

# SmartASP 0.2

## API Reference (2009.11.29)

### SmartASP.Core

#### \$ — Request

##### **\$(name[, subset])**

###### 函数说明

获取请求参数值。

###### 参数说明

name

**{String}** 参数名。

method

**{String}** 搜索的参数集合，省略时为不限制。

"get": QueryString 集合；

"post": Form 集合；

"serv": ServerVariables 集合。

###### 返回值

**{String}** 参数值。

##### **\$.getIp()**

###### 函数说明

获取客户端 IP。

###### 返回值

**{String}** IP。

## **\$.version — SmartASP 版本号**

## **\$.config — 配置信息**

### **prefix**

**{String}** 应用程序前缀，初始值为 "SmartASP\_"。

### **isDebug**

**{Boolean}** 是否调试模式，初始值为 true。

## **String 类型扩展**

### **String.prototype.trim([str])**

#### **方法说明**

去掉当前字符串两端的某段字符串。

#### **参数说明**

str

**{String}** 要去掉的字符串，默认为空白。

#### **返回值**

**{String}** 修整后的字符串。

### **String.prototype.left(n)**

#### **方法说明**

从左边开始截取一定长度的子字符串。

#### **参数说明**

n

**{Number}** 长度

#### **返回值**

**{String}** 子字符串

### **String.prototype.right(n)**

#### **方法说明**

从右边开始截取一定长度的子字符串。

### 参数说明

n

{Number} 长度。

### 返回值

{String} 子字符串。

## String.prototype.toNumber()

### 方法说明

获取当前字符串的数字类型值（调用 Number）。

### 返回值

{Number} 当前字符串的数字类型值。

## String.prototype.toInt()

### 方法说明

获取当前字符串的整型值（调用 parseInt）。

### 返回值

{Number} 当前字符串的整型值。

## String.prototype.toFloat()

### 方法说明

获取当前字符串的浮点值（调用 parseFloat）。

### 返回值

{Number} 当前字符串的浮点值。

## String.prototype.toBool()

### 方法说明

获取当前字符串的布尔值。"0"、"false"、空字符串均转换为 false，其他值为 true。

### 返回值

{Boolean} 当前字符串的布尔值。

## String.prototype.toDate()

### 方法说明

获取当前字符串的日期值，合法格式为 yyyy-MM-dd HH:mm:ss。

### 返回值

**{Date}** 当前字符串的日期值。如果该字符串无法转换为日期或日期不合法，则返回 undefined。

## Date 类型扩展

### Date.prototype.format(formation)

#### 方法说明

返回指定格式的日期字符串。

#### 参数说明

formation

**{String}** 格式字符串。其中，yyyy 是 4 位年份，yy 是 2 位年份；MM 是 2 位月份，M 是 1 位月份；dd 是 2 位日期，d 是 1 位日期；HH 是 2 位 24 小时制小时数，H 是一位 24 小时制小时数；hh 是 2 位 12 小时制小时数，h 是 1 位 12 小时制小时数；mm 是 2 位分钟，m 是 1 位分钟；ss 是 2 位秒数，s 是 1 位秒数。

