

Flash Builder 将从服务调用返回的数据绑定到 DataGrid 组件。它可修改 DataGrid 的列,将为每个 Employee 属性返回 的值绑定到 DataGrid 中的一个列。

16 请确保 DataGrid 仍处于选中状态。在"属性"视图中,单击"配置列",然后执行以下步骤:

- a 选择 hire_date 列。单击"删除"以删除该列。
- **b** 使用类似的方法,删除 emp_no、 first_name 和 last_name 以外的所有列。
- c 选择 emp_no 列。单击"向上",将其移到第一个位置。
- d 在"头文本"字段中,将emp_no列重命名为ID。
- e 使用类似的方法,重命名 first_name 和 last_name 列。
- **f** 单击"确定"。

副 配置列	- 🗆 🛛
? 确定	取消

17 在 DataGrid 仍处于选中状态的情况下,在"属性"视图中,为 editable 属性指定 False。

18 选择"文件">"保存"以保存应用程序文件。然后选择"运行">"运行 PHP_Service"以运行应用程序。

应用程序将在 Web 浏览器中运行。

🗿 h	ttp://localf	10st:8888/PHP	Service/P	HP_Service	. htn	l - Nicros	oft	Internet	Ex	
文件	·(F) 编辑(E)	查看(V) 收藏(A) 工具①	帮助(H)						1
G	后退 - 🕥	🖹 🐔) 🔎 搜索	🥎 收藏夹	Ø	Ø• 🎍	W	- 📃 🐖) 🛍 👌	8
地址	🛈 🙋 http://1	ocalhost:8888/PH	PService/PHP	_Service.html				*	🔁 转到	链接 »
										<u>~</u>
	ID	First Name	Last Name							
	10001	Georgi	Facello	A						
	10002	Bezalel	Simmel							
	10003	Parto	Bamford							
	10004	Chirstian	Koblick							
	10005	Kyoichi	Maliniak							
	10006	Anneke	Preusig	v						
										_
								671 - 14 14		<u>~</u>
き完	毕							1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	, Intranet	

第5章:通过分页管理对数据的访问

由于各种性能和网络瓶颈问题,通常情况下,一次调用并不能检索数据库中的所有记录。通过分页,可以仅根据需要,以增量 方式按组检索记录。

要使用分页检索记录,请实现包含两个具有下列签名的操作的数据服务:

- getItems_paged(\$startIndex, \$numItems)
- count()

分页教程 (ColdFusion)

在本教程中创建的应用程序可访问 fb_tutorial_db 数据库。有关该数据库的安装信息,请参阅第 14 页的"安装 Flash Builder 教程数据库"。

本教程将创建一个使用分页来填充 DataGrid 的应用程序。

🙆 ht	t <mark>p://local</mark> h	ost:8500/Pagin	gCF/PagingCF	.html - Licr	osoft Intern	et Explorer		
文件 (E) 编辑(E)	查看(V) 收藏(A)	工具(T) 帮助()	Ð				-
G	🕝 后退 🔹 🔊 - 🛃 🛃 🏠 🔎 搜索 🌟 收藏夹 🤣 🔗 - 🌺 🚾 - 🔜 灯 鑬 🦓							
地址@) 🕘 http://1	ocalhost:8500/Pagin	gCF/PagingCF.htm	1			🖌 🄁 转到	链接 »
								^
	first_name	hire_date	birth_date	emp_no	last_name	gender		
	Georgi	Wed Jun 25 23:(Wed Sep 2 00:0	10001	Facello	М	A	
	Bezalel	Thu Nov 21 00:0	Tue Jun 2 00:00	10002	Simmel	F		
	Parto	Wed Aug 27 23:1	Thu Dec 3 00:00	10003	Bamford	м		
	Chirstian	Mon Dec 1 00:0(Sat May 1 00:00:	10004	Koblick	м		
	Kyoichi	Mon Sep 11 23:(Fri Jan 21 00:00	10005	Maliniak	м		
	Anneke	Thu Jun 1 23:00	Mon Apr 20 00:0	10006	Preusig	F	v	
								\sim
🙆 完박	2						站也 Intranet	

数据网格

创建远程服务并将其导入到 Flex 项目中

- 1 在 Web 根文件夹中, 创建一个名为 PagingCF 的文件夹。
- 2 在您常用的 ColdFusion 编辑器中, 创建以下 CFC。将该 CFC 命名为 PagingService.cfc, 然后将其放在 Web 根文件夹的 PagingCF 文件夹中。

PagingService.cfc 包含 Flash Builder 实现分页所需的函数。

```
<cfcomponent output="false">
<!---
This sample service contains functions that illustrate typical service operations.
This code is for prototyping only.
Authenticate the user prior to allowing them to call these methods. You can find more
information at http://www.adobe.com/go/cf9_usersecurity
--->
   <cffunction name="getItems_paged" output="false" access="remote" returntype="any" >
        <cfargument name="startIndex" type="numeric" required="true" />
        <cfargument name="numItems" type="numeric" required="true" />
<!--- The LIMIT keyword is valid for MySQL database only, modify according to your database --->
        <cfset var qRead="">
        <cfquery name="qRead" datasource="fb_tutorial_db">
           SELECT * FROM employees LIMIT #startIndex#, #numItems#
        </cfquery>
        <cfreturn qRead>
    </cffunction>
    <cffunction name="count" output="false" access="remote" returntype="numeric" >
        <cfquery name="qread" datasource="fb_tutorial_db">
            SELECT COUNT(emp_no) AS empCount FROM employees
        </cfquery>
        <cfreturn gread.empCount>
    </cffunction>
</cfcomponent>
```

- **3** 在 Flash Builder 中, 创建一个 Flex 项目。将该项目命名为 PagingCF, 并指定 ColdFusion 作为服务器技术, 如下所示。 单击"下一步"。
 - 应用程序服务器类型: ColdFusion
 - 启用"使用远程对象访问服务"
 - 选择 "ColdFusion Flash Remoting"

🖬 新建 Flex 项目
创建 Flex 項目。 选择项目的名称和位置,并配置项目将使用的服务器技术。
项目名 (p): FagingCF 项目位置 ⑦使用默认位置 (U) 文件来: [:\Documents and Settings\genovese\Adobe Flash Builder 4\F] 浏览 (B) 应用程序类型 ④ @ Web(在 Adobe Flash Flayer 中运行) (U) ● @ 桌面(在 Adobe AIR 中运行) (U) Flex SDK 版本 ④ 使用默认 SDK(当前为 "Flex 4.0") (U) ● 使用默认 SDK(当前为 "Flex 4.0") (U) ● 使用影议 SDK(S): Flex 4.0" Flez 4.0 承要 Adobe Flash Flayer 10. 服务器技术 应用程序服务器类型 (U): ColdFusion ♥ ♥ 使用远程对象访问服务 (E) ● LiveCycle Data Services ES (L) ● BlazeDS (E) ● ColdFusion Flash Remoting (E)
() () 下一步(1) 完成(1) 取消

新建 ColdFusion 项目

- 4 验证 ColdFusion 设置,并将 "PagingCF" 目录指定为"输出文件夹"。单击"完成"。
- 5 从 Flash Builder 的"数据"菜单中选择"连接数据 / 服务"。选择"ColdFusion", 然后单击"下一步"。
- 6 单击"浏览",然后导航到在步骤 2 中创建的 PagingService.cfc 文件。单击"完成"。

