84PHP 开源框架

基础模块说明

V1.2.1

目录

Sql

简述：本模块底层通过 MySQLi 驱动，进行 CDUR（增删改查）操作。

配置：Sql .php。

$RW_Splitting（Bool）用以指定读写分离是否开启，当值为 TRUE 时开启，值为 FALSE 时关闭。读写分离按照一主多从模式进行，其中 default 组为写入组，其余组为读取组，读取组选择的方式是随机选取。如果要使用读写分离功能，至少需要一个读取组和一个写入组，此外您还需要对这些数据库进行额外的主从同步配置；

$DbInfo 是一个二维数组（**Array**），用以指定数据库信息。在第一维数组中，键为配置组名称，值为一个数组，且该数组中，有且只能有以下四个元素：

1、键为 address，值为数据库地址，如果非远程数据库，请使用默认值 localhost；

2、键为 username，值为数据库用户名；

3、键为 password，值为数据库密码；

4、键为 dbname，值为数据库名称。

配置组“default”不可删除，**也不可将 default 改为其他名称**，可以**在第一维数组中增加更多的元素**以配置多数据库。

方法：

Choose()：

- **语法：*#[变量=]<Sql@Choose(ChooseDb)>***
- **说明：**在预配置的多个数据库中选择一个数据库。**本方法在读写分离功能启用时无效。**

- **参数：**

  Choosedb：String 类型，必须。用以指定所查询数据库的名称。

- **返回：无。**
- **示例：**

  假设配置文件包含如下两个数据库：

```
$DbInfo=array(
    'default'=>array(
        'address'=>'localhost',
        'username'=>'root',
        'password'=>'0000',
        'dbname'=>'demo'
    ),
    'db2'=>array(
        'address'=>'192.168.1.123',
        'username'=>'user1',
        'password'=>'pwd1',
        'dbname'=>'db2'
    )
);
```

  当需要使用“db2”作为后续操作所使用的数据库时：

```
#$Result=<Sql@Choose('db2')>
```

- *特别注意：*

  一旦使用本方法后，**在单次运行中，除非再次使用本方法，否则将一直保持**当前指定的数据库**连接。**

  在单次运行中从未使用本方法时，将以配置文件中的“default”配置组作为默认的数据库连接配置。

  如果要使用默认配置组连接数据库，请传入“default”作为参数 ChooseDb 的值。

Select()：

- **语法：#[变量=]<Sql@*Select(Table, WhereField, WhereValue, Field)*>**
- **说明：**查询一条数据表记录。
- **参数：**

  Table：String 类型，必须。用以指定所查询数据表的名称。

  WhereField：String 类型，必须。用以指定所查询的字段。

  WhereValue：String 类型，必须。用以指定所查询字段的值。

  Field：String 类型，可选。用以指定仅返回查询结果中的某个字段，而不是结果中的所有字段。当指定 Field 时，返回的数组只包含一个元素，元素中键为指定的字段名，值为此字段对应的记录值。
- **返回：String 类型或 Array。**

  查询到记录时，返回值为一个数组，记录的每一个字段为一个键，每一个记录值为一个数组的值，且数组中键值的对应关系与记录中字段和记录值的对应关系相同。当指定参数 Field 且记录中存在字段 Field 时，返回一个字符串，此字符串为记录中字段 Field 对应的记录值。当查询到的记录为空时，返回值为一个空数组。

  存在多条记录时，将返回所查询到的第一条记录。请不要用这个方法去实现譬如“查找某用户发布的最新文章”之类的目的。（*排序不可*

靠!）。

- **示例：**

  查询 user 表中，用户名（字段 username）为 Jack 的用户信息中，并存储在变量$Result 中：

  ```
  #$Result=<Sql@Select('user', 'username', 'Jack')>
  ```

  $Result 为：

  ```
  Array(
      ['id']=>'1',
      ['username']=>'Jack',
      ['password']=>'abc123',
      ['phone']=>'',
      ['email']=>'test@84php.com'
  )
  ```

  想取得 Jack 的邮箱地址时，可以直接取得前例中$Result 中的 email 元素，也可以：

  ```
  #$Result=<Sql@Select('user', 'username', 'Jack','email')>
  ```

  $Result 为：

  ```
  'test@84php.com'
  ```

- *特别注意：*

不指定 Field 时，得到的结果由于包含全部字段因而具有更强的灵活性，但是当表中字段较多（一般 50 个以上时），或其它字段的内容很大（如一篇文章的内容），处于性能考虑，请指定 Field。

