

Data Service Space User Manual

1. Introduction

Data Service Space (DSS) can encapsulate various data sources, such as Web page, WebAPI, relational database and so on. DSS can provide a uniform data model and access interface for Internet data with various formats. Furthermore, DSS can compose different data services into a new composite one.

DSS is a Web application. That is to say users can use all the functionalities through a browser, without installing any other client software.

2. Encapsulation of a data service

2.1. Encapsulation of a Webpage

1. Enter the website and login, move to CreateDataService menu, click Create Clippers and see Figure 1.
2. Input the URL address of the web page you want to encapsulate into the address bar, shown in the Figure 1, for example, <http://www.baidu.com/s?bs=%D4%C6%BC%C6%CB%E3&f=8&wd=%D6%D0%B9%FA>. Click the “*Open*” button to open the web page.
3. Mark the data content by mouse clicking. The attribute of labeled data content must be same. In this example, choose two titles of search results (red mark as Figure 2 shows). According to the red mark, the software will conjecture the same attribute data automatically (blue mark as Figure 2 shows). After labeling, click the “*Add Property*” button to add a property.
4. User can repeat the above steps until no more properties need to be added. Click the “*Preview*” button to check the encapsulation process of a data service.
5. After adding properties, click the “*Create*” button and pop-up a dialog box as Figure 3 shows. In the dialog box, user can input the name, type and description of the new data service. At last click the Create button and you have created a data service.
Click menu MyDataService, you can see the data service that created by yourself. Click menu Browse, you can see services created by all users.
For detailed instructions please see 4.1.1.

