

数据服务空间 Data Service Space

用户手册

1. 软件介绍

本软件的主要功能是把网页、WebAPI 以及关系型数据库等数据格式异构的数据源封装为数据格式一致的数据服务，从而为其他应用提供统一的数据访问接口。同时，这些数据服务可以通过组合形成新的复合数据服务。

本软件为 Web 应用程序，即用户不必安装客户端，通过浏览器即可使用全部软件功能。

2. 数据服务的封装

2.1. 封装网页

1. 进入首页，登录，鼠标移动到 CreateDataService 菜单，点击 Create Clippers，呈现图 1 所示的界面。
2. 在网页中的地址栏内输入待封装的网页地址，例如百度搜索的结果网页 <http://www.baidu.com/s?bs=%D4%C6%BC%C6%CB%E3&f=8&wd=%D6%D0%B9%FA>，并点击 Open 按钮，打开该网页。
3. 在网页中通过点击，标注出需要加入数据服务的数据内容。标注的数据内容必须是同类的：在本例中，标注每个搜索结果的标题（被红色高亮显示）；系统根据用户的标注，可以自动推测类似数据（被蓝色高亮显示），如图 2 所示。标注完毕之后，点击 Add Property 按钮，从而完成一个属性的添加。
4. 可以重复上一步骤，实现多个属性的添加。也可以随时点击 Preview 按钮，查看封装结果。
5. 添加完属性之后，点击 Create 按钮，弹出图 3 所示的对话框。在该对话框中，输入数据服务名称、服务类型和描述。点击对话框中的 Create 按钮，完成数据服务的创建。点击菜单 MyDataService，看一看自己创建的服务，点击菜单 Browse，可以看到所有用户创建的服务。

4.1.1.将对此详细说明。



图 1. 网页标注界面



图 2. 样例网页

Parameters

Name

Service Type
Please choose the service type..... ▼

Description

Select	Description	URL Parameter Name	URL Parameter Value
<div> Create Cancel </div>			

图 3. 数据服务保存界面

2.2. 封装 WebAPI

1. 进入首页，登录，鼠标移动到 CreateDataService 菜单，点击 CreateDataService Wrapper。
2. 在网页中的地址栏内输入待封装的 Web API 地址（XML/JSON API 均可支持）。例如输入 Google 搜索 API: <https://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=Paris%20Hilton&callback=processResults>
3. 点击 Load 按钮，查看封装结果。
4. 点击 Create 按钮，弹出类似于图 3 的对话框。输入数据服务名称、服务类型和描述，点击 Create 完成数据服务的创建。

2.3. 封装关系型数据库

1. 进入首页，登录，鼠标移动到 CreateDataService 菜单，点击 CreateDataService Wrapper，点击 DataBase 选项卡。
2. 点击 Connect，在图 4 所示的对话框中点击 DataBase Type 下拉列表中，选择需要封装的数据库类型。然后依次添加要封装的数据库主机 IP、端口、数据库名、数据库用户、数据库密码。我们支持三种数据库，分别是：MySQL、SQL Server、Oracle。
3. 点击 Connect，在图 5 所示的对话框中，点击标注为红色的下拉菜单，选择需要封装的数据库表。
4. 点击 Create 按钮，弹出如图 3 所示的对话框，输入数据服务名称、服务类型和描述。点击 Create 按钮，完成数据服务的创建。