务必注意对*来自于用户提交的数据进行安全过滤*(使用 Receive 模块中 SafeCheck()方法)，如果最终执行的语句出现额外的’等 SQL 语法

符号时，*框架将报错*。

SelectMore()：

- **语法：#[变量=]<Sql@*SelectMore(Table, WhereField, WhereValue, WhereOp, Order, Desc, Limit, Like, Index)*>**
- **说明：**查询多条数据表记录。
- **参数：**

  Table：String 类型，必须。用以指定所查询数据表的名称。

  WhereField：String 类型或 Array 类型，可选，默认值为 NULL（空）。用以指定查询条件中的字段，类型应当与 WhereValue 保持一致。当传入 WhereField 时，必须传入 WhereValue。当 WhereField 为 String 类型时，表示仅限定一个查询条件；当 WhereField 为 Array 类型时，**每一个元素代表一个限制条件，且元素与 WhereValue 以及 WhereOp（如果传入）中的元素保持相互对应。**

  WhereValue：String 类型或 Array 类型，可选，默认值为 NULL（空）。用以指定查询条件中字段对应的值，类型应当与 WhereField 保持一致。当 WhereValue 为 String 类型时，表示仅限定一个查询条件；当 WhereValue 为 Array 类型时，**每一个元素代表一个限制条件，且元素与 WhereField 以及 WhereOp（如果传入）中的元素保持相互对应。**

  WhereOp：String 类型或 Array 类型，可选，默认值为“=”。用以确定 WhereField 和 WhereValue 之间的比较符以及与下一个条件之间的并列关系，以及指定各个条件之间的优先级关系。WhereOp 为 String 类型时，表示仅限定一个查询条件；当 WhereOp 为 Array 类型时，键无需定义，值为“比较符[,并列关系]”的字符串，**其中比较符为此元素对应的 WhereField 元素和 WhereValue 元素之间的比较符（=，<，>，>=，<=，<>），并列关系为与下一个限制条件之间的并列关系（and 或 or，未被定义时默认值为 and），并且每一个元素与 WhereField 以及 WhereValue 中的元素保持相互对应。**

  自 **v1.2.1** 起，新增了多个限定条件中，确定条件（或条件组合）之间的优先级的功能。即在最终的 SQL 语句中，框架会根据传入的优先级参数的值，使用括号“()”对查询条件进行优先级的指定，例如“... where `id` = ‘1’ and (`level` = ‘a’ or `class` >

‘3’ ）...”的查询语句。如需使用此新特性，则 WhereOp 必须为数组，且 WhereOp 数组的每一个元素的值将由 String 类型改变为 Array 类型，且包含两个元素，键无需定义，第一个元素的值为“比较符[,并列关系]”的字符串，该字符串的定义方式与 WhereOp 为 String 类型时的定义方式相同，不再赘述；第二个元素的值为“前置括号[,后置括号]”的字符串。**其中前置括号为添加在该条件前的左括号，可以为 0 个或多个“(”；后置括号为添加在该条件后、下一个并列关系符（AND 或 OR）前的右括号，可以为 0 个或多个“)”。**当仅需要添加右括号时，字符串应当定义为“,后置括号”，即起始字符为一个英文逗号。

Order：String 类型，可选。用以指定对结果排序时所依据的字段。

Desc：Bool 类型，可选，默认为 FALSE。用以指定排序方式：TRUE 倒序排列，FALSE 正序排列。

Limit：Array 类型，可选。用以从位置偏移量为起始行取得指定行数的记录。当数组仅存在一个元素时，此元素定义行数，此时位置偏移量默认为 0；当数组存在两个元素时，第一个元素定义位置偏移量，第二个元素定义行数。

Like：Int 类型，可选。用以进行关键字搜索。本参数使用的前提是 WhereField 和 WhereValue 均为 String 类型，Like 为 1 时代表包含关键字，Like 为 2 时代表排除关键字。并可在 WhereValue 中使用%作为通配符。请注意，*本参数很有可能造成全表扫描*，如果数据量较大，您应当进行分库分表或使用搜索中间件。

***Index*：String 类型，可选。**用以指定执行操作时使用的索引。

- **返回：二维数组或不含任何元素的空数组。**

查询到记录时，返回值为二维数组，在第一维的数组中，键为从 0 开始的自增键，值为一行记录的集合的一维数组；在第二维数组中，键为字段名，值为该条记录中字段所对应的记录值。

当查询到的记录为空时，返回值为一个空数组。

- **示例：**

假设在 finance 表中查询充值（payin 字段）大于 3000 或提现（payout 字段）小于 5000 的记录，并存储于$Result 变量中，则应当：

```
#$Result=<Sql@Select('finance',array('payin','payout'),array('3000','5000'),array('>,or','<'))>
```