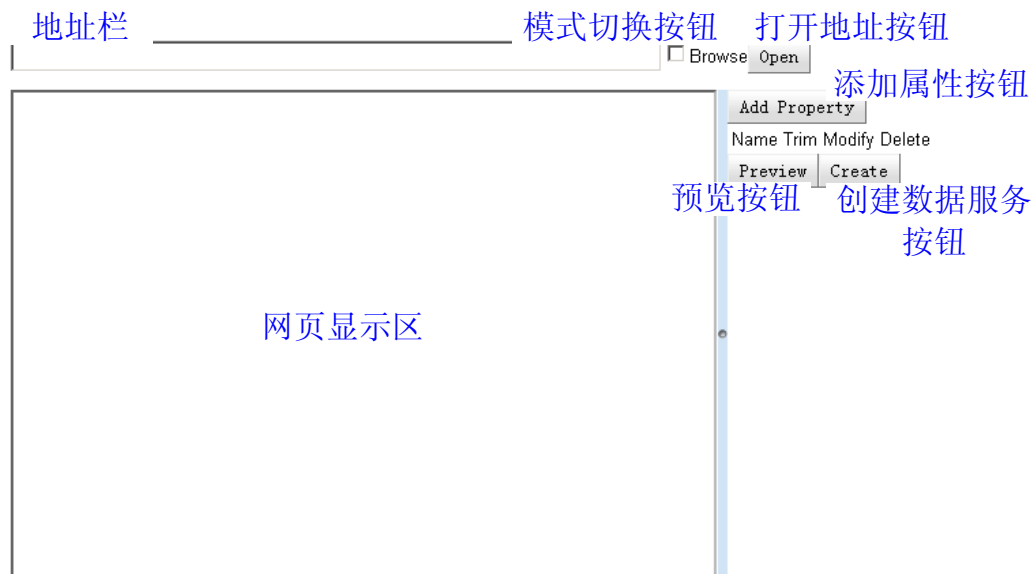


Figure 1. Web Page Labeling UI

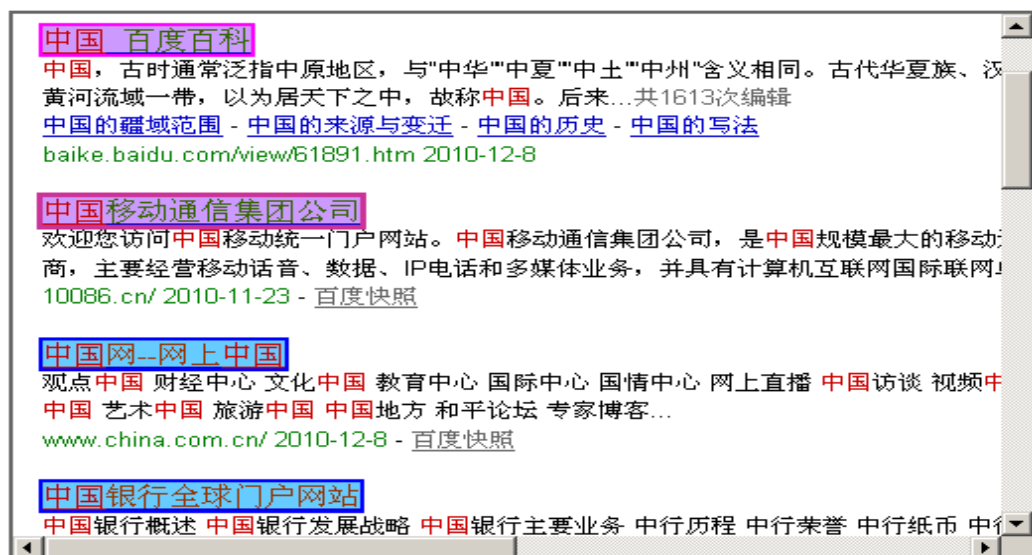


Figure 2. Sample Web Page

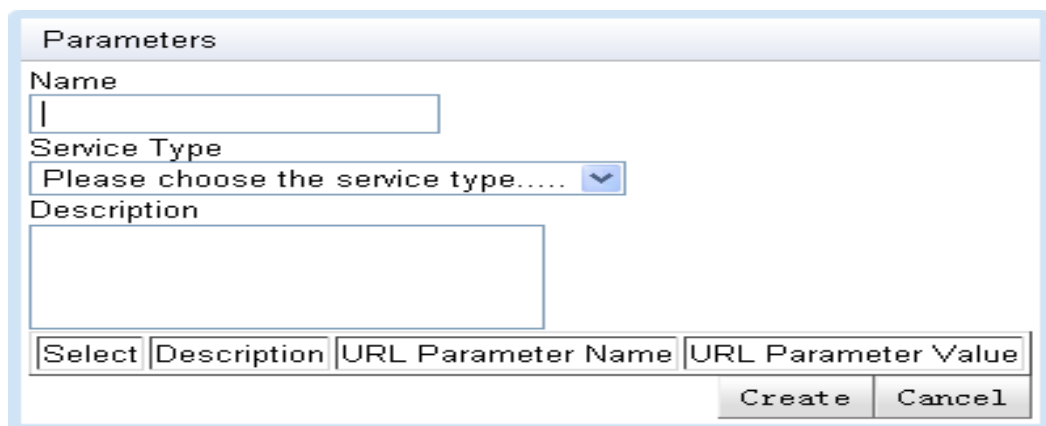


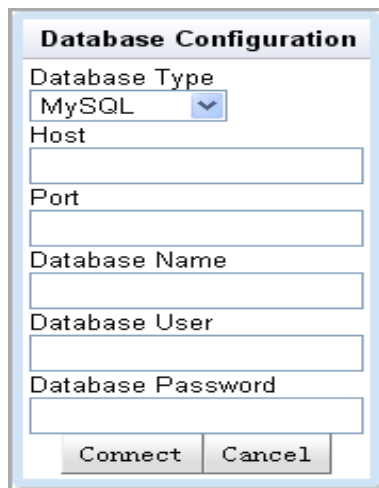
Figure 3. Data Service Saving Dialog

2.2. Encapsulation of a WebAPI

1. Enter the website and login, move to CreateDataService, click the CreateDataService Wrapper .
2. Input the Web API(support XML/JSON API) address in the address bar of Figure 4,. For example, we can input Google search API address like: <https://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=Paris%20Hilton&callback=processResults>
3. Click the “Load” button to check the encapsulation result.
4. Click the “Create” button and pop-up dialog box as Figure 3 shows. User can input the name of a data service, type of a data service and description. At last click the “Create” button to finish the creation of the data services.