7 在 Flash Builder 的"数据 / 服务"视图中,从 getItems_paged()操作的上下文菜单中,选择"配置返回类型"。

🖬 新建 Flex 项目
创建 Flex 項目。 选择项目的名称和位置,并配置项目将使用的服务器技术。
项目名 @): PagingCF 项目位置 ♥使用默认位置 @) 文件夹: [:\Documents and Settings\genovese\Adobe Flash Builder 4\P, 浏览 @) 应用程序类型 ● @ Web(在 Adobe Flash Player 中运行) @) ● @ Meb(在 Adobe AIR 中运行) @) Flex SDK 版本 ● 使用默认 SDK(当前为 "Flex 4.0") @) ● 使用特定的 SDK(S): Flex 4.0") @) ■ E置 Flex SDK ● 使用特定的 SDK(S): Flex 4.0 ♥ Flex 4.0 需要 Adobe Flash Player 10. 服务器技术 应用程序服务器类型 (1): ColdFusion ♥ ● 使用远程对象访问服务 @) ● LiveCycle Data Services ES (L) ● BlazeDS @) ● ColdFusion Flash Remoting (G)
(上一步 1) 下一步 (1) 完成 10 取消

- 8 在"配置返回类型"对话框中,默认情况下选中"自动检测返回类型"。单击"下一步"。
- 9 为 getItems_paged() 操作的参数指定值和类型,如下所述。单击"下一步":

参数	参数类型	值
startIndex	Number	0
numItems	Number	10

Fe 配置返回类型						
自动检测版 回类型 输入调用此操作所需的参数值。						
为了自动检测返回类型	실, Flash Builder 裕(使用您提供的下列参数	如调用操作并分析返回的数据。 、			
探作: getItems_pag	ed(startindex : Numb)失忆光刊	ber, numltems : Numt	ber)			
startIndex numItems	Number Number					
?	(〈上─步⑭)下	━步(20)> 完成(2)	取消		

新项目配置

10 在"以下类型的数组"字段中指定 Employee,为返回的数据定义自定义类型。

Flex 使用自定义数据类型访问并更新从服务器返回的复杂数据类型。

"配置返回类型"对话框显示服务返回的数据类型的属性。

- 11 单击"完成", 配置"Employee" 返回类型。
- 12 在"数据 / 服务"视图中,从 getItems_paged()操作的上下文菜单中,选择"启用分页"。
- 13 在"选择唯一标识符"对话框中,选择"emp_no"并单击"下一步"。
- 14 在"启用分页"对话框中,将"页大小"字段保留为空。从下拉列表中指定 count()操作。单击"完成"。

注: 默认情况下, Flash Builder 将页大小配置为 20 条记录。在该对话框中,可以为操作指定自定义页大小。

🖻 启用分页	
确认分页操作 通过分页,您可以在	E需要时通过请求较小数据集来检索大型数据集。
为以下源启用分页:	getItems_paged(startIndex : Number, numItems : Number)
页大小:	指定每个调用要访存的项数(可选)
计数操作:	count ()
♀ 启用分页后,说 自动生成。	选择将返回集合中数据总行数的操作 (可选) 调用此操作时不需要 startIndex 和 numItems 参数。这些参数的值可以
?	〈上一步 @) 下一步 @) > 完成 @ 取消

创建应用程序并将 getItems_paged() 绑定到 DataGrid

- 1 如果 MXML 编辑器处于源代码模式,请选择"设计"以在设计模式下打开该编辑器。
- 2 在"组件"视图中,将 DataGrid 组件拖到设计区域并将其放在靠近顶部的地方。 DataGrid 组件列在"控件"下。
- 3 确保选择了 DataGrid 组件, 然后在"属性"视图中, 为"Editable" 属性指定 false。
- 4 选中 DataGrid 组件后,从"数据"菜单中选择"绑定到数据"。
- 5 在"绑定到数据"操作中,选择"新服务调用"。对于 PagingService,选择 getItems_paged()操作。单击"确定"。

區 绑定到数据		X
要调用的操作 ③新服务调用		
服务:	PagingService	*
操作:	<pre>getItems_paged() : Employee[]</pre>	~
○现有调用结果:		\mathbf{v}
数据提供程序: Emp]	loyee[]	~
	确定 取消	

绑定到数据

6 保存,然后运行该文件。

在应用程序中,单击 DataGrid 滚动条,或移动滚动条缩略图,观察数据分页。

🗿 ht i	p://localf	ost:8500/Pagin	gCF/PagingCF	.html - Micr	osoft Intern	et Explorer		
文件 (E) 编辑(E)	查看(Y) 收藏(A)	工具(T) 帮助(b	Ð				-
G f	退 - 🕥	- 🗵 🗟 🏠	🔎 搜索 났	如藏夹 🧭 🍃	3 - 🌺 🔟 ·	· 📃 🗐 🛍	28	
地址@) 🕘 http://1	ocalhost:8500/Pagin	gCF/PagingCF.htm	1		•	🗸 芛 转到	链接 »
								<u>_</u>
	first_name	hire_date	birth_date	emp_no	last_name	gender		
	Georgi	Wed Jun 25 23:(Wed Sep 2 00:0	10001	Facello	M	A	
	Bezalel	Thu Nov 21 00:0	Tue Jun 2 00:00	10002	Simmel	F		
	Parto	Wed Aug 27 23:1	Thu Dec 3 00:00	10003	Bamford	M		
	Chirstian	Mon Dec 1 00:0(Sat May 1 00:00:	10004	Koblick	м		
	Kyoichi	Mon Sep 11 23:(Fri Jan 21 00:00	10005	Maliniak	M		
	Anneke	Thu Jun 1 23:00	Mon Apr 20 00:0	10006	Preusig	F	v	
Ĺ							_	~
🕘 完毕	2					🗐 本	地 Intranet	

分页教程 (PHP)

在本教程中创建的应用程序可访问先前下载和安装的员工数据库。有关该数据库的安装信息,请参阅第14页的"安装 Flash Builder 教程数据库"。

本教程将创建一个使用分页来填充 DataGrid 的应用程序。

🖨 h	ttp://locall	nost:8888/Pag	ingPHP/Pagin	gPHP.html - I	licrosoft Int	ernet Explor	er (
文件	⊧(፻) 编辑(፻)	查看 (V) 收藏 (A)) 工具(1) 帮助) (H)				
G	后退 - 🕥	- 💌 🖻 🏠	🔎 搜索 👷	• 收藏夹 🔗	2· 🎍 🖻	- 📃 👰 🛍	1 🔏 👘	
地址	🛈 🙆 http://1	ocalhost:8888/Pag	;ingPHP/PagingPHP	. html			💙 🄁 转到	j 链接 »
								~
	first_name	hire_date	birth_date	emp_no	last_name	gender		
	Georgi	1986-06-26	1953-09-02	10001	Facello	м	A	
	Bezalel	1985-11-21	1964-06-02	10002	Simmel	F		
	Parto	1986-08-28	1959-12-03	10003	Bamford	м		
	Chirstian	1986-12-01	1954-05-01	10004	Koblick	м		
	Kyoichi	1989-09-12	1955-01-21	10005	Maliniak	М		
	Anneke	1989-06-02	1953-04-20	10006	Preusig	F	v	
								~
(2) 完	毕					9	本地 Intrane	it 🦼