另，假设查询某个表的所有信息（WhereField 和 WhereValue 为空时），则返回数组为：

```
Array(
    [0]=>Array(
        ['id']=>'1',
        ['username']=>'Jack',
        ['phone']=>'888888'
    ),
    [1]=>Array(
        ['id']=>'2',
        ['username']=>'Lucy',
        ['phone']=>'123456'
    )
)
```

- *特别注意：*

  务必注意对*来自于用户提交的数据进行安全过滤*(使用 Receive 模块中 SafeCheck()方法)，如果最终执行的语句出现额外的’等 SQL 语法符号时，*框架将报错*。

Total()：

- **语法：#[变量=]<Sql@*Total(Table, WhereField, WhereValue, WhereOp, Like, Index)*>**

- **说明**：统计符合条件的记录数量。
- **参数**：

  Table：**String 类型，必须。**用以指定所查询数据表的名称。

  WhereField：**String 类型或 Array 类型，可选，默认值为 NULL（空）。**用以指定查询条件中的字段，类型应当与 WhereValue 保持一致。当传入 WhereField 时，必须传入 WhereValue。当 WhereField 为 String 类型时，表示仅限定一个查询条件；当 WhereField 为 Array 类型时，**每一个元素代表一个限制条件，且元素与 WhereValue 以及 WhereOp（如果传入）中的元素保持相互对应。**

  WhereValue：**String 类型或 Array 类型，可选，默认值为 NULL（空）。**用以指定查询条件中字段对应的值，类型应当与 WhereField 保持一致。当 WhereValue 为 String 类型时，表示仅限定一个查询条件；当 WhereValue 为 Array 类型时，**每一个元素代表一个限制条件，且元素与 WhereField 以及 WhereOp（如果传入）中的元素保持相互对应。**

  WhereOp：**String 类型或 Array 类型，可选，默认值为“=”。**用以确定 WhereField 和 WhereValue 之间的比较符以及与下一个条件之间的并列关系，以及指定各个条件之间的优先级关系。WhereOp 为 String 类型时，表示仅限定一个查询条件；当 WhereOp 为 Array 类型时，键无需定义，值为“比较符[,并列关系]”的字符串，**其中比较符为此元素对应的 WhereField 元素和 WhereValue 元素之间的比较符（=，<，>，>=，<=，<>），并列关系为与下一个限制条件之间的并列关系（and 或 or，未被定义时默认值为 and），并且每一个元素与 WhereField 以及 WhereValue 中的元素保持相互对应。**

  自 **v1.2.1** 起，新增了多个限定条件中，确定条件（或条件组合）之间的优先级的功能。即在最终的 SQL 语句中，框架会根据传入的优先级参数的值，使用括号“()”对查询条件进行优先级的指定，例如“... where `id` = '1' and (`level` = 'a' or `class` > '3') ...”的查询语句。如需使用此新特性，则 WhereOp 必须为数组，且 WhereOp 数组的每一个元素的值将由 String 类型改变为 Array 类型，且包含两个元素，键无需定义，第一个元素的值为“比较符[,并列关系]”的字符串，该字符串的定义方式与 WhereOp 为 String 类型时的定义方式相同，不再赘述；第二个元素的值为“前置括号[,后置括号]”的字符串。**其中前置括号为添加在该条件前的左括号，可以为 0 个或多个“(”；后置括号为添加在该条件后、下一个并列关系符（AND 或 OR）前的右括号，可以为 0 个或多个“)”。**当

仅需要添加右括号时，字符串应当定义为“,后置括号”，即起始字符为一个英文逗号。

Like：Int 类型，可选。用以进行关键字搜索。本参数使用的前提是 WhereField 和 WhereValue 均为 String 类型，Like 为 1 时代表包含关键字，Like 为 2 时代表排除关键字。并可在 WhereValue 中使用%作为通配符。请注意，*本参数很有可能造成全表扫描*，如果数据量较大，您应当进行分库分表或使用搜索中间件。

Index：String 类型，可选。用以指定执行操作时使用的索引。

- **返回：Int 类型。**
- *特别注意：*

  务必注意对*来自于用户提交的数据进行安全过滤*(使用 Receive 模块中 SafeCheck()方法)，如果最终执行的语句出现额外的’ 等 SQL 语法符号时，*框架将报错*。

Insert()：

- **语法：#[变量=]<Sql@*Insert(Table, Array)*>**
- **说明：**插入数据，并返回新记录的主键。
- **参数：**

  Table：String 类型，必须。用以指定所操作数据表的表名称。

  Array：Array 类型，必须。用以指定需要插入的数据，其中键为字段名，值为字段名所对应的值。
- **返回：Int 类型**，值为插入成功后该行记录的主键。
- *特别注意：*

  务必注意对*来自于用户提交的数据进行安全过滤*(使用 Receive 模块中 SafeCheck()方法)，如果最终执行的语句出现额外的’ 等 SQL 语法符号时，*框架将报错*。

