# 文档05 ruiLibrary封装类库的介绍

## rui.configHelper - 系统配置项

提供了系统配置项的读取方法

提供了系统开发过程中常用数据值的定义。

提供了的相关方法：

1）获取appSettings中配置项的值

string rui.config.getAppValue(string configKey)

2）获取connectionStrings中连接串的connectionString的属性值

string rui.config.getConnString (string connName)

3）获取connectionStrings中连接串的providerName的属性值

string rui.config.getConnProvider (string connName)

|  |  |  |
| --- | --- | --- |
| 包含的变量 | 默认值 | 作用 |
| rui.config.va请选择Text | "=请选择=" | 下拉框请选择Text值 |
| rui.config.va请选择Value | "" | 下拉框请选择Value值 |
| rui.config.dateFormatLang | "yyyy-MM-dd HH:mm:ss" | 长日期格式串 |
| rui.config.dateFormat | "yyyy-MM-dd" | 短日期格式串 |
| rui.config.pageSize | Web.config对应键的值 | 默认分页大小 |
| rui.config.pageSizeDlg | Web.config对应键的值 | 对话框默认分页大小 |
| rui.config.pageSizeSelect | Web.config对应键的值 | 选择框默认分页大小 |
| rui.config.pageSizeMax | 1000000 | 最大分页大小 |
| rui.config.platformName | Web.config对应键的值 | 系统平台名称 |
| rui.config.ctxKey | Web.config对应键的值 | 控制JS强制刷新的变量 |
| rui.config.adminName | Web.config对应键的值 | 超级管理员默认名称 |
| rui.config.isOnline | Web.config对应键的值 | 判断项目是否上线运行，用于相关代码执行控制，只有上线后或未上线执行某些代码。 |

## rui.jsonResult - Json辅助方法

MVC开发框架中，所有的POST操作都是通过AJAX进行提交的，提交返回的结果都是JSON格式。我们规定了所有的JSON返回结果都是如下的格式：

{

result:true/false

message:返回的消息

其它附加信息(键值对形式)

}

result值为true代表业务操作成功了，result值为false代表业务操作不成功。关于成功或者失败的提醒消息都放在message中，如果请求的返回结果中还要携带其它信息，则可以继续添加。

为了能够快速的构建该JSON结果，提供了jsonResult类。

1. 保存要转换为JSON的键值对数据

public Dictionary<string, string> data { get; set; }

1. 构造JSON常用返回结果

Dictionary<string, string> rui.jsonResult. getAJAXResult(string message, bool success,

Dictionary<string, string> returnDic = null)

1. 向data中添加键值对的方法：参数key/Value

void rui.jsonResult.addValue(string keyName, string keyValue)

1. 向data中添加键值对的方法：参数：Dictionary<string, string>

void rui.jsonResult.addDic(Dictionary<string, string> dic)

1. 将DataTable中的数据转为Json字符串数据,并返回

string rui.jsonResult.dataTableToJsonStr(DataTable table)

DataTable中要求包含code和name两个字段的值。该方法会把DataTable里边的数据转为Json字符串格式，并返回

1. 向data中添加DataTable的Json字符串数据

void rui.jsonResult.addTable(string keyName, DataTable dtData)

DataTable中要求包含code和name两个字段的值。该方法会把DataTable里边的数据转为Json字符串格式，并添加到data中。

1. 将SelectListItem集合中的数据转为Json字符串数据

string rui.jsonResult.SelectListToJsonStr(List<SelectListItem> list)

1. 将rowID的值放到Dictionary<string, string>中并返回

void Dictionary<string, string> rui.jsonResult.getDicByRowID(object rowID)

返回一个Dictionary，内部包含:rowID/rowID的值

1. 将两个Dictionary合并为一个，并返回合并之后的Dictionary

Dictionary<string, string> rui.jsonResult.mergeDictionary(Dictionary<string, string> dic1,Dictionary<string, string> dic2)

## rui.excptHelper - 异常处理

包含了获取异常信息和抛出异常的功能。

1. 封装了自定义的异常类型

ruiException，自定义异常主要用于抛出对用户的提示。

1. 抛出指定错误的异常:这里抛出的是自定义的异常对象ruiException

Void rui.excptHelper.throwEx(string msg)

抛出一个定义的异常错误提示

1. 获取异常的错误信息

string rui.excptHelper.getExMsg(Exception ex)