创建远程服务并将其导入到 Flex 项目中

- 1 在 Web 根文件夹中, 创建一个名为 PagingPHP 的文件夹。在 PagingPHP 目录中, 创建一个名为 services 的目录。
- **2** 在您常用的 PHP 编辑器中,创建以下 PHP 文件以实现所需的分页函数。将该文件命名为 PagingService.php,并将其保存 在 services 目录中。

```
<?php
/**
* This sample service contains functions that illustrate typical service operations.
* This code is for prototyping only.
 * Authenticate users before allowing them to call these methods.
*/
class PagingService {
   /* connection variables */
    var $username = "USERNAME";
   var $password = "PASSWORD";
   var $server = "localhost";
   var $port = "3306";
   var $databasename = "fb tutorial db";
   var $tablename = "employees";
   var $connection;
    public function __construct() {
        $this->connection = mysqli_connect(
            $this->server,
            $this->username,
            $this->password,
            $this->databasename,
            $this->port);
    $this->throwExceptionOnError($this->connection);
    }
    public function count() {
        $stmt = mysqli_prepare($this->connection,
                "SELECT COUNT(*) AS COUNT FROM $this->tablename");
        $this->throwExceptionOnError();
        mysqli stmt execute($stmt);
        $this->throwExceptionOnError();
        mysqli_stmt_bind_result($stmt, $rec_count);
        $this->throwExceptionOnError();
        mysqli stmt fetch($stmt);
        $this->throwExceptionOnError();
        mysqli_stmt_free_result($stmt);
        mysqli_close($this->connection);
        return $rec count;
    }
    public function getItems_paged($startIndex, $numItems) {
        $stmt = mysqli_prepare($this->connection,
            "SELECT * FROM $this->tablename LIMIT ?, ?");
        $this->throwExceptionOnError();
        mysqli_bind_param($stmt, 'ii', $startIndex, $numItems);
        mysqli_stmt_execute($stmt);
        $this->throwExceptionOnError();
        $rows = array();
        mysqli_stmt_bind_result($stmt, $row->emp_no, $row->birth_date,
                                     $row->first_name, $row->last_name,
```

} ?>

```
$row->gender, $row->hire_date);
    while (mysqli_stmt_fetch($stmt)) {
       $rows[] = $row;
        $row = new stdClass();
        mysqli_stmt_bind_result($stmt, $row->emp_no, $row->birth_date,
                                    $row->first_name, $row->last_name,
                                     $row->gender, $row->hire_date);
    }
    mysqli_stmt_free_result($stmt);
    mysqli_close($this->connection);
   return $rows;
}
/**
* Utitity function to throw an exception if an error occurs
* while running a mysql command.
*/
private function throwExceptionOnError($link = null) {
   if($link == null) {
        $link = $this->connection;
    }
    if(mysqli_error($link)) {
        $msg = mysqli_errno($link) . ": " . mysqli_error($link);
        throw new Exception('MySQL Error - '. $msg);
    }
}
```

3 在 PagingService.php 中,修改连接变量,以提供访问 fb_tutorial_db 数据库所需的服务器、用户名和密码。

4 在 Flash Builder 中, 创建一个 Flex 项目。将该项目命名为 PagingPHP 并指定 PHP 作为服务器技术。单击"下一步"。

🖻 新建 Flex 项目 🛛 🗌 🔀
创建 Flex 項目。 选择项目的名称和位置,并配置项目将使用的服务器技术。
项目名(P): PagingFHP
项目位置 □ 使用器1位置 00
文件夹: C:\Documents and Settings\ning.guo\Adobe Flash Builder 4\P(浏览 (B)
○ 应用程序类型
💿 🝺 Web(在 Adobe Flash Player 中运行)(例)
○ 🔂 桌面(在 Adobe AIR 中运行) ⑪)
Flex SDK 版本
●使用默认 SDK(当前为 "Flex 4.0") (K) 配置 Flex SDK
○使用特定的 SDK(S): Flex 4.0 ▼
应用程序服务器类型 (1): PHP 🗸
✓使用远程对象访问服务(E)
🔿 LiveCycle Data Services ES(L)
O BlazeDS (B)
○ColdFusion Flash Remoting@)
(⊥-步 @) 下-步 @) 完成 @) 取消

- 5 指定系统的 Web 根目录和根 URL。验证服务器设置,并将 PagingPHP 目录指定为"输出文件夹"。单击"完成"。
- 6 从 Flash Builder 的"数据"菜单中选择"连接数据 / 服务"。选择"PHP"并单击"下一步"。
- 7 单击"浏览",然后导航到在步骤 1 中创建的 PagingService.php 文件。单击"完成"。
 如果系统上没有安装包含 Zend AMF 的 Zend Framework,请单击"确定"并安装 Zend Framework。
- 8 在 Flash Builder 的"数据 / 服务"视图中,从 getItems_paged()操作的上下文菜单中,选择"配置返回类型"。

🖹 问题 🧠 数据/服务 💢 🛃 网络监视器	🇯 🗞 📰 😵 🔗 🝕 🥠 🌃 🖡 🏹 🗆
□ o ^C ₀ PagingService ↓ 数据类型	
🥥 count() : Object	
② getItems_paged(startIndex : Object, numI 🭕 测试操作	
💰 配置返回类型	
30 配置输入类型	
★ 删除	
启用数据管理 启用分页	
■。 生成表单 ● <mark>⑦</mark> 生成服务调用	
◎ <mark>念</mark> 刷新 属性	

9 在"配置返回类型"对话框中,默认情况下选中"自动检测返回类型"。单击"下一步"。

10 为 getItems_paged() 操作的参数指定值和类型,如下所述。单击"下一步":

参数	参数类型	值
startIndex	int	0
numItems	int	20

配置返回类型				
自动检测返回类型 输入调用此操作所需自	的参数值。			
为了自动检测返回类型	앁,Flash Builder 将(使用您提供的下列参数调用搜	操作并分析返回的数据。	
操作: getItems_pag	ed(startIndex : Obje	ct, numItems : Object)		
参数	选择类型	输入值		
startIndex	int	0		
numItems	int	20		
?	[〈上一步(8) 下一步(0	()> 完成(2) 取	消
•				

11 在"以下类型的数组"字段中指定 Employee,为返回的数据定义自定义类型。

Flex 使用自定义数据类型访问并更新从服务器返回的复杂数据类型。

"配置返回类型"对话框显示服务返回的数据类型的属性。

12 单击"完成",配置"Employee"返回类型。

13 在"数据 / 服务"视图中,从 getItems_paged()操作的上下文菜单中,选择"启用分页"。

14 在"选择唯一标识符"对话框中,选择"emp_no"并单击"下一步"。

15 在"启用分页"对话框中,将"页大小"字段保留为空。从下拉列表中指定 count() 操作。单击"完成"。

注:默认情况下, Flash Builder 将页大小配置为 20 条记录。在该对话框中,可以为操作指定自定义页大小。

▣ 启用分页	
确认分页操作 通过分页,您可以在	T需要时通过请求较小数据集来检索大型数据集。
为以下源启用分页:	getItems_paged(startIndex : int, numItems : int)
页大小:	指定每个调用要访存的项数(可选)
计数操作:	count 0
♀ 启用分页后, ↓ 自动生成。	选择将返回集合中数据总行数的操作 (可选) 哥用此操作时不需要 startIndex 和 numItems 参数。这些参数的值可以
?	(上一步®) 下一步®) 完成® 取消