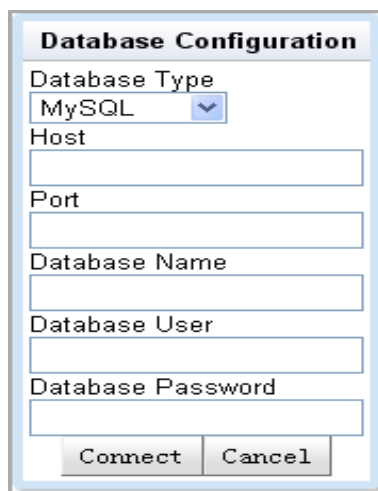


图 4. 封装数据库界面

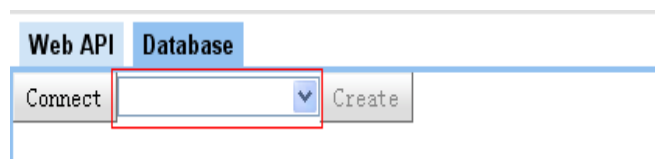


图 5. 封装数据库存储界面

3. 数据服务的组合

前面封装的数据服务可以通过组合生成新的数据服务。数据服务可以被视为一个数据生成器，每次调用都可以得到一个数据实例；数据实例可以用 XML/JSON 的方式表示，也可以用嵌套表的可视化方式表示。数据服务的组合就是取若干数据服务的数据实例，对它们以嵌套表的方式呈现；用户对这些嵌套表进行操作，得到新的嵌套表；最后把操作过程封装成复合数据服务。

用户对嵌套表的操作可以分为列操作，行操作和表操作三种。其中，列操作包括新建，复制，移动，重命名，删除和编辑；行操作包括筛选和排序；表操作包括导入，新建，复制，重命名，删除和编辑（相比于列操作，多了导入，而无移动，因为表没有次序）。其中，编辑是指对列或表的公式的定义与修改。任何操作和操作的组合都会产生合法的嵌套表。

进入首页，登录，点击 **Create MashUp**，可以见到图 6 所示的界面，其菜单结构如图 7 所示。其中部分菜单项的功能如下（其他菜单项将在后面说明）：

表 1. 菜单项功能说明

菜单项	功能
New	新建（空的）数据服务
Save	保存当前的操作，并封装为数据服务。
Save As	将当前的数据服务另存为副本，用户可以指定新的数据服务名称。
Rename	重命名当前的数据服务
Restore	撤销当前的操作，恢复到上次保存的状态
Undo/Redo	撤销/重做上一步操作
In Text/HTML	以文本/HTML 的方式查看嵌套表

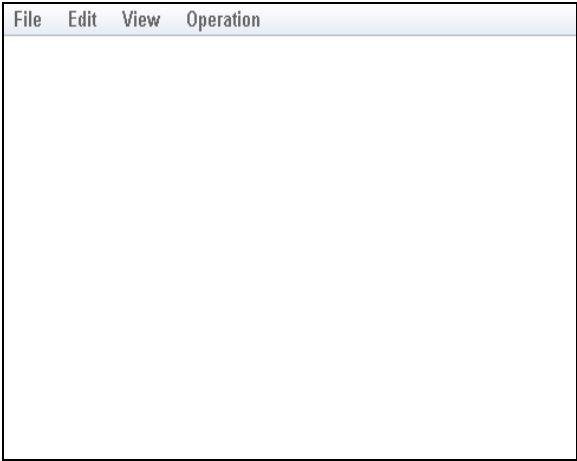


图 6. 数据服务组合用户界面

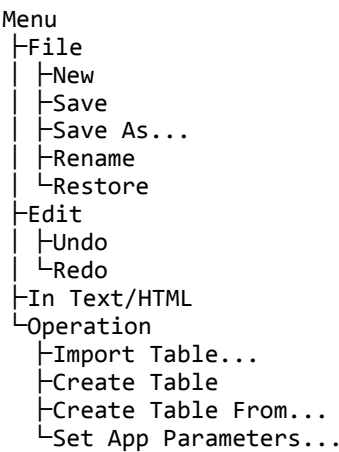


图 7. 数据服务组合菜单结构

3.1. 表操作

3.1.1. 导入(import)

导入表操作用于导入以常量方式定义的表。

操作过程：用户选择菜单项 **Operation->Import Table...**，在弹出的对话框中指定要导入的常量数据，然后实施导入。该数据将自动转换为嵌套表，并赋予自动生成的标题。

3.1.2. 新建(create)

新建表操作用于新建空表。

操作过程：用户选择菜单项 **Operation->Create Table** 实施新建表操作。一个新的空表将被创建，并赋予自动生成的标题。

3.1.3. 复制(copy)

复制表操作用于复制现有的表。

操作过程：用户右击要复制的表的标题，在右键菜单中选择 **Copy**。复制出的表与选定的表除标题被自动重命名外，其他属性（包括各列属性）完全相同。

3.1.4. 重命名(rename)

重命名表操作用于重命名表的标题。
操作过程：用户右击要复制的表的标题，在右键菜单中选择 **Rename**，然后在弹出的对话框中给出该表的新标题。新标题必须是合法的 XQuery 变量名，且与其他表标题不同。