获取异常的错误描述信息，并返回

## rui.logTools - 日志组件

为了程序运行过程中记录程序的运行结果，提供日志组件封装，记录的内容会被写到log目录内，每天一个文件。

1. 只记录文本内容，如果上线后，就不记录了

Void rui.logTools.log(string message)

1. 记录异常，并可以决定是否继续排除异常，一般业务类中的需要继续抛出给Action，Action中的异常不会不要继续抛出了。

Void rui.logTools.log(Exception ex,bool isThrow=false)

1. 记录文本和异常，如果异常是自定义的异常不进行记录

Void rui.logTools.log(string message, Exception ex, bool isThrow = false)

1. 输出调试信息，上线后还输出

Void rui.logTools.debug(string message)

## rui.typerHelper - 类型辅助相关

数据类型之间的相互转换，主要解决类型转换过程不合法会抛出异常问题，通过封装方法可以返回指定的默认值。

### 日期类型的转换方法

1. 转为日期字符串（默认短格式，日期格式见系统配置）

string rui.typeHelper.toDateTimeString(object value, bool islang = false)

1. 转为日期字符串（指定格式）

string rui.typeHelper.toDateTimeFormat(object value, string format)

1. 转为日期类型（允许返回Null值）

DateTime? rui.typeHelper.toDateTimeAllowNull(object value)

1. 转为日期类型（如果不合法则返回当前时间）

DateTime rui.typeHelper.toDateTime(object value)

1. 获取某个日期所在月份的开始时刻

string rui.typeHelper.getMonthStart(object value)

1. 获取某个日期所在月份的结束时刻

string rui.typeHelper.getMonthEnd(object value)

1. 获取某天时间的开始时刻

string getDayStart(object value)

1. 获取某天时间的结束时刻

string getDayEnd(object value)

1. 获取某个日期所在周的第一天

string rui.typeHelper.getWeekFirstDay(DateTime datetime)

### 字符串转换

1. 数据转为字符串，如果Null则返回空字符串。

string rui.typeHelper.toString(object value)

1. 数据转为字符串，允许返回Null

string rui.typeHelper.toStringAllowNull(object value)

### 数值类型的转换方法

1. 转为Decimal，如果不合法返回0

decimal rui.typeHelper.toDecimal(object value)

1. 转为Decimal，如果不合法返回Null

decimal rui.typeHelper.toDecimalAllowNull(object value)

1. 转为Boolean，如果不合法返回False

bool rui.typeHelper.toBoolean(object value)

1. 转为Int，如果不合法返回0

int rui.typeHelper.toInt(object value)

1. 转为Int，如果不合法返回Null

int rui.typeHelper.toIntAllowNull(object value)

1. 转为Long，如果不合法返回0

long rui.typeHelper.toLong(object value)

1. 转为Long，如果不合法返回Null

long rui.typeHelper.toLongAllowNull(object value)

### 浮点数相关处理

1. 移除浮点数后边多余额的0

string rui.typeHelper.removeZero(object value, bool returnEmptyStr = true)

用途说明：如果数据库里边定义的浮点数保留6位小数位，如果不处理，展示的时候会把6位都显示出来，并且后边几位很多情况都是0，通过该方法移除多余的0，展示起来好看一些。

1. 浮点数进行四舍五入，可以设定保留的位数

decimal rui.typeHelper.va四舍五入(object value, int length)

1. 浮点数四舍五入为整数

int rui.typeHelper.va四舍五入(object value)

1. 返回浮点数指定小数位的字符串

string rui.typeHelper.va小数位(decimal value, int length)

用途说明：不尽兴四舍五入，直接进行小数位截取。

1. 小数转为百分比

string rui.typeHelper.toPercent(object value)

用途说明：需要显示xx%形式的使用用。0.56显示为56%

### 判空的相关方法

1. 判断是否为空或者Null，空返回True，否则False

bool rui.typeHelper.isNullOrEmpty(object value)

1. 判断是否为空或者Null，如果是则抛出异常

void rui.typeHelper.isNullOrEmpty(object value, string msg)

1. 判断是否非Null，非空则返回True，否则False

bool rui.typeHelper.isNotNullOrEmpty(object value)

### 数据累加

1. 整数累加，如果其中有不合法的数据，则以零处理。

int rui.typeHelper.addInt(object val1, object val2)

1. 整数累减，如果其中有不合法的数据，则以零处理。

int rui.typeHelper.subInt(object val1, object val2)

1. 浮点数累加，如果其中有不合法的数据，则以零处理。

decimal rui.typeHelper.addDecimal(object val1, object val2)