创建应用程序并将 getItems_paged() 绑定到 DataGrid

- 1 如果 MXML 编辑器处于源代码模式,请选择"设计"以在设计模式下打开 MXML 编辑器。
- 2 在"组件"视图中,将 DataGrid 组件拖到设计区域并将其放在靠近顶部的地方。 DataGrid 组件列在"控件"下。
- 3 确保选择了 DataGrid 组件, 然后在"属性"视图中, 为"Editable" 属性指定 false。
- 4 选中 DataGrid 组件后,从"数据"菜单中选择"绑定到数据"。
- **5** 在"绑定到数据"操作中,选择"新服务调用"。对于 PagingService,从下拉列表中选择 get Items Paged()操作。单击"确定"。

匝 绑定到数据		×
要调用的操作 ④新服务调用		
服务:	PagingService	/
操作:	getItems_paged() : Employee[]	
	更改返回类型	
○ 现有调用结果:		
数据提供程序: Emp	loyee[]	~
	确定 取消	
		_

6 保存,然后运行该文件。

在应用程序中,单击 DataGrid 滚动条,或移动滚动条缩略图,观察数据分页。

🗿 h	ttp://localf	nost:8888/Pag	ingPHP/Pagin	gPHP.html - I	licrosoft Int	ernet Explor	er		×
文件	・(ア) 编辑(E)	查看 (V) 收藏 (A)	工具(T) 帮助) (H)				4	7
G	后退 - 🕥	- 💌 🖻 🏠	🔎 搜索 🤸	• 收藏夹 🕢	2- 🎍 🖻	- 📃 👰 🛍	1 🔏		
地址	🛈 🙋 http://1	ocalhost:8888/Pag	ingPHP/PagingPHP	. html			💌 🔁 ¥	专到 链接	»
									-
	first_name	hire_date	birth_date	emp_no	last_name	gender			
	Georgi	1986-06-26	1953-09-02	10001	Facello	м	A		
	Bezalel	1985-11-21	1964-06-02	10002	Simmel	F			
	Parto	1986-08-28	1959-12-03	10003	Bamford	M			
	Chirstian	1986-12-01	1954-05-01	10004	Koblick	М			
	Kyoichi	1989-09-12	1955-01-21	10005	Maliniak	M			
	Anneke	1989-06-02	1953-04-20	10006	Preusig	F	v		
									~
3 🕑	毕						本地 Intra	anet	

第6章:使用数据管理同步服务器更新

通过数据管理功能,可以在数据库中同步添加、更新和删除记录。在调用 commit 方法之前,在客户端应用程序中所做的更改 不会写入服务器。您可以调用 revert 方法来恢复在客户端应用程序中所做的更改。

此外,数据管理还有助于将客户端应用程序与服务上的数据同步。数据管理功能可以自动更新显示从服务检索的数据的用户界 面控件。

本教程将说明如何使用 Flash Builder 实现数据管理。本教程介绍了如何将服务器上的数据与客户端应用程序控件同步。

数据管理需要实现一个或多个具有下列签名的操作的数据服务:

- createItem(item:CustomDatatype):int
- deleteItem(itemID:Number):void
- updateItem((item: *CustomDatatype*):void
- getItem(itemID:Number): *CustomDatatype*

CustomDatatype 这种数据类型表示从服务器返回的复杂数据。在服务器端键入中,服务定义了自定义数据类型。在客户端键入中,使用 Flash Builder 内部检查服务并配置自定义数据类型。

注: 在本教程中列出的所需数据管理操作签名是可以使用的签名的子集。有关可以使用的操作签名的完整列表,请参阅启用数 据管理。

数据管理教程 (ColdFusion)

您在本教程中创建的应用程序可访问 fb_tutorial_db 数据库。关于该数据库的安装信息,请参阅第 14 页的"安装 Flash Builder 教程数据库"。.

在本教程中,将创建一个包含可编辑 DataGrid 的应用程序,此 DataGrid 用于显示员工记录。可以在 DataGrid 中适当的位置修改一条或多条记录。还可以在 DataGrid 中添加或删除选定记录。在您选择"保存全部更改"按钮更新数据库之前,DataGrid 中的所有更改都只是本地的。

@ 1	ttp://locall	nost:8500/Dat	allgtCF/Datal	lgtCF.html	- Hicrosoft]	Internet 🔳	
文作	キ(27) 编辑(23)	查看(V) 收藏(A) 工具(1) 帮助	ታዊን			-
G	后退 - 🕥	- 🖹 💈 🎸) 🔎 搜索 🔮	了收藏夹 🍕	3 🔗 💊 🗖	3 🚳	
地址	: 🕕 🙋 http://1	.ocalhost:8500/Da	taMgtCF/DataMgtC	F. html		💙 🄁 转到 🕴	连接 »
							~
	ID	Last Name	First Name	Gender	Date of Birth	Hire Date	
	10003	Bamford	Parto	М	Thu Dec 3 00:00:0	Wed Aug 27 23:00: 🔺	
	10004	Koblick	Chirstian	м	Sat May 1 00:00:00	Mon Dec 1 00:00:01	
	10005	Maliniak	Kyoichi	м	Fri Jan 21 00:00:01	Mon Sep 11 23:00:1	
	10006	Preusig	Anneke	F	Mon Apr 20 00:00:1	Thu Jun 1 23:00:00	
	10007	Zielinski	Tzvetan	F	Thu May 23 00:00:	Fri Feb 10 00:00:00	
	10008	Kalloufi	Saniya	м	Wed Feb 19 00:00	Thu Sep 15 00:00:0 🔻	
	添加	删除 重	2 2	保存]		
ľ					_		
							\sim
8 7	記悼					🧐 本地 Intranet	

创建远程服务并将其导入到 Flex 项目中

- 1 在 Web 根文件夹中, 创建一个名为 DataMgtCF 的文件夹。
- 2 在您常用的 ColdFusion 编辑器中, 创建以下 CFC。将该 CFC 命名为 EmployeeService.cfc, 然后将其放在 Web 根文件夹的 DataMgtCF 文件夹中。