#### 返回值

**{String}** 指定格式的日期字符串。

## Error 类型扩展

### Error.prototype.addMsg(msg)

#### 方法说明

根据调试标志修改异常信息。

#### 参数说明

msg

**{String}** 附加的异常信息。

#### 返回值

**{Error}** 当前异常对象。

## **\$.convert — 类型转换**

### **toBool(value)**

#### **方法说明**

把值转换为布尔类型。null、undefined、0、"0"、"false"、空字符串均转换为 false，其他值为 true。

#### **参数说明**

value

**{Mixed}** 值。

#### **返回值**

**{Boolean}** 转换后的布尔值。

### **toDate(value)**

#### **方法说明**

把字符串转换为日期类型。

#### **参数说明**

value

**{String}** 字符串，格式为 yyyy-MM-dd HH:mm:ss。

#### **返回值**

**{Date}** 转换后的日期，如果该字符串无法转换为日期或日期不合法，则返回 undefined。

## **\$.json — JSON 序列化和反序列化**

### **stringify(value)**

#### **方法说明**

把值序列化为 JSON 字符串。

#### **参数说明**

value

**{Mixed}** 值。

#### **返回值**

**{String}** JSON 字符串。

## parse(str)

### 方法说明

把 JSON 字符串反序列化为值。

### 参数说明

str

`{String}` JSON 字符串。

### 返回值

`{Mixed}` 值。

## \$.cache — 缓存处理

### set(name, value[, isToSerialize])

#### 方法说明

设置缓存。

#### 参数说明

name

`{String}` 缓存名。

value

`{Mixed}` 缓存值。

isToSerialize

`{Boolean}` 是否存储为 JSON 字符串，默认为否；如果缓存的变量是引用类型，请设为 `true`。

### get(name[, isToDeserialize])

#### 方法说明

获取缓存值。

#### 参数说明

name

`{String}` 缓存名。

isToDeserialize

`{Boolean}` 是否把值反序列化，默认为否；如果缓存的是 JSON 字符串，可以设为 `true` 以获取对应的值。

### 返回值

`{Mixed}` 缓存值。

## **del([name])**

### 方法说明

清理缓存。

### 参数说明

name

`{String}` 缓存名，省略时为清理全部。

## **\$.cookie — Cookie 操作**

## **get(name)**

### 方法说明

获取 Cookie。

### 参数说明

name

`{String}` Cookie 名。

### 返回值

`{String,Object}` Cookie 值或 Cookie 字典。

## **set(name, value[, expires, domain, path, secure])**

### 方法说明

设置 Cookie。

### 参数说明

name

`{String}` Cookie 名。

value

`{Mixed}` Cookie 值。

expires

`{Number,Date,String}` 过期时间，值可以为：

数字型：单位为秒；

日期型；

字符串：由数字和单位组成的字符串，如"1y"，其中 y 是年，M 是月，d 是日，h 是小时，m 是月，s 是秒。

domain

{String} 域。

path

{String} 路径。

secure

{Boolean} 是否仅把 Cookie 发送给受保护的服务器(https)。

## **del([name])**

### **方法说明**

删除 Cookie。

### **参数说明**

name

{String} Cookie 名，省略时为删除全部。

## **\$.session — Session 操作**

### **set(name, value)**

#### **方法说明**

设置 Session。

#### **参数说明**

name

{String} Session 名。

value

{Mixed} Session 值。

### **get(name)**

#### **方法说明**

获取缓存值。

#### **参数说明**

name

{String} Session 名。



### 返回值

`{Mixed}` Session 值。

## **del([name])**

### 方法说明

清理 Session。

### 参数说明

name

`{String}` Session 名，省略时为清理全部。

## **SmartASP.Validate**

### **\$.validate** — 数据验证

## **isNumbers(value)**

### 方法说明

验证指定值是否用逗号隔开的一段非负整数

### 参数说明

value

`{Mixed}` 值。

### 返回值

`{Boolean}` 是否符合条件。

## **SmartASP.Encrypt**

### **\$.encrypt** — 数据加密

## **toMd5(s[, len])**

### 方法说明

MD5 加密。

### 参数说明

s

{String} 明文。

len

{Number} 长度，16 或 32，默认为 32。

### 返回值

{String} 十六进制形式的密文。

## toSha1(s)

### 方法说明

SHA1 加密。

### 参数说明

s

{String} 明文。

### 返回值

{String} 十六进制形式的密文。

# SmartASP.IO

## \$io.Upload — 上传类

### getTotalBytes()

#### 方法说明

获取上传数据总量，单位为 b。

#### 返回值

{Number} 数据总量。

### getFileCount()

#### 方法说明

获取上传文件数。

#### 返回值

{Number} 文件数。

## getFile(index)

### 方法说明

获取文件信息对象。