Delete()：

- **语法：#[变量=]<Sql@*Delete(Table, WhereField, WhereValue, WhereOp, Limit, Index)*>**
- **说明：**删除符合条件的数据。
- **参数：**

  Table：String 类型，必须。用以指定所操作数据表的名称。

  WhereField：String 类型或 Array 类型，可选，默认值为 NULL（空）。用以指定查询条件中的字段，类型应当与 WhereValue 保持一致。当传入 WhereField 时，必须传入 WhereValue。当 WhereField 为 String 类型时，表示仅限定一个查询条件；当 WhereField 为 Array 类型时，**每一个元素代表一个限制条件，且元素与 WhereValue 以及 WhereOp（如果传入）中的元素保持相互对应。**

  WhereValue：String 类型或 Array 类型，可选，默认值为 NULL（空）。用以指定查询条件中字段对应的值，类型应当与 WhereField 保持一致。当 WhereValue 为 String 类型时，表示仅限定一个查询条件；当 WhereValue 为 Array 类型时，**每一个元素代表一个限制条件，且元素与 WhereField 以及 WhereOp（如果传入）中的元素保持相互对应。**

  WhereOp：String 类型或 Array 类型，可选，默认值为“=”。用以确定 WhereField 和 WhereValue 之间的比较符以及与下一个条件之间的并列关系，以及指定各个条件之间的优先级关系。WhereOp 为 String 类型时，表示仅限定一个查询条件；当 WhereOp 为 Array 类型时，键无需定义，值为“比较符[,并列关系]”的字符串，**其中比较符为此元素对应的 WhereField 元素和 WhereValue 元素之间的比较符（=，<，>，>=，<=，<>），并列关系为与下一个限制条件之间的并列关系（and 或 or，未被定义时默认值为 and），并且每一个元素与 WhereField 以及 WhereValue 中的元素保持相互对应。**

  自 **v1.2.1** 起，新增了多个限定条件中，确定条件（或条件组合）之间的优先级的功能。即在最终的 SQL 语句中，框架会根据传入的优先级参数的值，使用括号“()”对查询条件进行优先级的指定，例如“... where `id` = '1' and (`level` = 'a' or `class` > '3') ...”的查询语句。如需使用此新特性，则 WhereOp 必须为数组，且 WhereOp 数组的每一个元素的值将由 String 类型改变为

Array 类型，且包含两个元素，键无需定义，第一个元素的值为“比较符[,并列关系]”的字符串，该字符串的定义方式与 WhereOp 为 String 类型时的定义方式相同，不再赘述；第二个元素的值为“前置括号[,后置括号]”的字符串。**其中前置括号为添加在该条件前的左括号，可以为 0 个或多个“(”；后置括号为添加在该条件后、下一个并列关系符（AND 或 OR）前的右括号，可以为 0 个或多个“)”。**当仅需要添加右括号时，字符串应当定义为“,后置括号”，即起始字符为一个英文逗号。

Limit：Array 类型，可选。用以从位置偏移量为起始行取得指定行数的记录。当数组仅存在一个元素时，此元素定义行数，此时位置偏移量默认为 0；当数组存在两个元素时，第一个元素定义位置偏移量，第二个元素定义行数。

Index：String 类型，可选。用以指定执行操作时使用的索引。

- **返回：Bool 类型**，值为 TRUE。
- *特别注意：*

  务必注意对*来自于用户提交的数据进行安全过滤*(使用 Receive 模块中 SafeCheck()方法)，如果最终执行的语句出现额外的’ 等 SQL 语法符号时，*框架将报错*。

Update()：

- **语法：#[变量=]<Sql@*Update(Table, WhereField, WhereValue, Array, WhereOp, Limit, AutoOp, Index)*>**
- **说明：**更新符合条件的数据。
- **参数：**

  Table：String 类型，必须。用以指定所操作数据表的名称。

  WhereField：String 类型或 Array 类型，可选，默认值为 NULL（空）。用以指定查询条件中的字段，类型应当与 WhereValue 保持一致。当传入 WhereField 时，必须传入 WhereValue。当 WhereField 为 String 类型时，表示仅限定一个查询条件；当 WhereField 为 Array 类型时，**每一个元素代表一个限制条件，且元素与 WhereValue 以及 WhereOp（如果传入）中的元素保持相互对应。**

WhereValue：String 类型或 Array 类型，可选，默认值为 NULL（空）。用以指定查询条件中字段对应的值，类型应当与 WhereField 保持一致。当 WhereValue 为 String 类型时，表示仅限定一个查询条件；当 WhereValue 为 Array 类型时，**每一个元素代表一个限制条件，且元素与 WhereField 以及 WhereOp（如果传入）中的元素保持相互对应。**