EmployeeService.cfc 包含 Flash Builder 实现数据管理所需的函数。

```
<!---
This sample service contains functions that illustrate typical service operations.
This code is for prototyping only.
Authenticate the user prior to allowing them to call these methods. You can find more
information at http://www.adobe.com/go/cf9_usersecurity
--->
<cfcomponent output="false">
    <cffunction name="getAllItems" output="false" access="remote" returntype="any" >
        <cfset var qAllItems="">
        <cfquery name="qAllItems" datasource="fb tutorial db">
            SELECT * FROM employees
        </cfquery>
        <cfreturn qAllItems>
    </cffunction>
    <cffunction name="getItem" output="false" access="remote" returntype="any" >
        <cfargument name="itemID" type="numeric" required="true" />
        <cfset var qItem="">
        <cfquery name="qItem" datasource="fb_tutorial_db">
            SELECT * FROM employees
                WHERE emp_no = <CFQUERYPARAM CFSQLTYPE="CF_SQL_INTEGER" VALUE="#ARGUMENTS.itemID#">
        </cfquery>
         <cfreturn qItem>
    </cffunction>
    <cffunction name="createItem" output="false" access="remote" returntype="any" >
        <cfargument name="item" required="true" />
            <cfquery name="createItem" datasource="fb_tutorial_db" result="result">
                INSERT INTO employees (first_name, last_name, gender, birth_date, hire_date)
                VALUES (<CFQUERYPARAM cfsqltype="CF_SQL_VARCHAR" VALUE="#item.first_name#">,
                         <CFQUERYPARAM cfsqltype="CF SQL VARCHAR" VALUE="#item.last name#">,
                         <CFQUERYPARAM cfsqltype="CF SQL VARCHAR" VALUE="#item.gender#">,
                        <CFQUERYPARAM cfsqltype="CF SQL DATE" VALUE="#item.birth date#">,
                         <CFQUERYPARAM cfsqltype="CF_SQL_DATE" VALUE="#item.hire_date#">)
            </cfquery>
        <cfreturn result.GENERATED KEY/>
    </cffunction>
    <cffunction name="updateItem" output="false" access="remote" returntype="void" >
        <cfargument name="item" required="true" />
```

<cfquery name="updateItem" datasource="fb_tutorial_db">

```
UPDATE employees SET birth_date = <CFQUERYPARAM cfsqltype="CF_SQL_DATE"
       VALUE="#item.birth date#">,
       hire_date = <CFQUERYPARAM cfsqltype="CF_SQL_DATE" VALUE="#item.hire_date#">,
       gender = <CFQUERYPARAM cfsqltype="CF SQL VARCHAR" VALUE="#item.gender#">,
        first_name = <CFQUERYPARAM cfsqltype="CF_SQL_VARCHAR" VALUE="#item.first_name#">,
       last_name = <CFQUERYPARAM cfsqltype="CF_SQL_VARCHAR" VALUE="#item.last_name#">
       WHERE emp_no = <CFQUERYPARAM CFSQLTYPE="CF_SQL_INTEGER" VALUE="#item.emp_no#">
    </cfguery>
</cffunction>
<cffunction name="deleteItem" output="false" access="remote" returntype="void" >
    <cfargument name="itemID" type="numeric" required="true" />
   <cfquery name="delete" datasource="fb_tutorial_db">
            DELETE FROM employees
            WHERE emp no = <CFQUERYPARAM CFSQLTYPE="CF SQL INTEGER" VALUE="#ARGUMENTS.itemID#">
    </cfquery>
</cffunction>
```

```
</cfcomponent>
```

- **3** 在 Flash Builder 中,创建一个 Flex 项目。将该项目命名为 DataMgtCF,并指定 ColdFusion 作为服务器技术,如下所示。单击"下一步"。
 - 应用程序服务器类型: ColdFusion
 - 启用"使用远程对象访问服务"
 - 选择 "ColdFusion Flash Remoting"
- 4 验证 ColdFusion 设置,并将 DataMgtCF 文件夹指定为"输出文件夹"。单击"完成"。
- 5 在 Flex Builder 的"数据"菜单中,选择"连接数据 / 服务"。选择"ColdFusion"。单击"下一步"。
- 6 单击"浏览",然后导航到在步骤 2 中创建的 EmployeeService.cfc 文件。单击"完成"。 根据需要为系统提供授权凭据。
- 7 在 Flash Builder 的"数据 / 服务"视图中,从 getItem()操作的上下文菜单中选择"配置返回类型"。

Flash Builder 需要为从数据服务返回的数据配置自定义数据类型。通过内部检查 getItem() 操作,可以定义自定义数据类型 Employee。

- 8 在"配置返回类型"对话框中,默认情况下选中"自动检测返回类型"。单击"下一步"。
- 9 为 getItem() 操作的参数指定值和类型,如下所述。单击"下一步":

参数	参数类型	值
itemID	Number	10001

对于 getItem(),将提供对应于数据库键字段的有效参数。"Number"和"10001"是"itemID"的有效类型和值。

10 在"以下类型的数组"字段中指定 Employee,为返回的数据定义自定义类型。

Flex 使用自定义数据类型访问并更新从服务器返回的复杂数据类型。getItem() 返回数据库的 Employees 表中的记录的字段。

"配置返回类型"对话框显示服务返回的数据类型的属性。

11 单击"完成", 配置"Employee"返回类型。

12 在 Flash Builder 的"数据 / 服务"视图中,从 getAllItems() 操作的上下文菜单中,选择"配置返回类型"。

13 选择"使用现有数据类型"。从"以下类型的数组"下拉列表中选择 Employee,以指定 Employee 数组。单击"完成"。

启用数据管理功能

- 1 在"数据/服务"视图中,展开 EmployeeService 的"数据类型"节点并选择"Employee"数据类型。
- 2 从"Employee"数据类型的上下文菜单中,选择"启用数据管理"。

🚼 问题 🧠 数据/服务	😥 🕎 网络监视器	🇯 🗞 🎫 📽 🐟 🍕 🤣 🌃 🖡 🗸 🗆 🗆
■ ● EmployeeServic ● ● ● 数据类型 ● ● ● Employee ● ○ createItem ● ▲ deleteItem ● ● ● getAllItem	ce ● ② 测试操作 ● 配置返回类型 』① 配置输入类型	
œ ⊶ ≩ getItem (it ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥	★ 删除 启用数据管理 启用为方页	
	■ 生成表单 ● 生成服务调用	
	● <mark>② 刷新</mark> 属性 <u>Source</u>	

- 3 在"选择唯一标识符"对话框中,选择"emp_no"并单击"下一步"。
- 4 在"映射数据库操作"对话框中,指定下列操作,然后单击"完成"。
 - 创建 (添加) 操作: createItem()
 - 获取项操作: getItem()
 - 更新操作: updateItem()
 - 删除操作: deleteItem()

▶ 数据管理		
映射数据管理操作 数据管理将自动执行	宁在应用程序与服务器之间同步数 据 的过程。	
服务:	EmployeeService	*
选择与下列操作对	应的数据服务操作:	
创建(添加)项:	createItem(item : Object) : Object	~
	例如: createItem(item:Employee)	
获取项:	getItem(itemID : Number) : Employee[]	~
	例如: getItem(itemID)	
更新项:	updateItem(item : Object) : void	*
00.02A75	例如: updateItem(item:Employee)	
卅Ⅳ乐坝∶	deleteltem(itemil) : int) : void	×
	[μ]gU: deleteltem(ltemil) sk deletitem(ltem:Employee)	
?	〈上一步 ⑧〉 下一步 ⑲〉 完成 ⑨	取消

现在即为此操作启用了数据管理。Flash Builder 将生成客户端代码,该代码可以使用已映射的操作的组合更新数据。

创建应用程序并添加 DataGrid 和 Button

- 1 如果 MXML 编辑器处于源代码模式,请选择"设计"以在设计模式下打开 MXML 编辑器。
- 2 在"组件"视图中,将 DataGrid 组件拖到设计区域并将其放在靠近顶部的地方。

DataGrid 组件列在" 控件"下。

3 在"属性"视图中,选中 DataGrid 后,指定下列属性:

属性	值
ID	dg
Editable	true

- 4 将四个 Button 组件拖动到设计区域,使其在 DataGrid 下面排成一行。
- 5 双击各个按钮以编辑它们的标签。提供下列标签:

标签	
添加	
删除	
还原	
保存全部更改	

6 在"数据 / 服务"视图中,选择 getAllItems() 操作并将其拖放到 DataGrid 上。

在"绑定到数据"对话框中, EmployeeService 和 getAllItems()处于选中状态。单击"确定"。

🖉 first_name	hire_date	birth_date	emp_no	last_name	gender	ľ
ļ						ł

7 (可选)选中 DataGrid 后,在"属性"视图中,单击"配置列",然后执行以下操作: 对列进行重命名和重新排列。

为 Button 生成事件处理函数

每个 Button 都需要事件处理函数指定其被单击时要执行的操作。Flash Builder 会为事件处理函数生成存根,然后您可以对其进行编码,以指定要进行的服务操作。

选择"添加"按钮。在"属性"视图中,在"添加"按钮的"单击时"字段附近,单击该图标。选择"生成事件处理函数"。
 MXML编辑器将切换到"源代码"视图,且光标位于生成的事件处理函数中。

2 在事件处理函数主体中, 键入以下内容:

```
var e:Employee = new Employee();
var birthDate:Date = new Date(2000, 01, 01);
var hireDate:Date = new Date(2000, 01, 01);
```

```
e.first_name = "New";
e.last_name = "New";
e.birth_date = birthDate;
e.hire_date = hireDate;
e.gender = "M";
dg.dataProvider.addItem(e);
dg.verticalScrollPosition = dg.dataProvider.length -1;
```

键入时, Flash Builder 内容辅助将帮助您查看可用的方法和值。

3 在脚本块中,将以下导入语句添加到现有导入语句后面:

import services.employeeservice.EmployeeService; import valueObjects.Employee;

4 在设计模式下,为"删除"按钮添加"单击时"事件处理函数并指定以下代码:

employeeService.deleteItem(dg.selectedItem.emp_no);

5 同样,为"还原"按钮添加"单击时"事件处理函数,并指定下列代码:

employeeService.getDataManager(employeeService.DATA_MANAGER_EMPLOYEE).revertChanges();
getAllItemsResult.token = employeeService.getAllItems();

6 为"保存全部更改"按钮添加"单击时"事件处理函数,并指定下列代码:

employeeService.commit();

7 对于 DataGrid 的 creationCompleteHandler(), 添加以下代码行:

protected function dg_creationCompleteHandler(event:FlexEvent):void

employeeService.getDataManager(employeeService.DATA MANAGER EMPLOYEE).autoCommit=false;

employeeService.getDataManager(employeeService.DATA_MANAGER_EMPLOYEE).deleteItemOnRemoveFromFil
l=true;

getAllItemsResult.token = employeeService.getAllItems();

}

此代码是为了确保可通过选择"还原"按钮来还原删除的项。

8 保存该应用程序,选择"运行">"运行 DataMgtCF"。

@ 1	http://localk	nost:8500/Dat	allgtCF/Datal	gtCF.html	- Licrosoft]	Internet		X
文化	牛(2) 编辑(2)	查看(V) 收藏(A) 工具(1) 帮助	ታዊጋ				
G	🔇 后退 🔹 🕑 🔹 🛃 🏠 🔎 搜索 🥎 收藏夹 🤣 🔂 🍓 🔀							
地址	🗄 🛈 🙆 http://1	ocalhost:8500/Da	taMgtCF/DataMgtCF	?.html		💙 🄁 转到	链:	接 »
								^
	ID	Last Name	First Name	Gender	Date of Birth	Hire Date		
	10003	Bamford	Parto	М	Thu Dec 3 00:00:0	Wed Aug 27 23:00:	۸	
	10004	Koblick	Chirstian	м	Sat May 1 00:00:00	Mon Dec 1 00:00:01		
	10005	Maliniak	Kyoichi	м	Fri Jan 21 00:00:01	Mon Sep 11 23:00:1		
	10006	Preusig	Anneke	F	Mon Apr 20 00:00:1	Thu Jun 1 23:00:00		
	10007	Zielinski	Tzvetan	F	Thu May 23 00:00:	Fri Feb 10 00:00:00		
	10008	Kalloufi	Saniya	М	Wed Feb 19 00:00	Thu Sep 15 00:00:0	Ŧ	
	添加	删除 重調	置	保存				
					_			
_						1.000	_	\sim
٤	完毕					🧐 本地 Intranet		

可以在 DataGrid 中的适当位置更新员工。单击"保存全部更改"后,更改将在服务器上更新。服务器更新的原因是,数据 管理功能调用 updateItem()操作,以保持数据在客户端和服务器之间同步。

在单击"保存全部更改"之前,单击"还原"按钮可撤消所做的所有更改。

可以添加和删除员工。添加和删除员工后,数据管理功能会考虑添加或删除相应的 DataGrid 行。如果不启用数据管理,您 将自己对 DataGrid 的更新进行编码。

添加员工后,员工编号默认为0。由于该新员工尚未在服务器上更新,因此该默认值为0。单击"保存全部更改"后,该新员工将添加到服务器。此时,生成员工编号。

单击"保存全部更改"按钮,将所有更改写入数据库。

数据管理教程 (PHP)

在本教程中创建的应用程序可访问先前下载和安装的员工数据库。有关该数据库的安装信息,请参阅第14页的"安装 Flash Builder 教程数据库"。

在本教程中,将创建一个包含可编辑 DataGrid 的应用程序,此 DataGrid 用于显示员工记录。您可以在 DataGrid 中适当的 位置修改一条或多条记录,也可以从 DataGrid 添加或删除选定记录。在您选择"保存全部更改"按钮更新数据库之前, DataGrid 中的所有更改都只是本地的。

@ 1	http://locall	host:8888/Dat	allgtPHP/Data	allgtPHP.html	- H icrosoft	Internet E	
文任	牛(27) 编辑(26)	查看(V) 收藏(A) 工具(1) 帮!	መ			.
G	后退 - 🕥	- 💌 🖻 🎸) 🔎 搜索 🚽	了收藏夹 🧭	🖉 🍓 🖻	-25	
地址	1 🕖 🛃 http://1	localhost:8888/Da	taMgtPHP/DataMgt	PHP. html		💌 🔁 🕸	到 链接 》
							<u></u>
	first_name	hire_date	birth_date	emp_no	last_name	gender	
	Parto	1986-08-28	1959-12-03	10003	Bamford	м	A
	Chirstian	1986-12-01	1954-05-01	10004	Koblick	м	
	Kyoichi	1989-09-12	1955-01-21	10005	Maliniak	м	
	Anneke	1989-06-02	1953-04-20	10006	Preusig	F	
	Tzvetan	1989-02-10	1957-05-23	10007	Zielinski	F	
	Saniya	1994-09-15	1958-02-19	10008	Kalloufi	м	•
	添加	删除 重調		保存	-		
							~
E) 9	完毕					🧐 本地 Intra	net