3.1.5. 删除(delete)

删除表操作用于删除现有的表。
操作过程：用户右击要复制的表的标题，在右键菜单中选择 **Delete**。

3.1.6. 编辑(edit)

编辑表操作用于修改表上的公式。
操作过程：用户右击要复制的表的标题，在右键菜单中选择 **Edit**，然后在弹出的对话框中修改该表的公式。表公式的构造规则见 4.3 节。

3.2. 列操作

列操作都作用于某个列的级别上，它能够改变该级别的列的数量、次序和公式，但它不改变该级别列的行数和行次序。

3.2.1. 新建(create)

新建列操作用于新建空列。
操作过程：用户右击某列的标题，在右键菜单中选择 **Create**。随后，以该列为准，在其后会插入一个新的空列。新建的列与基准列的基数相同。
例如，用户在图 8 中分别以 **Model** 和 **Price** 作为基准新建列的结果如图 9 所示。

	Model	Detail		
		ID	Price	
1	Jetta	1 872	17000	
		2 901	16000	
2	Civic	1 879	15000	

图 8. 样例嵌套表

	Model	c0	Detail		
			ID	Price	c0
1	Jetta		1 872	17000	
			2 901	16000	
2	Civic		1 879	15000	

图 9. 新建列样例

3.2.2. 复制(copy)

复制列操作用于复制现有的列。
操作过程：用户右击任意一列的标题，在右键菜单中选择 **Copy**。复制出的新列将插入在原列的后面，它与选定的列除标题被自动重命名外，其他属性（包括它的公式与子列）完全相同。
例如，用户在图 8 中分别以 **Model** 和 **Price** 作为基准复制列的结果如图 10 所示。

	Model	ModelCopy0	Detail			
			ID	Price	PriceCopy0	
1	Jetta	Jetta	1 872	17000	17000	
			2 901	16000	16000	
2	Civic	Civic	1 879	15000	15000	

图 10. 复制列样例

3.2.3. 移动(move)

移动列操作用于移动列到同级的其他位置。
操作过程：用户用鼠标拖拽某列的标题到目标位置。目标位置必须与选定的列同级，否则操作无效。

3.2.4. 重命名(rename)

重名列操作用于重名列的标题。
操作过程：用户右击任意一列的标题，在右键菜单中选择 **Rename**，然后在对话框中给出该列的新标题。新标题必须是合法的 XML 名称，且与同级的其他列标题不同。

3.2.5. 删除(delete)

删除列操作用于删除现有的列。
操作过程：用户右击任意一列的标题，在右键菜单中选择 **Delete**。
例如，用户在图 8 中删除 **Price** 列的结果如图 11 所示。

	Model	Detail	
		ID	
1	Jetta	1	872
		2	901
2	Civic	1	879

图 11. 删除列样例

3.2.6. 解嵌套(unnest)

解嵌套列操作用于把用户选定的单个复合列解开，且复合列的每一行与其他行进行“交”操作。
操作过程：用户右击一个复合列，在右键菜单中选择 **Unnest**。
例如，用户在图 8 中对 **Detail** 解嵌套的结果如图 12 所示。

	Model	ID	Price
1	Jetta	872	17000
2	Jetta	901	16000
3	Civic	879	15000

图 12. 解嵌套样例

3.2.7. 嵌套(nest)

嵌套列操作用于把用户选定的多个相邻的列合并成复合列，并将其他列的内容相同的行合并为同一行。嵌套操作是解嵌套操作的逆操作。
操作过程：用户选择多个相邻的同级列，并右击其中任意一个列的标题，在右键菜单中选择 **Nest**。
例如，用户在图 12 中对 **ID**, **Price** 实施嵌套的结果如图 13 所示(与图 8 的区别在于 **Detail** 变成了自动命名的 **c0**)。

	Model	c0		
			ID	Price
1	Jetta	1	872	17000
		2	901	16000
2	Civic	1	879	15000

图 13. 嵌套样例

3.2.8. 解组(ungroup)