1. 浮点数累减，如果其中有不合法的数据，则以零处理。

decimal rui.typeHelper.subDecimal(object val1, object val2)

## rui.requestHelper - 请求辅助方法

1. 获取提交变量的值（单个）

string rui.requestHelper.getValue(string key, bool isShowError = false, string errorMsg = "要获取的参数不存在")

1. 获取提交变量的值（多个返回List<string>()）

List<string> rui.requestHelper.getList(string name)

用途说明：如果前端是表单数组，同一个变量会提交多个值，返回值的数组。

1. 获取提交变量的值（多个返回：a,b,c,形式）

string rui.requestHelper.getMulValue(string key, bool isShowError = false, string errorMsg = "要获取的参数不存在")

用途说明：如果前端是表单数组，同一个变量会提交多个值，返回：a,b,c,形式。

1. 获取所有请求的变量的键集合

List<string> rui.requestHelper.getAllKey(string entryPreTag = "")

1. 判断是否是第一次访问该页面

bool rui.requestHelper.isFirstRequest()

1. 判断是否换页请求

bool rui.requestHelper.isPagerRequest()

1. 判断是否点击搜索请求

bool rui.requestHelper.isSearchRequest()

1. 判断是否AJAX请求

bool rui.requestHelper.isAjax()

1. 是否人工点击保存按钮的判断，主要用于判断是不是人工点击保存

bool rui.requestHelper.isClickSave()

用途说明：有的时候需要程序自动保存，如果是程序自动保存需要给后台传递一个auto=true的参数，该方法主要通过是否有该参数进行判断。

## rui.sessionHelper - session操作辅助类

封装了常见的Session操作

1. 向Session中添加变量和值

void rui.sessionHelper.saveValue(string sessionName, object value)

1. 从Session中获取某个变量的值，如果不存在是否抛出异常

string rui.sessionHelper.getValue(string sessionName, bool isThrowOnNull = false)

1. 从Session中获取某个变量的值，如果不存在是否抛出异常

T rui.sessionHelper.getValue<T> (string sessionName, bool isThrowOnNull = false)

1. 判断Session中是否有某个变量

bool rui.sessionHelper.hasKey(string sessionName)

1. 从Session中移除某个变量

void rui.sessionHelper.removeKey(string sessionName)

1. Session取消

void rui.sessionHelper.exit()

## rui.excelHelper - excel导入和导出

封装了Excel的导入和导出，内部采用了aspose.cell实现。

1. 把excel文件内容读取到DataTable中并返回

DataTable rui.excelHelper.excelToDataTable("excel文件绝对路径")

1. 将DataTable内的内容导出到Excel指定SheetName中

MemoryStream rui.excelHelper.dataTableToExcel(DataTable, "SheetName名称")

导出的是excel的流，还需要自己进行存储操作

## rui.webDiskHelper - 站点内文件操作

1. 获取网站的部署位置（网站根目录的绝对路径）

string rui.webDiskHelper.getWebPath()

1. 获取站内目录的物理路径

string rui.webDiskHelper.getAbsolutePath(string webInnerPath, string fileName = "")

1. 保存上传的文件，并返回文件的保存站内路径集合

List<string> rui.webDiskHelper.saveUploadFiles(string webInnerPath, string nameKey, string filename = "")

1. 文件复制

void rui.webDiskHelper.copyFile(string webInnerSourPath, string webInnerDescPath, bool isOverWrite)

1. 文件删除

void rui.webDiskHelper.deleteFile(string webInnerPath)

1. 判断文件是否存在

bool rui.webDiskHelper.isFileExist(string webInnerPath)

## rui.stringHelper - 字符串常用操作

1. 生成编号（返回下一个可用编号）

string rui.stringHelper.codeNext(string code, int NumLength)

使用举例：4代表编码中数值为的长度，根据数值位获取下一个数值对应编号。

codeNext(”DF0001”,4)，返回”DF0002”

1. 取子串，如果字符串长度不满足length，则直接返回原字符串。

string rui.stringHelper.subString(object value, int length)

1. 去除字符串最后一个字符，如果原字符串为空，则不处理。

string rui.stringHelper.removeLastChar(string str)

1. 字符串包含内容合并

string rui.stringHelper.unionString(string val1, string val2, char splitTag)

1. 字符串包含内容排除

string rui.stringHelper.exceptString(string val1, string val2, char splitTag)