创建远程服务并将其导入到 Flex 项目中

- 1 在 Web 根文件夹中, 创建一个名为 DataMgtPHP 的文件夹。在该文件夹内, 创建一个名为 services 的文件夹。
- 2 在您常用的 PHP 编辑器中,创建以下 PHP 文件以实现服务。将该文件命名为 EmployeeService.php,并放在 Web 根文件 夹的 DataMgtPHP/services 文件夹中。

```
<?php
```

class EmployeeService {

```
var $username = "USERNAME";
var $password = "PASSWORD";
var $server = "localhost";
var $port = "3306";
var $databasename = "fb_tutorial_db";
var $tablename = "employees";
var $connection;
public function __construct() {
    $this->connection = mysqli_connect(
                             $this->server.
                             $this->username,
                             $this->password,
                             $this->databasename,
                             $this->port
                         );
    $this->throwExceptionOnError($this->connection);
```

}

```
public function getAllEmployees() {
    $stmt = mysqli prepare($this->connection, "SELECT * FROM $this->tablename");
    $this->throwExceptionOnError();
    mysqli_stmt_execute($stmt);
    $this->throwExceptionOnError();
    $rows = array();
    mysqli_stmt_bind_result($stmt, $row->emp_no, $row->birth_date,
            $row->first_name, $row->last_name, $row->gender, $row->hire_date);
   while (mysqli stmt fetch($stmt)) {
     $rows[] = $row;
      $row = new stdClass();
     mysqli_stmt_bind_result($stmt, $row->emp_no, $row->birth_date,
            $row->first_name, $row->last_name, $row->gender, $row->hire_date);
    }
    mysqli stmt free result($stmt);
    mysqli close($this->connection);
   return $rows;
}
public function getEmployeesByID($itemID) {
    $stmt = mysqli_prepare($this->connection,
                             "SELECT * FROM $this->tablename where emp_no=?");
    $this->throwExceptionOnError();
    mysqli bind param($stmt, 'i', $itemID);
    $this->throwExceptionOnError();
    mysqli_stmt_execute($stmt);
    $this->throwExceptionOnError();
    mysqli stmt bind result($stmt, $row->emp no, $row->birth date,
            $row->first_name, $row->last_name, $row->gender, $row->hire_date);
    if(mysqli_stmt_fetch($stmt)) {
        return $row;
    } else {
        return null;
    }
}
public function createEmployees($item) {
    $stmt = mysqli_prepare($this->connection,
        "INSERT INTO $this->tablename (emp_no, birth_date,
        first_name, last_name, gender, hire_date) VALUES (?, ?, ?, ?, ?, ?)");
    $this->throwExceptionOnError();
    mysqli_bind_param($stmt, 'isssss', $item->emp_no, $item->birth_date,
        $item->first_name, $item->last_name, $item->gender, $item->hire_date);
    $this->throwExceptionOnError();
    mysqli_stmt_execute($stmt);
    $this->throwExceptionOnError();
```

}

"。

```
$autoid = mysqli_stmt_insert_id($stmt);
          mysqli stmt free result($stmt);
           mysqli_close($this->connection);
          return $autoid;
       }
      public function updateEmployees($item) {
           $stmt = mysqli_prepare($this->connection,
           "UPDATE $this->tablename SET emp_no=?, birth_date=?,
              first_name=?, last_name=?, gender=?, hire_date=?
              WHERE emp no=?");
           $this->throwExceptionOnError();
          mysqli_bind_param($stmt, 'isssssi', $item->emp_no, $item->birth_date,
               $item->first_name, $item->last_name, $item->gender, $item->hire_date, $item->emp_no);
           $this->throwExceptionOnError();
          mysqli stmt execute($stmt);
           $this->throwExceptionOnError();
          mysqli_stmt_free_result($stmt);
           mysqli close ($this->connection);
       }
      public function deleteEmployees($itemID) {
          $stmt = mysqli_prepare($this->connection,
               "DELETE FROM $this->tablename WHERE emp no = ?");
           $this->throwExceptionOnError();
          mysqli_bind_param($stmt, 'i', $itemID);
          mysqli_stmt_execute($stmt);
          $this->throwExceptionOnError();
          mysqli_stmt_free_result($stmt);
          mysqli close($this->connection);
       }
       /**
       * Utitity function to throw an exception if an error occurs
       * while running a mysql command.
        */
      private function throwExceptionOnError($link = null) {
          if($link == null) {
              $link = $this->connection;
           }
          if(mysqli_error($link)) {
              $msg = mysqli_errno($link) . ": " . mysqli_error($link);
               throw new Exception('MySQL Error - '. $msg);
          }
       }
   ?>
3 修改连接变量,以提供访问 fb_tutorial_db 数据库所需的服务器、用户名和密码。
4 在 Flash Builder 中,创建一个 Flex 项目。将该项目命名为 DataMgtPHP,并指定 PHP 作为服务器技术。单击"下一步
```

上次更新 2010/3/18

- 5 指定系统的 Web 根目录和根 URL。验证服务器设置,并将 DataMgtPHP 目录指定为"输出文件夹"。单击"完成"。
- 6 从 Flash Builder 的"数据"菜单中选择"连接数据 / 服务"。选择"PHP"。单击"下一步"。
- 7 单击"浏览",然后导航到在步骤2中创建的 EmployeeService.php 文件。选择 EmployeeService.php。单击"完成"。
- 8 在 Flash Builder 的"数据/服务"视图中,从 getEmployeesbyID()操作的上下文菜单中,选择"配置返回类型"。 Flash Builder 需要为从数据服务返回的数据配置自定义数据类型。通过内部检查 getEmployeesbyID()操作,可以定义自定义数据类型 Employee。
- 9 在"配置返回类型"对话框中,默认情况下选中"自动检测返回类型"。单击"下一步"。

10 为 getEmployeesByID() 操作的参数指定值和类型,如下所述。单击"下一步":

参数	参数类型	值
itemID	int	10001

对于 getEmployeesByID(),将提供对应于数据库键字段的有效参数。"int"和"10001"是"itemID"的有效类型和值。

11 指定 Employee,以便为返回的数据定义自定义类型。

Flex 使用自定义数据类型访问并更新从服务器返回的复杂数据类型。getEmployeesByID() 返回数据库的 Employees 表中的记录的字段。

"配置返回类型"对话框显示服务返回的数据类型的属性。

12 单击"完成",将"Employee" 配置为返回类型。

- 13 在 Flash Builder 的"数据/服务"视图中,从 getAllEmployees()操作的上下文菜单中,选择"配置返回类型"。
- 14 在"配置返回类型"对话框中,默认情况下选中"自动检测返回类型"。单击"下一步"。
- 15 选择"使用现有数据类型"。从"以下类型的数组"下拉列表中选择 Employee。单击"完成"。
- 16 为 deleteEmployees() 配置参数类型和返回类型,如下所示:
 - a 从 deleteEmployees() 操作的上下文菜单中,选择"配置输入类型"。指定"int"作为类型。单击"确定"。
 - **b** 从 deleteEmployees() 操作的上下文菜单中,选择"配置返回类型"。选择"使用现有数据类型"。指定"void"。单击"完成"。

启用数据管理功能

- 1 在"数据/服务"视图中,展开 EmployeeService 的"数据类型"节点并选择"Employee"数据类型。
- 2 从"Employee"数据类型的上下文菜单中,选择"启用数据管理"。