解组列操作用于把用户选定的单个复合列解开，用其子列替换之。用户选定的复合列将被其子列取代；由于复合列对应的子表可能有多行，在解组后只保留第一行。

操作过程：用户右击一个复合列，在右键菜单中选择 Ungroup。

例如，用户在图 8 中对 Detail 解组的结果如图 14 所示。

	Model	ID	Price
1	Jetta	872	17000
2	Civic	879	15000

图 14. 解组样例

3.2.9. 编组(group)

编组列操作用于把用户选定的一个或多个同级列编成组，形成一个复合列。编组操作与嵌套操作相同的地方在于：对表头相同的两个表进行操作之后其表头仍相同；不同之处在于：嵌套操作合并其他列的相同行，而编组不执行合并。

操作过程：用户选定一个或多个相邻的同级列，然后右击其中任意一列的标题，在右键菜单中选择 Group。用户选定的同级列将被一个复合列取代，该复合列包含用户选定的同级列。

例如，用户在图 12 中对 ID, Price 实施编组的结果如图 15 所示。

	Model	c0		
			ID	Price
1	Jetta	1	872	17000
2	Jetta	1	901	16000
3	Civic	1	879	15000

图 15. 编组样例

3.2.10. 编辑(edit)

编辑列操作用于修改列上的公式。列公式会作用于本列的每个行上，公式的上下文（用 ~表示）就是当前的行。

操作过程：用户右击一列，然后修改该列的公式。

例如，图 16 中列公式 $\sim\text{Price} \leq 16000$ 作用于 Low 的每个行上，各生成了布尔值；由于上下文就是当前的行，use 所代表的数据就是本行的 Price 值，因此生成的布尔值的意义就是本行的 Price 值是否小于等于 16000。又例如，average 列作用于 average 的每个行上，各生成了数值；由于上下文就是当前行，~Detail/Price 是本行上的所有 Price 值，它是一个序列，通过 avg 函数对该序列求平均值，就是本行上的 Price 值的平均值。

	Model	Average =avg(~Detail/Price)	Detail			
				ID	Price	Low =~Price<=16000
1	Jetta	16500	1	872	17000	false
			2	901	16000	true
2	Civic	15000	1	879	15000	true

图 16. 编辑列样例

3.3. 行操作

行操作能够改变行的数量和次序，但它无法对列做出改变。

3.3.1. 筛选(filter)

筛选行操作用于筛选特定的行。

操作过程：用户右击某列，然后在右键菜单中选择 Filter->Remove FALSE/TRUE。如果用户选择了 Remove FALSE，对于该列的每个值，如果其转换为布尔值的结果为 true，则它所在的行得以保留，否则则它所在的行被清除；如果用户选择了 Remove TRUE，则结果相反。行的清除是级连的，即，如果低级列的所有行都被清除，则高级列的相应行也会被清除。

例如，用户在图 16 中对 Low 实施 Filter->Remove TRUE 的结果如图 17 所示。

	Model	Average	Detail			
				ID	Price	Low
1	Jetta	16500	1	872	17000	false

图 17. 筛选样例

3.3.2. 排序(sort)

排序行操作用于对行排序。

操作过程：用户右击某列，然后在右键菜单中选择 Sort->Ascending/Descending 按升序/降序对该列所处子表的所有行排序。

如果该列为简单型，则把列上的数据视为字符串，按字典次序排序。排序算法是稳定的，即相同数据在排序后的相对次序不发生改变。排序只发生在同一子表的行中。例如，对图 16 按 use 列升序排序，结果如图 18 所示。如果该列为复合型，则依次按每个子列排序；如果子列中有复合型，则递归进行。

	Model	Average	Detail			
				ID	Price	Low
1	Jetta	16500	1	901	16000	true
			2	872	17000	false
2	Civic	15000	1	879	15000	true

图 18. 排序样例

3.4. 公式

公式用于从既有的数据生成新数据，并通过用户的编辑操作指定新数据的填充位置。公式分列公式与表公式两种：列公式通过编辑列操作用于填充指定的列，表公式通过编辑表操作用于填充指定的表。