2.3. Encapsulation of a Relational Database

1. Enter the website and login, move to CreateDataService, click the CreateDataService Wrapper and click the “Database” tab.
2. Click the “Connect” button, choose the database type from pull-down menu as Figure 4 shows. Add the database host IP, port, database name, database user, database password. We support three databases, namely: MySQL, SQL Server, Oracle.
3. Click “Connect” button, choose the database table from the pull-down menu that has been note as red as Figure 5.
4. Click the “Create” button and pop-up dialog box as Figure 3 shows. User can input the name of a data service, type of a data service and description. At last click the “Create” button to finish the creation of the data services.



Database Configuration

Database Type
MySQL

Host

Port

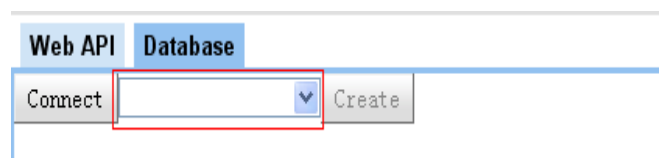
Database Name

Database User

Database Password

Connect Cancel

Figure 4. Database Configuration



Web API Database

Connect [Dropdown Menu] Create

Figure 5. Choose Table

3. Compose Data Services

The encapsulated data services can be composed into a new composite one. Data service can be seen as a data generator, which gets a data instance for each calling. Data instances can be expressed with XML/JSON, or visual nested tables. In DSS, the data service instances are modeled as a working set with parameters. The working set includes several nested tables, and each table is associated with a unique name. A table contains a set of columns and a column contains a set of cells. Each table, column and cell has its value and operations. The operation

allows users to process the current data in multiple granularities and derives new data by setting formulas interactively.

The operations can be classified into three categories. They are column operation, row operation and table operation. The column operations include new, copy, move, rename, delete, and edit. Row operations include filter and sort. Table operations include import, new, copy, rename, delete and edit. Edit means defining and modifying the formula of column or table. Any operation and operation combination can produce legal nested tables.

Enter the web site and login, click “*Create MashUp*” ,you will see Figure 6. The structure of menu is shown as Figure 7. Some functions of the menu item are below:

Table 1. Menu Item Function Description

Menu item	Function
New	New (empty) Data Service
Save	Save the current Data Service.
Save As	Save the current Data Service as copy.
Rename	Rename the current Data Service
Restore	Cancel the current operation, restore the last step operation.
Undo/Redo	Undo/redo last step operation
In Text/HTML	Use the text/HTML way to present nested tables

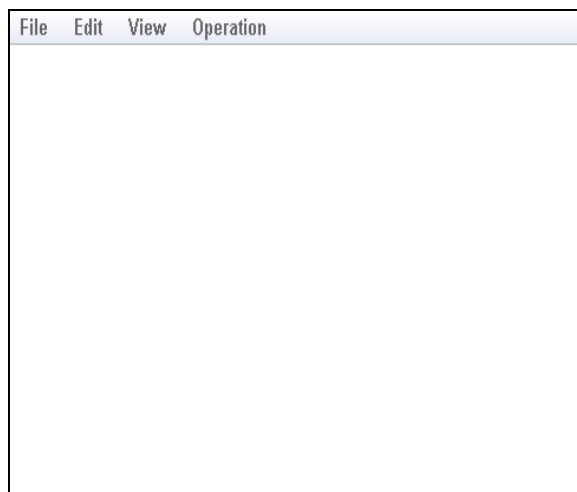


Figure 6. Data Service Composition Page

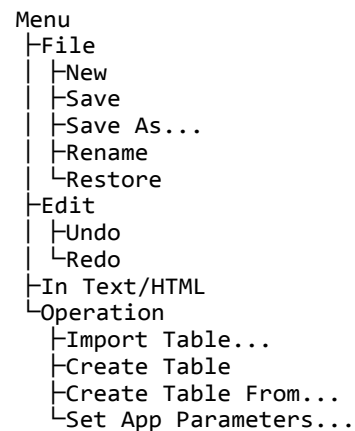


Figure 7. Data Services Composition Menu Structure

3.1. Table Operation

3.1.1. import

Import table operation is used to import table defined in a constant way.

Operation process: User selects the menu item Operation->Import Table..., then specifies the constants data in the pop-up dialog, and import. The data will be automatically converted to a nested table, and generated the title.