Array：Array 类型，必须。用以指定需要插入的数据，其中**键为字段名，值为需要被更新的与键中字段名所对应的记录值。**

WhereOp：String 类型或 Array 类型，可选，默认值为“=”。用以确定 WhereField 和 WhereValue 之间的比较符以及与下一个条件之间的并列关系，以及指定各个条件之间的优先级关系。WhereOp 为 String 类型时，表示仅限定一个查询条件；当 WhereOp 为 Array 类型时，键无需定义，值为“比较符[,并列关系]”的字符串，**其中比较符为此元素对应的 WhereField 元素和 WhereValue 元素之间的比较符（=，<，>，>=，<=，<>），并列关系为与下一个限制条件之间的并列关系（and 或 or，未被定义时默认值为 and），并且每一个元素与 WhereField 以及 WhereValue 中的元素保持相互对应。**

自 **v1.2.1** 起，新增了多个限定条件中，确定条件（或条件组合）之间的优先级的功能。即在最终的 SQL 语句中，框架会根据传入的优先级参数的值，使用括号“()”对查询条件进行优先级的指定，例如“... where `id` = '1' and (`level` = 'a' or `class` > '3') ...”的查询语句。如需使用此新特性，则 WhereOp 必须为数组，且 WhereOp 数组的每一个元素的值将由 String 类型改变为 Array 类型，且包含两个元素，键无需定义，第一个元素的值为“比较符[,并列关系]”的字符串，该字符串的定义方式与 WhereOp 为 String 类型时的定义方式相同，不再赘述；第二个元素的值为“前置括号[,后置括号]”的字符串。**其中前置括号为添加在该条件前的左括号，可以为 0 个或多个“(”；后置括号为添加在该条件后、下一个并列关系符（AND 或 OR）前的右括号，可以为 0 个或多个“)”。**当仅需要添加右括号时，字符串应当定义为“,后置括号”，即起始字符为一个英文逗号。

Limit：Array 类型，可选。用以从位置偏移量为起始行取得指定行数的记录。当数组仅存在一个元素时，此元素定义行数，此时位置偏移量默认为 0；当数组存在两个元素时，第一个元素定义位置偏移量，第二个元素定义行数。

AutoOp：Array 类型，可选。用以指定自动更新的方式，例如“+1”则会使指定的字段值自动+1，详见 MySQL 官方手册。键无需定义，值为表达式，并且每一个元素与 Array 中的元素保持相互对应。

Index：String 类型，可选。用以指定执行操作时使用的索引。

- **返回：Bool 类型**，值为 TRUE。
- *特别注意：*

  务必注意对*来自于用户提交的数据进行安全过滤*(使用 Receive 模块中 SafeCheck()方法)，如果最终执行的语句出现额外的’ 等 SQL 语法符号时，*框架将报错*。

Other()：

- **语法：#[变量=]<Sql@*Other(SqlString, Fetch)*>**
- **说明：**执行自定义的 SQL 语句。
- **参数：**

  SqlString：String 类型，必须。用以指定所执行的 SQL 语句。

  Fetch：Bool 类型，可选，默认值为 FALSE。用以指定是否取回结果集。当值为 FALSE 时，将返回一个 Mysqli->query 对象；当值为 TRUE 时，将以 MYSQLI_ASSOC 方式取回数据集，返回一个数组。
- **返回：对象或二维数组或不含任何元素的空数组。**

  当参数 Fetch 值为 FALSE 时，将返回一个 Mysqli->query 对象。

  当参数 Fetch 值为 TRUE 且查询到记录时，返回值为二维数组，在第一维的数组中，键为从 0 开始的自增键，值为一行记录的集合的一维数组；在第二维数组中，键为字段名，值为该条记录中字段所对应的记录值。

  当参数 Fetch 值为 TRUE 且查询到的记录为空时，返回值为一个空数组。
- *特别注意：*

  务必注意对*来自于用户提交的数据进行安全过滤*(使用 Receive 模块中 SafeCheck()方法)。

Wrong

简述：输出预定义的报错信息。

配置：无。

方法：

Report()：

- **语法：Wrong::*Report(ErrorDetail, ShowErrorInfo, OnlyMessage)***
- **说明：输出报错，并中断代码运行。**
- **参数：**

  ErrorDetail：String 类型，必须。用以指定报错的内容。

  ShowErrorInfo：Bool 类型，可选，默认值为 FALSE。用以指定是否在调试模式关闭时，仍然输出报错信息（即传入的 ErrorDetail 的值）。当值为 FALSE 时，不输出详细报错信息；当值为 TRUE 时，输出详细报错信息。