### 参数说明

index

`{Number|String}` 文件索引。

### 返回值

`{Object}` 文件信息对象。该对象包含的属性如下：

fieldName：表单字段名；

fileName：文件名（不包含扩展名）

extName：截取扩展名（不包含“.”）；

mineType：MIME 类型；

dataPos：数据开始位置；

dataSize：文件大小，单位 b。

## getAutoFileName()

### 方法说明

获取年月日时分秒和 5 位随机数组成的自动文件名。

### 返回值

`{String}` 文件名。

## saveToFile(index, path[, fileName, extName])

### 方法说明

保存数据为文件。

### 参数说明

index

`{Number|String}` 文件索引。

path

`{String}` 保存路径（绝对路径）

fileName

`{String}` 文件名，默认为原文件名。

extName

`{String}` 扩展名，默认为原扩展名。

返回值

{String} 完整路径。

## Close()

方法说明

关闭文件流。

# SmartASP.Data

## \$.data.DbHelper — 数据库操作辅助类

**DbHelper(server, user, pwd, dbName[, drv])** *+1 重载*

函数说明

构造函数。

参数说明

server

{String} 数据库服务器地址。

user

{String} 用户名。

pwd

{String} 密码。

dbName

{String} 数据库名。

drv

{String} 数据库驱动,, 默认为 SQL Server。

**DbHelper(path[, drv])** *+1 重载*

函数说明

构造函数。

参数说明

server

{String} 文件数据库地址。

drv

{String} 数据库驱动, 默认为 Access。

## **connect()**

### **方法说明**

连接数据库。

## **setTimeout(timeout)**

### **方法说明**

设置数据库操作超时时间。

### **参数说明**

timeout

**{Number}** 超时时间，单位秒。

## **getAccessTime()**

### **方法说明**

获取数据库访问次数。

### **返回值**

**{Number}** 数据库访问次数。

## **createParam(name, type, value[, size, direction])**

### **方法说明**

创建命令参数。

### **参数说明**

name

**{String}** 参数名。

type

**{String,Number}** 参数类型。

int : 3,

bigint : 20,

smallint : 2,

tinyint : 16,

decimal : 14,

double : 5,

char : 129,

nchar : 130,

varchar : 200,  
text : 201,  
nvarchar : 202,  
ntext : 203,  
datetime : 135

value

{Mixed} 参数值。

size

{Number} 参数长度限制。

direction

{Number} 参数方向，默认为传入。

input : 1,  
output : 2,  
inputoutput : 3,  
returnvalue : 4

返回值

{Object} 参数对象。

## **attachParams(params)    +1 重载**

方法说明

附加命令参数。

参数说明

params

{Array,Object} 参数

## **attachParams(p1[, p2, ... , pN])    +1 重载**

方法说明

附加命令参数。

参数说明

p1—pN

{Object} 参数

## **copyParams(params)**

### **方法说明**

复制参数。

### **参数说明**

params

`{Object,Array}` 要复制的参数。

### **返回值**

`{Array}` 原参数及其副本。

## **clearParams()**

### **方法说明**

清理所有命令参数。

## **executeReader(text[, params, type])**

### **方法说明**

执行命令并返回记录集。

### **参数说明**

text

`{String}` 命令文本。

params

`{Object,Array}` 命令参数。

type

`{Number}` 命令类型，1 为 sql 语句，4 为存储过程，默认为 sql 语句。

### **返回值**

`{RecordSet}` 数据集。

## **executeNonQuery(text[, params, type])**

### **方法说明**

执行命令，无返回结果。

### **参数说明**

（见 executeReader 方法）

## **executeScalar(text[, params, type])**

### **方法说明**

执行命令并返回结果集中第一行第一列的值。

### **参数说明**

（见 executeReader 方法）

### **返回值**

`{Mixed}` 结果集中第一行第一列的值。

## **executeJson(text[, params, type])**

### **方法说明**

执行命令并返回 Json 数据。

### **参数说明**

（见 executeReader 方法）

### **返回值**

`{Array}` JSON 对象数组。

## **beginTransaction()**

### **方法说明**

开始事务。

## **rollbackTrans()**

### **方法说明**

回滚事务。

## **commitTrans()**

### **方法说明**

提交事务。

## **rsToJson(rs)**

### **方法说明**

把记录集转换成 JSON 对象数组。



## 参数说明

rs

`{RecordSet}` 记录集。

## 返回值

`{Array}` JSON 对象数组。

## close()

### 方法说明

释放数据库连接。