3.1.2. create

Create table operation is used to create a new empty table.

Operation process: User selects the menu item Operation->Create Table. A new table will be created and generated the title.

3.1.3. copy

Copy table operation is used to copy an existing table.

Operation process: Users right-click the title of the table, and select the Copy in the menu. The title of the copied table is automatically renamed compared with the selected table, but other properties (including each column attributes) are exactly the same.

3.1.4. rename

Rename table operation is used to rename table.

Operation process: Users right-click the title of the table, and select the Rename in the menu. Then write a new name in the pop-up dialog. New name must be legitimate XQuery variable name, and different from other tables' names.

3.1.5. delete

Delete table operation is used to delete an existing table.

Operation process: Users right-click the title of the table, and select the Delete in the menu.

3.1.6. edit

Edit table operation is used to modify formula of the table.

Operation process: Users right-click the title of the table, and select the Edit in the menu. Then modify formula in the pop-up dialog. The rules of table formula as shown in section 4.3.

3.2. Column Operation

Column operation is applied to a column level, and can change the number of columns, order and formula, but can not change the number of rows and row order in the column..

3.2.1. create

Create column operation is used to create a new empty column.

Operation process: Users right-click the title of the column, and select the Create in the menu. Then a new null column will be inserted behind selected column. The new column is the same as the selected column.

For example, in Figure 8 user respectively selects the Model and the Price to create new columns. The results are shown in Figure 9.

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

Figure 8. Nested Table Sample

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

Figure 9. New Column Sample

3.2.2. copy

Copy column operation is used to copy a new empty column.

Operation process: Users right-click the title of the column, and select the Copy in the menu. Then a new null column will be inserted behind selected column. The title of the copied column is automatically renamed compared with the selected column, but other properties (including formula and subsequence) are exactly the same.

For example, in Figure 8 user respectively selects the Model and the Price to create new columns. The results are shown in Figure 10.

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

Figure 10. Copy Column Sample

3.2.3. move

Move column operation is used to move the column to elsewhere at the same level.

Operation process: User drag and drop the title of the column to the target location with the mouse. Target location must be at the same level with selected column, otherwise operation is invalid.

3.2.4. rename

Rename column operation is used to rename the title of column.

Operation process: Users right-click the title of the column, and select the Rename in the menu. Then write a new name in the pop-up dialog. New name must be legitimate XQuery variable name, and different from other columns' names.

3.2.5. delete

Delete column operation is used to delete an existing column.

Operation process: Users right-click the title of the column, and select the Delete in the menu.

For example, in Figure 8 users delete the Price column. The results are shown in Figure 11.

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

Figure 11. Delete Column Sample

3.2.6. unnest

Unnest column operation is used to untie a composite column that users select, and each row of the composite column will intersect with other rows.

Operation process: Users right-click the title of the composite column, and select the Unnest in the menu.

For example, in Figure 8 users unnest the Detail column. The results are shown in Figure 12.

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

Figure 12. Unnest Sample

3.2.7. nest

Nest column operation is used to merge multiple adjacent columns into a composite column. Nest column operation is the inverse operation of unnest column operation.

Operation process: Users select multiple adjacent columns, and right-click the title of the any column, and select the Nest in the menu.

For example, in Figure 12 users nest the ID and Price columns. The results are shown in Figure 13. (the difference from Figure 8 is that the Detail is named c0 automatically)

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

Figure 13. Nest Sample

3.2.8. ungroup

Ungroup column operation is used to untie a composite column that users select. The composite column will be replaced with subsequence. Because subsequence may have multiple rows, only the first row will be retained after ungroup.

Operation process: Users right-click the title of the composite column, and select the Ungroup in the menu.

For example, in Figure 8 users ungroup the Detail column. The results are shown in Figure 14.

	Model	ID	Price
1	Jetta	872	17000
2	Civic	879	15000

Figure 14. Ungroup Sample

3.2.9. group

Group column operation is used to merge multiple columns into a composite column. The similarity of group column operation and nest column operation : After operating two tables that have the same header, the header is still the same. The difference: Nest column operation merges the same rows of other columns, but group column operation does not.

Operation process: Users select multiple adjacent columns, and right-click the title of the any column, and select the Group in the menu.