  OnlyMessage：Bool 类型，可选，默认值为 FALSE。用以指定是否输出友好报错界面的 HTML。当值为 FALSE 时，输出报错文本和友好报错界面的 HTML；当值为 TRUE 时，仅输出报错文本，而不输出友好报错界面的 HTML。

- **返回：无。**

Dir

简述：获取目录或文件的属性、状态；复制、删除目录或文件。

配置：无。

方法：

State()：

- **语法：#[变量=]<Dir@*State(PathArray)*>**
- **说明：**获取文件夹或文件的可读性、可写性、可执行性。
- **参数：**

  PathArray：Array 类型，必须。用以指定需要获取属性的文件夹或文件的路径。键无需定义，值为需要操作的文件夹或文件的相对于应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径，以“/”开始。
- **返回：Array 类型的二维数组。**

  在第一维数组中，键为文件夹或文件的路径，值为数组；在第二维数组中，有以下 3 个元素：

  1、键为 **Read**，值为 String 类型。当值为 Yes 时，代表该文件夹或文件可读，当值为 No 时，代表该文件夹或文件不可读；

  2、键为 **Write**，值为 String 类型。当值为 Yes 时，代表该文件夹或文件可写，当值为 No 时，代表该文件夹或文件不可写；

  3、键为 **Execute**，值为 String 类型。当值为 Yes 时，代表该文件可写，当值为 No 时，代表该文件不可写。文件夹没有此属性。

  当文件夹或文件不存在时，第二维数组将是一个空数组。
- **示例：**

  检测公共配置文件 Common.php 是否可读：

```
#$Result=<Dir@State(array('/Core/Common.php'))>
```

$Result 为：

```
Array(
    ['/Core/Common.php']=>Array(
        ['Read']=>'Yes',
        ['Write']=> 'Yes',
        ['Execute']=>'Yes'
    )
)
```

- *特别注意：*

  本方法不会遍历目录，请指定所有需要获取状态的文件夹或文件。

Size()：

- **语法：#[变量=]<Dir@*Size(Dir, Unit)*>**
- **说明：**获取指定目录所包含的所有文件的体积之和。
- **参数：**

  Dir：String 类型，必须。用以指定文件夹路径，值为需要操作的文件夹或文件的相对于应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径，以“/”开始。

  Unit：String 类型，可选，默认值为 NULL（空）。可传入 KB/MB/GB，用以指定返回体积的计算单位。当未传入本参数时，返回体积的计算单位为 B（字节）。

- **返回：String 类型。**

  根据计算单位（如果指定）计算得出的体积大小。

Delete()：

- **语法：#<Dir@*Delete(Dir)*>**
- **说明：**删除指定的目录。
- **参数：**

  Dir：String 类型，必须。用以指定文件夹路径，值为需要操作的文件夹或文件的相对于应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径，以“/”开始。

- **返回：无。**

Copy()：

- **语法：#<Dir@*Copy(From, To)*>**
- **说明：将一个指定的目录复制到指定的位置。**
- **参数：**

  From：String 类型，必须。用以指定需要复制的文件夹路径，值为需要操作的文件夹或文件的相对于应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径，以“/”开始。

  To：String 类型，必须。用以指定复制文件夹到何处，即新文件夹的父目录路径，值为需要操作的文件夹或文件的相对于应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径，以“/”开始。

- **返回：无。**

Send

简述：通过 HTTP 协议，以 GET 或 POST 方式发送数据。

配置：无。

方法：

Post()：

- **语法：#[变量=]<Send@*Post(Url, Data, Header, BuildQuery)*>**
- **说明：**通过 HTTP-POST 方式向目标地址发送数据。
- **参数：**

  Url：String 类型，必须。用以指定向何处发送数据。

  Data：Array 类型或 String 类型，必须。用以指定所发送的数据。当 Data 为 Array 类型时，键为字段名，值为字段对应的值；如果 Date 为 String 类型，例如需要发送 XML 数据，则需要定义 BuildQuery 为 FALSE。

  Header：String 类型，可选。用以指定需要发送的 Header 头。

  BuildQuery：Bool 类型，可选，默认值为 TRUE。用以指定是否对 Data 转换为 URI 参数。当传入 FALSE 时，Data 必须为 String 类型；当传入 TRUE 时，Data 必须为 Array 类型。
- **返回：String 类型或 Bool 类型。**

  当目标地址对请求响应时，会返回 String 类型的目标地址的响应。**当服务器网络不畅，或目标地址超时响应时，可能返回空字符串（String 类型）或 FALSE（Bool 类型）。**

Get()：

- **语法：#[变量=]<Send@*Get(Url, Data, Header)*>**
- **说明：**通过 HTTP-GET 方式向目标地址发送数据,并将目标地址的响应存储在指定的变量**$变量名**中。
- **参数：**