## rui.dbTools - 数据库操作辅助方法

包含数据库操作过程中相关的辅助方法

### DataTable相关

1. DataTable转换方法，将DataTable的列类型设置为字符串类型，并设置为可写（默认只读），当需要修改DataTable中数组类型的列的时候，需要进行转换。

DataTable rui.dbTools.tranferDataTable (DataTable table)

1. 从DataTable中移除列

DataTable rui.dbTools. removeShowColumn(DataTable table, string removeColumns)

1. 给DataTable的最左边增加序号列，方便展示

DataTable rui.dbTools. insert序号(DataTable table, int startIndex = 0)

1. 将DataRow数组转为DataTable

DataTable rui.dbTools.toDataTable(DataRow[] rows)

1. 将Dictionary转为DataTable

DataTable rui.dbTools.getTableByDic(Dictionary<string, string> dic)

1. 检查DataTable中某列是否存在某个值

bool rui.dbTools.checkExist(DataTable table, string columnName, string value)

1. DataTable数据复制

bool rui.dbTools.copyDataTable(DataTable sourTable, DataTable descTable)

1. 两个DataTable进行合并，需要设定那个是主键字段

DataTable rui.dbTools.margeDataTable(DataTable table1, DataTable table2, string keyName)

1. DataTable内日期列转为短格式展示

DataTable rui.dbTools.dateFormat(DataTable table, string exceptFieldName = "")

1. 设置DataTable列的值允许为null

DataTable rui.dbTools.setAllowNull(DataTable table)

1. 获取DataRow中某列值

string rui.dbTools.getRowValue(DataRow row, string fieldName, bool isThrow = true)

### In表达式相关

1. 将List集合内的值转为In表达式，用于拼接sql的in表达式值

string rui.dbTools.getInExpression(List<string> list, bool handleEmpty = true)

1. 将DataTable内的某列值转为In表达式

string rui.dbTools.getInExpression(DataTable table, string columnName, bool handleEmpty = true)

1. 将逗号分隔的字符串转为In表达式

string rui.dbTools.getInExpression(string value, bool handleEmpty = true)

### 数据展示格式

1. 将DataTable内某列的值转为显示格式

string rui.dbTools.getShowExpression(DataTable table, string columnName, char splitTag = ',')

1. 将List集合内的值转为显示格式

string rui.dbTools.getShowExpression(List<string> list, char splitTag = ',')

### Dapper参数辅助类

1. 返回Like参数的参数值

当sql语句中使用like时，like后边直接带参数名[ like @参数 ]，参数值需要用该方法返回。

string rui.dbTools.getDpLike(string value)

1. 返回In参数的参数值

当sql语句中使用in时，in后边直接带参数名[ in @参数 ]，参数值需要用该方法返回。

List<string> rui.dbTools.getDpList(string values)

List<string> rui.dbTools.getDpList(DataTable table, string columnName)

### 拼接搜索条件

1. = 拼接，下拉框用

string rui.dbTools.searchDdl(string fieldName, string fieldValue, DynamicParameters cmdPara = null)

and 字段名=参数值

1. Like拼接，文本框用

string rui.dbTools.searchTbx(string fieldName, string fieldValue, DynamicParameters cmdPara = null)

and 字段名 like ‘%参数值%’

1. 字段值包含值

string rui.dbTools.searchContainValue(string fieldName, string subValue, DynamicParameters cmdPara = null)

and charindex(参数值，字段名)>0

1. 值包含字段值

string rui.dbTools.searchContainField(string fieldName, string parentValue, DynamicParameters cmdPara = null)

and charindex(字段名, 参数值)>0

1. In查询

string rui.dbTools.searchIn(string fieldName, string inValue, DynamicParameters cmdPara = null)

and 字段名 in (参数值)>0

1. Not In查询

string rui.dbTools.searchNoIn(string fieldName, string inValue, DynamicParameters cmdPara = null)

and 字段名 not in (参数值)>0

1. 日期搜索

string rui.dbTools.searchDate(string fieldName, string dateStart, string dataEnd, DynamicParameters cmdPara = null)

and 字段名>=日期开始 and 字段名<=日期结束

1. 范围搜索

string rui.dbTools.searchRange(string fieldName, string startValue, string endValue, DynamicParameters cmdPara = null)

and 字段名>=开始值and 字段名<=结束值

### 集合操作

1. In表达式转List

List<string> rui.dbTools.getListByInExpr(string inExpr)

1. 带分隔符的字符串转List

List<string> rui.dbTools.getList(string value, char splitTag = ',')

1. 将Datatable中某列值转为List

List<string> rui.dbTools.getList(DataTable table, string columnName)