For example, in Figure 12 users group the ID and Price columns. The results are shown in Figure 15.

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

Figure 15. Group Sample

3.2.10. edit

Edit column operation is used to modify formula of the column.

Operation process: Users right-click the title of the table, and select the Edit in the menu. Then modify formula in the pop-up dialog.

For example, in Figure 16 the formula “~Price<=16000” applies to each row of “Low”, and gets boolean value that means if the value of Price is less than or equal to 16000. For another example, the formula “avg(~Detail/Price)” applies to each row of “Average”, and gets average value that means the average for each row of Price.

	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

Figure 16. Edit Column Sample

3.3. Row Operation

Row operation can change the number and the sequence of rows, but it can't change columns.

3.3.1. filter

Filter row operation is used to filter specific row.

Procedure: Right-click at any column, and then in the right-click menu, select Filter-> Remove FALSE / TRUE. If user selects Remove FALSE, for each value of the column, it is converted to a Boolean value to true, then the line it will be preserved, otherwise the line will be cleared; If user selects Remove TRUE, the result will be the opposite. Clear lines are cascaded, this is, if all the low column are cleared, the high column corresponding row will be cleared.

For example, a user in Figure 16 implements the Low Filter-> Remove TRUE results showing in Figure 17.

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

Figure 17. Filter Sample

3.3.2. sort

Sorting line is used for sorting rows.

Procedure: Right-click a column, then in the context menu, select Sort-> Ascending / Descending ascending / descending sorting for all the rows of child table which the columns in.

If the column is a simple type, put the column data as strings, sorting according to the order of dictionary. Sorting algorithm is stable, this is, after sorting the relative order is not changed. Sorting occurs only in the same row of child table.

For example, sort the column of Table 14 column in ascending order, the results showing in Figure 16. If the column is complex, then sorting by each sub-column; if the child column has a composite type, then sort recursively.

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

Figure18. Sort Sample

3.4. Formula

Formulas are used to generate a new data from an existing data and fill the specified position for the new data through “edit” operation. Formulas are classified into Column Formula and Table Formula: Column Formula is used to fill specified columns through “edit” operation, Table Formula is used to fill the specified tables through “edit” operation.

3.4.1. Column Formulas

The classification of column formulas is shown in the following table:

Table 2. Column formula

Type	Description	Operator or Key word
Base	Constants, functions, expressions in parentheses	
Comparison	Compare values	=, !=, <, <=, >, >=, eq, ne, lt, le, gt, ge, is, <<, >>
Condition	if-then-else expression	if, then, else
Logic	Boolean and/or operators	and, or
Path	Select a value from XML or nested table	/

Quantization	determine whether a sequence satisfy a given condition	some, every, in, satisfy
Arithmetic	Addition, subtraction, multiplication, division,	+, -, *, div, idiv, mod

3.4.2. Table Formula

The classification of table formulas is shown in the following table:

Table 3. Table Formula

Type	Description	Operator or Key word
Function	Output a function of nested table	N/A
Cartesian product	Cartesian product between tables	^
Collection	Set operations between tables	union(), intersect, except
Path	Select a value from XML or nested table	/

3.4.3. Reference property element levels

The formula “~” is used to represent the current line context. Without this formula, XPath will unable to distinguish between the Xpath context and the line context. The following table shows the formula to reference the current line, the top level and the child level.

Table 4. Data Reference

Desription	Operator or Key word
Indicate the current line context	~
Back to the top level	..
Down to the child level	/ /

3.4.4. Some functions

1. replace(location , "[^u4e00-u9fa5]", "")
e.g. : replace (~Address, "[^u4e00-u9fa5]", ""), remove all the punctuation of the column text, remaining valid text message only.
Rule: replace(need to modify the text position , "regex", "")
For example: replace(~Address, "[^u4e00-u9fa5]", ""), replace the first parameter is the position the text need to modify, the second parameter is regex, the function of [^u4e00-u9fa5] is to remain the Chinese characters of the column, remove all the symbols and English letters of the column. Also be designed into other regular expressions.
2. wi:ds(“Service ID”), call id as ID service, it can nest other formulas inside.
3. wi:ds(“Name”), service call named “name”.
4. concat (‘’, ‘’) , stitch two strings or stitch a string with a variable quantity. Can be used to stitch parameters.
5. wi:ds(concat("ae346694-3e07-47dc-9e08-e9fcd99c5e8d?_q=", \$q)): service call with parameter data. Call service which ID number is ae346694-3e07-47dc-9e08-e9fcd99c5e8d, and pass parameters, parameter named q, the

value is the value of q.