3.4.1. 列公式

~用于表示当前行上下文。如果没有它，XPath 将无法区分 XPath 上下文和行上下文。列公式的类型下表所示。

表 2. 列公式

类型	描述	操作符或关键词
基本	常量，函数，括号表达式	
比较	通过值进行比较	=, !=, <, <=, >, >=, eq, ne, lt, le, gt, ge, is, <<, >>
条件	if-then-else 表达式	if, then, else
逻辑	布尔 and/or 操作符	and, or
路径	从 XML 或嵌套表中选取值	/
量化	确定某个序列是否满足给定条件	some, every, in, satisfy
算术	加，减，乘，除	+, -, *, div, idiv, mod

3.4.2. 表公式

表公式的类型如下表所示。

表 3. 表公式

类型	描述	操作符或关键词
函数	输出嵌套表数据的函数	N/A
笛卡尔积	表之间的笛卡尔积	^
集合	表之间的集合操作	union(), intersect, except
路径	从 XML 或嵌套表中选取值	/

3.4.3. 引用各级的属性元素

表 4. 数据引用

描述	操作符或关键词
表示当前行上下文	~
返回最上层	..
向下退一层	/ /

3.4.4. 函数

1. `replace(位置 , "[^\u4e00-\u9fa5]", "")`
例如：`replace(~Address, "[^\u4e00-\u9fa5]", "")`，表示将该列文本的标点全部去掉，只剩余有效文本信息。
规则：`replace(需要修改的文本位置 , "正则表达式", "")`。
例如：`replace(~Address, "[^\u4e00-\u9fa5]", "")`，`replace` 的第一个参数是需要修改的文本位置，第二个参数是一个正则表达式，`[^\u4e00-\u9fa5]`的作用是只保留列中汉字，把列中符号和英文字母全部去掉。也可以设计成别的正则表达式。
2. `wi:ds(“服务 ID”)`，调用 `id` 为 `ID` 的服务，里面可以嵌套其他公式。

3. `wi:ds("Name")`，调用名字为 `Name` 的服务。
4. `concat ("","")`，将两个字符串，或者字符串跟一个变量拼接起来。可以用于拼接参数。
5. `wi:ds(concat("ae346694-3e07-47dc-9e08-e9fcd99c5e8d?_q=", $q))`：带参数的数据服务调用。调用 ID 编号为 `ae346694-3e07-47dc-9e08-e9fcd99c5e8d` 的服务，并传参数，参数名为 `q`，值为自己设置的变量 `q`。
`wi:ds(concat("f72bee3e-82bf-4f86-834f-e0abbb97582a?_address=", ~Address))`，参数为变量
`wi:ds(concat("f72bee3e-82bf-4f86-834f-e0abbb97582a?_address=", "莱芜"))`，参数为常量
`wi:ds(concat("ae346694-3e07-47dc-9e08-e9fcd99c5e8d?_q=", $q, "ae346694-3e07-47dc-9e08-e9fcd99c5e8d?_n=", $n))`，带有多个参数的数据服务调用。

3.5. 自定义数据服务参数

在数据服务组合过程中，可以自定义复合数据服务的参数。

操作过程：选择菜单命令 `Operation->Set App Parameters...`，打开自定义参数对话框（如图 19 所示）。点击 `Add Parameter` 按钮即可添加自定义参数。用户需要提供参数名和参数的样例值；随后，在表公式和列公式中就可以引用该自定义参数，引用参数时，需要加 `$` 前缀（与 `XQuery` 参数的引用方式相同）。

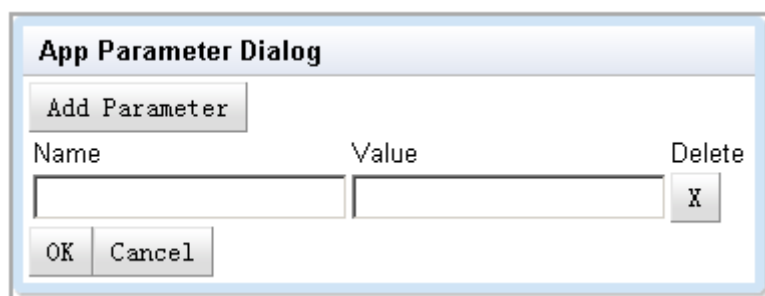


图 19. 自定义参数用户界面

3.6. 标记公开表和 KPI 数据

用户进行数据服务组合时可能涉及多个表，但是组合得到的数据服务只能输出其中的一个表，该表称为公开表。用户可以自定义公开表，操作方法为：右击表的标题，在右键菜单中选择 `Set Public`。