1. 向集合添加元素

List<string> rui.dbTools.addList<T>(List<T> list, T item)

1. 集合合并元素

List<string> rui.dbTools.unionList<T>(List<T> list1, List<T> list2)

1. 两个集合差运算

List<string> rui.dbTools.exceptList<T>(List<T> list1, List<T> list2)

1. 向Dictionary中添加元素

Dictionary<K, V> rui.dbTools.addDic<K, V>(Dictionary<K, V> dic, K key, V value)

### 其它方法

1. 检查Row中某字段的某个值是否存在

void rui.dbTools.checkRowFieldValue(DataRow[] rows, string fieldName, string fieldValue, string errorMsg)

1. 检查Row中某字段的某个值是否不存在

void rui.dbTools.checkRowFieldNotValue(DataRow[] rows, string fieldName, string fieldValue, string errorMsg)

1. 获取批量操作的错误提醒

string rui.dbTools.getBatchMsg(string opName, int rowCount, Dictionary<string, string> errorDic)

## rui.pagerBase - 主界面视图模型类基类

保存主界面所有数据模型类中所需要的相关属性

1. ResourceCode保存资源编码
2. keyField勾选行的标识字段
3. sumSql保存汇总查询语句 – 该语句会通过查询语句自动计算出来
4. sumRange保存汇总查询的数据范围（page、all）
5. sumRow保存汇总查询的结果
6. PageSize保存分页大小
7. RowCount保存汇总查询的数据条数
8. PageCount根据数据条数和分页大小计算出的页数
9. PageIndex保存当前分页位置
10. ExeCountSql保存是否需要执行汇总查询
11. ExportRange保存导出的数据范围（当页，所有数据）
12. SheetName保存导出的单元格名称
13. DataTable保存View中要展示的列表数据
14. showColumn保存View中要展示的列定义数据
15. LongDateField保存需要显示长日期格式的字段，多个用,分割。

保存主界面所有模型类中所需要的相关方法

1. Search数据搜索方法，页面的视图模型需要重写。
2. getPageConfig 获取页面的分页大小和页面宽度配置
3. getOrderSql获取排序子语句
4. getShowColumn获取配置的展示列
5. ExportToXls将查询结果导出到Excel文件中

## rui.pagerHelper - 分页获取数据辅助方法

根据查询语句获取分页数据。



1. 构造函数：利用查询语句和排序子语句，构造分页对象
2. processSql利用传入的Sql语句和排序子语句，构造汇总查询语句
3. Execute，根据构造的语句进行分页数据查询。
4. Result保存查询的分页结果

## rui.dataCheck - 数据检查

检查相关变量值是否满足要求，提供的相关方法如下：

1）检查是否是合法身份证

bool rui.dataCheck.check身份证(string value)

2）检查是否是合法手机号

bool rui.dataCheck.check手机号(string value)

3）插件布尔表达式是否为True，如果不是则抛出错误提示。

void rui.dataCheck.checkExpr(bool boolExpr, string errorMsg)

## rui.encryptHelper - 加密和解密

包含了MD5加密和对称加密方法的封装。

1. MD5加密：采用MD5对明文进行加密，加密后不可以被破解

string rui.encryptHelper.toMD5("明文")

1. DES加密：采用DES加密算法对明文加密

string rui.encryptHelper.DES加密("明文")

1. DES解密：采用DES加密算法对密文解密

string rui.encryptHelper.DES解密("密文")

## HTML辅助方法扩展

对MVC的HTML辅助方法进行扩充，添加一些方便使用的方法。

|  |  |
| --- | --- |
| 使用的方法 | 作用 |
| TextBoxDate(参数) | 创建日期选择的文本框 |
| TextBoxDateFor(参数) | 创建强类型的日期选择的文本框 |
| DisplayForDate(参数) | 展示日期的辅助方法，可以设定是否长格式日期 |
| DropDownListMulFor(参数) | 创建强类型的多选下拉框 |
| DropDownListMul(参数) | 创建多选下拉框 |
| RadioButtonListFor(参数) | 创建强类型的单选列表 |
| RadioButtonList(参数) | 创建单选列表 |
| CheckBoxListFor(参数) | 创建强类型复选列表 |
| CheckBoxList(参数) | 创建复选列表 |
| orderColumn(参数) | 创建排序列 |

日期选择文本框：要实现日期选择的效果，页面中引入my97DataPicker的相关资源后，给文本框加上onfocus=" WdatePicker()"

这些方法在View界面中使用。