wi:ds(concat("f72bee3e-82bf-4f86-834f-e0abbb97582a?_address=",~Address)),the parameter is variable quantity.

wi:ds(concat("f72bee3e-82bf-4f86-834f-e0abbb97582a?_address=", "laiwu")), argument is a constant.

The data service call can has more than two parameters. For example,wi:ds(concat("ae346694-3e07-47dc-9e08-e9fcd99c5e8d?_q=",\$q,"ae346694-3e07-47dc-9e08-e9fcd99c5e8d?_n=",\$n)) .

3.5. Custom Data Service Parameters

In the process of data services composition, you can customize the parameters of a composite service.

Procedure: Select the menu command Operation-> Set App Parameters ..., open the Customize Parameters dialog box (Figure 17). Click the Add Parameter button to add a custom parameter. Users need to provide parameter name and sample value; Subsequently, the formula in the table and column formulas can reference the custom parameters, when reference parameters, you need to add \$ prefix (and references in the same way with XQuery).

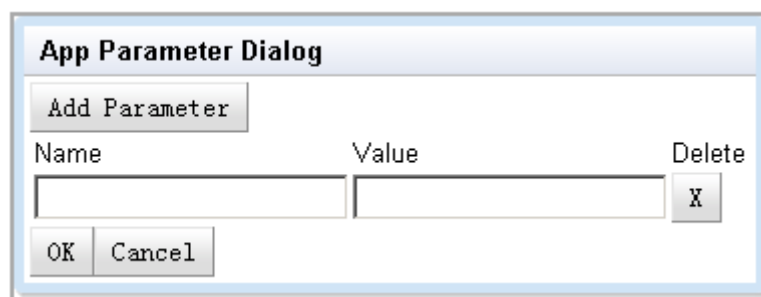


Figure 19. Custom Parameters Page

3.6. Save the composite data service

When user combines data services, many tables may be involved, but composite data service can only output one table. This table must be assigned as the public table. Users can customize the public table as following: Right-click the table title, in the context menu selecting Set Public. After that, the user can save the composite data service.

4 . The use of data service

Regardless of the underlying metadata is a website, WebAPI or database, if it encapsulates as data services, the access methods will be consistent.

4.1.1. Access via the web

Enter the website and login, click MyDataService. You can see a list of services as shown in Figure 20. Click “Table”, the data service execution results with the default input parameter values will be presented in the form of HTML. Click Browse, you can see all the data service created by all users as showing in Figure 21.

You can also customize the parameters when access a data service. For example, the created data services in 2.1 has an input parameter wd (search key word), so its access address to HTML

Table is: Project Path /UUID?paramEncoding=GBK&_wd=% B2% E2% CA % D4, % B2% E2% CA% D4 is the default parameter values of wd(search keywords "China") URL encoding. Users can change it to other URL value to obtain search results of other keyword.

Please add "_" before custom parameters. Without "_" the parameter (e.g. paramEncoding) will be a control parameter, and will be reserved for the system.

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 15:25:08		Last Updated: 2013-6-5 15:25:08	
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	

Figure 20. Data Service Created by User

shujuku Open Mode: Description: Domain: environment Created by: user Atomic(Composite): atomic	Table	XML	JSON		Last Updated: 2013-6-5 15:08:47
caidan_tomeat Open Mode: Description: Domain: food Created by: user Atomic(Composite): atomic	Table	XML	JSON		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		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		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		Last Updated: 2013-6-5 10:26:41

Figure 21. Data Service Created by All Users

4.1.2. Sync programmatic access through the API

In addition to HTML format, data service also publishes as XML / JSON format, making programming easy:

The address of XML is : Project Path/UUID.xml

The address of JSON is: Project Path /UUID.json

UUID is the UUID of data service . XML and JSON documents can be resolved by dom4j and Json-lib, etc.