用户还可以将某些列的数据标记为 **KPI** 数据。操作方法为：在任意一列的标题上右击，在右键菜单中选择 `Set as KPI`。KPI 标记可以在提供数据服务目录的 **Web** 服务中被读取。标记了 **KPI** 列的表自动成为公开表。

4. 数据服务的使用

无论底层数据源是网页、WebAPI 还是数据库，一旦封装为数据服务，其访问方式都是一致的。

4.1.1. 通过网页访问

进入首页，登录，点击 `MyDataService`，可以看到自己创建的所有数据服务，如图 20

所示。点击数据服务的 Table 即可通过网页实现数据服务的网页访问。点击 Browse 可以浏览所有用户创建的数据服务，如图 21 所示。数据服务访问结果的网页按照系统设置的时间间隔（默认为 10 秒）刷新。

数据服务在访问时可以自定义参数。例如，2.1 创建的数据服务设定了运行时参数 wd，因此，其 Table 访问地址为：项目路径/*UUID*?paramEncoding=GBK&_wd=%B2%E2%CA%D4，其中%B2%E2%CA%D4 是默认的 wd 参数值（搜索关键词“中国”）的 URL 编码。用户可以将其更改为其他值的 URL 编码，从而获得其他关键词的搜索结果。

注意访问数据服务的自定义参数需要加前缀“_”。不加该前缀的参数（如 paramEncoding）是控制参数，为系统保留使用。

RestaurantMenu_20130605_1 Open Mode: Description: Domain: travel Created by: user Atomic(Composite): composite	Table	XML	JSON	Edit	Remove
Created on: 2013-6-5 15:30:38		Last Updated: 2013-6-5 15:30:38			
EnglishMenu_20130605_1 Open Mode: Description: Domain: travel Created by: user Atomic(Composite): atomic	Table	XML	JSON	Remove	
Created on: 2013-6-5 15:25:06		Last Updated: 2013-6-5 15:25:06			
Restaurant_20130605_1 Open Mode: Description: Domain: travel Created by: user Atomic(Composite): atomic	Table	XML	JSON	Remove	
Created on: 2013-6-5 15:23:08		Last Updated: 2013-6-5 15:23:08			
shujuku Open Mode: Description: Domain: environment Created by: user Atomic(Composite): atomic	Table	XML	JSON	Remove	
Created on: 2013-6-5 15:08:47		Last Updated: 2013-6-5 15:08:47			

图 20. 自己创建的数据服务

shujuku Open Mode: Description: Domain: environment Created by: user Atomic(Composite): atomic	Table	XML	JSON		
Created on: 2013-6-5 15:08:47		Last Updated: 2013-6-5 15:08:47			
caidan_tomeat Open Mode: Description: Domain: food Created by: user Atomic(Composite): atomic	Table	XML	JSON		
Created on: 2013-6-5 11:19:29		Last Updated: 2013-6-5 11:19:29			
RestaurantMenu_20130605 Open Mode: Description: combine english menu and restaurant Domain: food Created by: user Atomic(Composite): composite	Table	XML	JSON		
Created on: 2013-6-5 11:07:26		Last Updated: 2013-6-5 11:07:26			
EnglishMenu_20130605 Open Mode: Description: service about restaurant has english menu or not Domain: food Created by: user Atomic(Composite): atomic	Table	XML	JSON		
Created on: 2013-6-5 11:00:42		Last Updated: 2013-6-5 11:00:42			
restaurant_20130605 Open Mode: Description: service about restaurant Domain: food Created by: user Atomic(Composite): atomic	Table	XML	JSON		
Created on: 2013-6-5 10:26:41		Last Updated: 2013-6-5 10:26:41			

图 21. 所有用户创建的数据服务

4.1.2. 通过 API 同步编程访问

除了 HTML 格式外，数据服务也以 XML/JSON 格式发布，从而方便编程访问：

- XML 格式的访问地址为：项目路径/*UUID*.xml
- JSON 格式的访问地址为：项目路径/*UUID*.json

其中 *UUID* 为数据服务的 UUID。XML 和 JSON 文档可以通过 dom4j 和 Json-lib 等得以解析。