  Url：String 类型，必须。用以指定向何处发送数据。

  Data：Array 类型，必须。用以指定所发送的数据。键为字段名，值为字段对应的值。

  Header：String 类型，可选。用以指定需要发送的 Header 头。
- **返回：String 类型或 Bool 类型。**

  当目标地址对请求响应时，会返回 String 类型的目标地址的响应。**当服务器网络不畅，或目标地址超时响应时，可能返回空字符串（String 类型）或 FALSE（Bool 类型）。**

Receive

简述：接收数据并对数据的可靠性进行验证和对数据进行安全处理。

配置：Receive .php。

$DangerChar 是一个数组，元素为需要被剔除的字符串；

$BeforeDomainCheck 用以设置是否对数据来源域名（例如表单提交页的域名）进行检测。部分运行环境（如 Apache）下的设置，会影响到 parse_url($_SERVER)['HTTP_REFERER']['host']，请查看服务器设置是否异常，或关闭本项以跳过对数据来源域名进行检测；

$**TokenExpTime** 用以指定 Token 的失效时间，单位秒。

方法：

FromCheck()：

- **语法：#[变量=]<Receive@*FromCheck(TokenCheck, UnsetToken)*>**
- **说明：**对数据的来源进行检查。

  方法会依次检查 HTTP Referer 信息、Token 信息，当数据来源可疑时，将报错。
- **参数：**

  TokenCheck：Bool 类型，可选，默认值为 FALSE。用以指定是否检查 Token。

  UnsetToken：Bool 类型，可选，默认值为 TRUE。用以指定是否在每一次检查后是否清空 Token，TRUE 为清空，FALSE 为不清空。
- **返回：无。**
- *特别注意：*

  需要检测 Token 时，需在此之前已经通过 Session 中的 Token()方法生成了 Token。

SafeCheck()：

- **语法：#[变量=]<Receive@*SafeCheck(WillCheck)*>**
- **说明：**对字符串进行安全过滤以防止 XSS 和 SQL 注入。
- **参数：**

  WillCheck：**String 类型，必须。**用以指定需要过滤的字符串。
- **返回：String 类型 。**

  如果需要在网页中显示过滤之前的内容，请使用 Tool 方法中的 Html()方法将内容转换。

Post()：

- **语法：#[变量=]<Receive@*Post(TokenCheck, FromCheck, SafeCheck, FieldCheck)*>**
- **说明：**对通过 POST 方式接收的数据的进行可靠性验证和对数据进行安全处理。
- **参数：**

  TokenCheck：**Bool 类型，可选，默认值为 FALSE。**用以指定是否检查 Token。

  FromCheck：**Bool 类型，可选，默认值为 TRUE。**用以指定是否对数据的来源进行检查，TRUE 为进行，FALSE 为不进行。

  SafeCheck：**Bool 类型，可选，默认值为 TRUE。**用以指定是否对数据进行安全过滤，TRUE 为进行，FALSE 为不进行。

  FieldCheck：**Array 类型，可选。**用以指定需要检测是否存在、值是否为空的字段。键无需定义，值为“字段名[,非空检查]”的字符串，其中字段名为需要检查的字段，非空检查为检查字段对应的值是否为空（值为 TRUE 或 FALSE，未被定义时默认值为 FALSE）。
- **返回：Array 类型的数组。**

  其本质就是经过处理的$_POST 变量，因此可以将其代替$_POST 使用。

- *特别注意：*

  需要检测 Token 时，需在此之前已经通过 Session 中的 Token()方法生成了 Token。

Get()：

- **语法：#[变量=]<Receive@*Get(TokenCheck, FromCheck, SafeCheck, FieldCheck)*>**
- **说明：**对通过 GET 方式接收的数据的进行可靠性验证和对数据进行安全处理。
- **参数：**

  TokenCheck：Bool 类型，可选，默认值为 FALSE。用以指定是否检查 Token，TRUE 为检查，FALSE 为不检查。

  FromCheck：Bool 类型，可选，默认值为 TRUE。用以指定是否对数据的来源进行检查，TRUE 为进行，FALSE 为不进行。

  SafeCheck：Bool 类型，可选，默认值为 TRUE。用以指定是否对数据进行安全过滤，TRUE 为进行，FALSE 为不进行。

  FieldCheck：Array 类型，可选。用以指定需要检测是否存在、值是否为空的字段。键无需定义，值为“字段名[,非空检查]”的字符串，其中字段名为需要检查的字段，非空检查为检查字段对应的值是否为空（值为 TRUE 或 FALSE，未被定义时默认值为 FALSE）。
- **返回：Array 类型的数组。**