🚼 问题 🤹 数据/服务	🛛 🖾 网络监视器	🚺 ASDoc 🔉 🝰 📽 🐟 🍕 🥵 🌃 🛼 🏹 🗆
 ● ● ● ● ● Employe ● ○ CreateEmpl ● △ deleteEmpl ● ▲ getAllEmpl ● ▲ getEmploye 	• • • • • • • • • • • • • •	ect Employee
∟… updateEmpl	 ▲ MURY 启用效氮名管理 启用分页 生成表单 ·⑤, 生成服务调用 ·⑥, 引新 属性 	ect

- 3 在"选择标识属性"对话框中,选择 emp_no。单击"下一步"。
- 4 在"映射数据库操作"对话框中,指定下列操作,然后单击"完成"。
 - 创建 (添加) 操作: createEmployees()
 - 获取项操作: getEmployeesByID()
 - 更新操作: updateEmployees()
 - 删除操作: deleteEmployees()

區 數据管理		
映射数据管理操作 数据管理将自动执行	宁在应用程序与服务器之间同步数 据 的过程。	
服务:	EmployeeService	~
选择与下列操作对	应的数据服务操作:	
创建(添加)项:	createEmployees(item : Object) : Object	~
	例如: createItem(item:Employee)	
获取项:	getEmployeesByID(itemID : int) : Employee	~
	例如: getItem(itemID)	
更新项:	updateEmployees(item : Object) : Object	*
8818 A -77	例如: updateItem(item:Employee)	
册除项:	deleteEmployees(itemID : int) : void	*
	(列幼D: deleteItem(itemID) 気 deletItem(item:Employee)	
?	(上一步 @) 下一步 (@)> 完成 @)	取消

单击"完成"。现在即为此操作启用了数据管理。Flash Builder 将生成客户端代码,该代码可以使用已映射的操作的组合更新数据。

创建应用程序并添加 DataGrid 和 Button

- 1 如果 MXML 编辑器处于源代码模式,请选择"设计"以在设计模式下打开 MXML 编辑器。
- 2 在"组件"视图中,将 DataGrid 组件拖到设计区域并将其放在靠近顶部的地方。

DataGrid 组件列在"控件"下。

3 在"属性"视图中,选中 DataGrid 后,指定下列属性:

属性	值
ID	dg
Editable	true

4 将四个 Button 组件拖动到设计区域, 使其在 DataGrid 下面排成一行。

5 双击各个按钮以编辑它们的标签。提供下列标签:

标签
添加
删除
还原
保存全部更改

😓 *Dat	aMgtCF.mxml 🔀						
∽ 源	调 🔚 设计 🗧) 📰 🛛 状态: 🕓	atel> 🔽			🥎 🔍 100%	*
							Ĥ
	Ø						
	first_name	hire_date	birth_date	emp_no	last_name	gender	
	2					ę	=
	-			-		-	
							Н
	添加	删除 重置		保存			
		,					

6 在"数据 / 服务"视图中,选择 getAllEmployees()操作并将其拖放到 DataGrid 上。

在"绑定到数据"对话框中, EmployeeService 和 getAllEmployees()处于选中状态。单击"确定"。

7 (可选)选中 DataGrid 后,在"属性"视图中,单击"配置列"。对列进行重命名和重新排列。

为 Button 生成事件处理函数并进行编码

对于每个 Button,请创建一个事件处理函数,来指定当单击该 Button 时要执行的操作。Flash Builder 会为事件处理函数生成存根,然后您可以对其进行编码,以指定要进行的服务操作。

- 选择"添加"按钮。在"属性"视图中,在"添加"按钮的"单击时"字段附近,单击该图标。选择"生成事件处理函数"。
 MXML编辑器将切换到"源代码"视图,且光标位于生成的事件处理函数中。
- 2 在事件处理函数主体中, 键入以下内容:

```
var e:Employee = new Employee();
e.first_name = "New";
e.last_name = "New";
e.birth_date = "2000-01-01";
e.hire_date = "2000-01-01";
e.gender = "M";
dg.dataProvider.addItem(e);
dg.verticalScrollPosition = dg.dataProvider.length -1;
```

键入时, Flash Builder 内容辅助将帮助您查看可用的方法和值。

3 在脚本块中,将以下导入语句添加到现有导入语句后面:

import services.employeeservice.EmployeeService; import valueObjects.Employee;

4 在设计模式下,为"删除"按钮添加"单击时"事件处理函数并指定以下代码:

employeeService.deleteEmployees(dg.selectedItem.emp_no);

5 同样,为"还原"按钮添加"单击时"事件处理函数,并指定下列代码:

employeeService.getDataManager(employeeService.DATA_MANAGER_EMPLOYEE).revertChanges();
getAllEmployeesResult.token = employeeService.getAllEmployees();

6 为"保存全部更改"按钮添加"单击时"事件处理函数,并指定下列代码:

employeeService.commit();

7 对于 DataGrid 的 creationCompleteHandler(), 添加以下代码行:

protected function dg_creationCompleteHandler(event:FlexEvent):void

employeeService.getDataManager(employeeService.DATA_MANAGER_EMPLOYEE).autoCommit=false;

employeeService.getDataManager(employeeService.DATA_MANAGER_EMPLOYEE).deleteItemOnRemoveFromFil
l=true;

```
getAllItemsResult.token = employeeService.getAllItems();
```

此代码是为了确保可通过选择"还原"按钮来还原删除的项。

8 保存该应用程序, 然后选择"运行">"运行 DataMgtPHP"。

@)	http://localk	nost:8888/Dat	allgtPHP/Data	llgtPHP.html	- Microsoft	Internet E				
文1	牛(27) 编辑(23)	查看(V) 收藏(A) 工具(1) 帮!	助(H)			.			
G	后退 • 🕥	- 💌 🖻 🎸) 🔎 搜索 🚽	了收藏夹 🔗	🖉 🍓 🖻	-26				
地力	地址 🕕 🧉 http://localhost:8888/DataMgtFHF/DataMgtFHF.html 🛛 🍷 转到 链接 🎽									
							<u>~</u>			
	first_name	hire_date	birth_date	emp_no	last_name	gender				
	Parto	1986-08-28	1959-12-03	10003	Bamford	M				
	Chirstian	1986-12-01	1954-05-01	10004	Koblick	M				
	Kyoichi	1989-09-12	1955-01-21	10005	Maliniak	М				
	Anneke	1989-06-02	1953-04-20	10006	Preusig	F				
	Tzvetan	1989-02-10	1957-05-23	10007	Zielinski	F				
	Saniya	1994-09-15	1958-02-19	10008	Kalloufi	M 👻				
	添加	删除 重打	R	保存						
							~			
e]:	🔮 完毕 📃 😼 本地 Intranet									

可以在 DataGrid 中的适当位置更新员工。单击"保存全部更改"后,更改将在服务器上更新。服务器更新的原因是,数据 管理功能调用 updateEmployees()操作,以保持数据在客户端和服务器之间同步。

在单击"保存全部更改"之前,单击"还原"按钮可撤消所做的所有更改。

可以添加和删除员工。添加和删除员工后,数据管理功能会考虑添加或删除相应的 DataGrid 行。如果不启用数据管理,您 将自己对 DataGrid 的更新进行编码。

添加员工后,员工编号默认为0。由于该新员工尚未在服务器上更新,因此该默认值为0。单击"保存全部更改"后,该新员工将添加到服务器。此时,生成员工编号。

单击"保存全部更改"按钮,将所有更改写入数据库。