数据格式化

1. 描述

在应用中，原始的数据格式经常不能满足实际需求，需要将数据进行格式化处理使其更符合需要。

比如下图中所示，将订购日期改成“年-月-日”格式，在订单金额前面加上货币符号“￥”并去掉小数点后数字。

订单编号	购买数量	订购日期	订购日期	订单金额	订单金额
DD000045	13	2017/3/10 0:00:00	2017/3/10	901.55	¥902
DD000046	7	2017/10/25 0:00:00	2017/10/25	179.2	¥179
DD000046	7	2017/10/25 0:00:00	2017/10/25	132.72	¥133
DD000047	19	2017/6/7 0:00:00	2017/6/7	771.21	¥771
DD000047	5	2017/6/7 0:00:00	2017/6/7	423.15	¥423
DD000047	10	2017/6/7 0:00:00	2017/6/7	940.9	¥941
DD000047	20	2017/6/7 0:00:00	2017/6/7	1231.2	¥1,231
DD000048	15	2017/12/24 0:00:00	2017/12/24	900.9	¥901
DD000048	15	2017/12/24 0:00:00	2017/12/24	627	¥627
DD000048	6	2017/12/24 0:00:00	2017/12/24	28.8	¥29

上图中的格式化效果可以通过属性面板中的“数据格式”设置项实现，也可以通过函数表达式来实现。

表达式中常用的数据格式化函数有两个：.ToString和Format。

本节介绍如何使用属性设置和数据格式化函数对数据进行格式化处理。

2. 属性设置与格式化函数

操作步骤

1

新建报表，添加报表数据源后，使用表格组件按照下图所示绑定数据字段。

2

为了方便对比转换效果，在“订购日期”和“订单金额”右侧分别增加两列，并绑定同样的数据字段。

3

在属性设置面板中的“数据格式化”对“订购日期”和“订单金额”进行格式化设置。

4

使用Format函数或.ToString函数进行数据格式化，比如将订购日期显示成“yyyy/MM/dd”格式，将订单金额显示成货币格式并且不显示小数点后数字。

如果使用Format函数，则表达式为“ {Format(订购日期,"yyyy/MM/dd")} 和 {Format(订单金额,"c0")}；

-

订购日期	订购日期	订单金额	订单金额	订单金额
{订购日期}	{Format(订购日期, "yyyy/MM/dd")}	{订单金额}	{订单金额}	{Format(订单金额, "c0")}
{Count(订购日期)}	{Count(订购日期)}	{Sum(订单金额)}	{Sum(订单金额)}	{Sum(订单金额)}

5

预览报表，可见通过这两种方式对数据进行格式化的效果是一致的。

订单编号	购买数量	订购日期	订购日期	订购日期	订单金额	订单金额	订单金额
DD000045	13	2017/3/10 0:00:00	2017/3/10	2017/03/10	901.55	¥902	¥902
DD000046	7	2017/10/25 0:00:00	2017/10/25	2017/10/25	179.2	¥179	¥179
DD000046	7	2017/10/25 0:00:00	2017/10/25	2017/10/25	132.72	¥133	¥133
DD000047	19	2017/6/7 0:00:00	2017/6/7	2017/06/07	771.21	¥771	¥771
DD000047	5	2017/6/7 0:00:00	2017/6/7	2017/06/07	423.15	¥423	¥423
DD000047	10	2017/6/7 0:00:00	2017/6/7	2017/06/07	940.9	¥941	¥941
DD000047	20	2017/6/7 0:00:00	2017/6/7	2017/06/07	1231.2	¥1,231	¥1,231
DD000048	15	2017/12/24 0:00:00	2017/12/24	2017/12/24	900.9	¥901	¥901
DD000048	15	2017/12/24 0:00:00	2017/12/24	2017/12/24	627	¥627	¥627

结束

从格式化的步骤中可以看出，进行数据格式化的关键在于格式化参数的设定，如示例中使用了"yyyy/MM/dd"和"c0"。

除了这两个以外，还可以使用下表中更多的格式化参数。

比如：

.ToString(“n2")可以将数据转换为带有千分位标识的数字格式，并且小数点后保留两位。如果将n2改为n0，则表示不保留小数位数。

.ToString(“p2")可以将数据转换为百分数格式，并且小数点后保留两位。如果将p2改为p0，则表示不保留小数位数。

类型	字母	含义
数字类型	“C”或“c”	货币
	“D”或“d”	数值
	“E”或“e”	科学计数法
	“F”或“f”	固定小数点
	“G”或“g”	常规
	“N”或“n”	数字
	“P”或“p”	百分比
	“R”或“r”	往返过程
	“X”或“x”	十六进制
日期时间类型	"d"	短日期
	"D"	长日期

"f"	完整日期/短时间
"F"	完整日期/长时间
"g"	普通日期/短时间
"G"	普通日期/长时间
"t"	短时间
"T"	长时间
“M”或“m”	月/日模式
“O”或“o”	往返日期
“R”或“r”	RFC1123 模式
“s”	可排序日期
“u”	通用可排序日期
“U”	通用完整日期
“Y”或“y”	年月

3. 格式化函数示例

一些场景中不能使用属性设置进行格式化处理时，如处理表达式中的数据格式时，可以使用格式化函数来实现。

例如为表格添加一个表头，以特定的数据格式来显示查询报表的时间和销售总额，如下图所示。此时不能直接通过属性设置来实现目标效果，只能使用格式化函数修改表达式中数据的格式。

使用的表达式为：报表生成日期：{Format(Today(), "yyyy年MM月dd日")}，订单总金额为：{Format(sum(订单金额), "c2")}

报表生成日期：2021年03月19日，订单总金额为：¥481,412.81					
订单编号	购买数量	订购日期	订购日期	订购日期	订单金额
DD000045	13	2017/3/10 0:00:00	2017/3/10	2017/03/10	901.55
DD000046	7	2017/10/25 0:00:00	2017/10/25	2017/10/25	179.2
DD000046	7	2017/10/25 0:00:00	2017/10/25	2017/10/25	132.72
DD000047	19	2017/6/7 0:00:00	2017/6/7	2017/06/07	771.21
DD000047	5	2017/6/7 0:00:00	2017/6/7	2017/06/07	423.15
DD000047	10	2017/6/7 0:00:00	2017/6/7	2017/06/07	940.9
DD000047	20	2017/6/7 0:00:00	2017/6/7	2017/06/07	1231.2
DD000048	15	2017/12/24 0:00:00	2017/12/24	2017/12/24	900.9

回到顶部