  其本质就是经过处理的$_GET 变量，因此可以将其代替$_GET 使用。
- **特别注意：**

  需要检测 Token 时，需在此之前已经通过 Session 中的 Token()方法生成了 Token。

Cookie()：

- **语法：#[变量=]<Receive@*Cookie(FieldCheck)*>**
- **说明：**对浏览器存储的 Cookie 数据进行安全处理。

- **参数：**

  FieldCheck：Array 类型，可选。用以指定需要检测是否存在、值是否为空的字段。键无需定义，值为“字段名[,非空检查]”的字符串，其中字段名为需要检查的字段，非空检查为检查字段对应的值是否为空（值为 TRUE 或 FALSE，未被定义时默认值为 FALSE）。

- **返回：Array 类型的数组。**

  其本质就是经过处理的$_COOKIE 变量，因此可以将其代替$_COOKIE 使用。

Json()：

- **语法：#[变量=]<Receive@*Json(JsonString, FieldCheck)*>**
- **说明：**对一个由一维数组转换而成的 Json 字符串进行安全处理。
- **参数：**

  JsonString：String 类型，必须。用以指定需要操作的 Json 字符串。JsonString 应当是由一维数组转换而成的 Json 字符串。

  FieldCheck：Array 类型，可选。用以指定需要检测是否存在、值是否为空的字段。键无需定义，值为“字段名[,非空检查]”的字符串，其中字段名为需要检查的字段，非空检查为检查字段对应的值是否为空（值为 TRUE 或 FALSE，未被定义时默认值为 FALSE）。

- **返回：Array 类型的数组或空（NULL）。**

  其本质就是经过处理的 json_decode()的返回值，因此可以将其代替 json_decode()使用。

  当返回值为空（NULL）

- *特别注意：*

  如果 JsonString 是由一个多维数组转换而来，那么第二维及第二维以上的数组不会被处理。

Load

简述：将浏览器上传的文件存储至服务器以及将远程的文件下载至服务器。

配置：无。

方法：

Up()：

- **语法：#[变量=]<Load@*Up(Name, Dir, SaveFileName, Size, Type)*>**
- **说明：**接收并处理浏览器 POST 上传的文件。
- **参数：**

  Name：String 类型，必须。用以指定表单中的文件域名称。

  Dir：String 类型，可选，默认值为“/Upload”。用以指定将上传的文件存储到何处，相对于/Web 目录，**以“/”开始。**

  SaveFileName：String 类型，可选，默认值为 NULL（空）。用以指定新文件的文件名前缀，当未定义或值为空（NULL）时，将采用时间戳+3 位随机数作为文件名的前缀。

  Size：Int 类型，可选，默认值为 10240。用以指定所上传文件的最大大小，单位 KB。

  Type：Array 类型，可选。用以指定所允许的文件上传类型。键无需定义，值为允许上传的文件后缀名。
- **返回：String 类型或 Bool 类型。**

  当返回值为 String 类型时，表示上传成功，返回的字符串即为文件相对应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径。

  当返回值为 Bool 类型 FALSE 时，说明上传失败，请在公共配置文件 Common.php 中开启调试模式后，复现场景查看详细的错误信息。

- *特别注意：*

  务必注意在服务器中设置资源的响应方式(例如访问一个 PDF 文档时，服务器是直接展示还是提供下载)，*这关系到服务器的安全*。

  务必在前端页面中，即表单页通过 HTML 代码和 JS 来限制文件的上传类型和大小，这样用户体验更加友好，也可以拦截大部分使用者“无心的非法操作”。

Down()：

- **语法：#[变量=]<Load@*Down(Url, Path, TimeLimit)*>**
- **说明：**从远程下载文件至服务器。
- **参数：**

  Url：String 类型，必须。用以指定需要下载的 URL。

  Path：String 类型，必须。用以指定将下载的文件存储到何处，值为相对于应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径，**以“/”开始。**

  TimeLimit：Int 类型，可选，默认值为 86400。用以指定超时时间，单位秒（s）。
- **返回：String 类型。**

  返回的字符串为文件相对应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径。
- *特别注意：*

  当服务器网络不畅，或目标地址超时响应时，框架将报错。

Session

简述：对 SESSION 进行操作。

配置：无。

方法：无。

特别注意：

自 1.2.1 版本起，原 Session 模块下的 Token 方法，现已转移至 Tool 模块下，在今后版本的文档中不再赘述。

Img

简述：对图片进行处理。

配置：Img.php。

$FontFile 用于指定字体文件的路径。

方法：

Base()：

- **语法：#[变量=]<Img@*Base(From, To, Width, Height, Scale, Word, WordColor)*>**
- **说明：**对图片进行缩放或添加水印。
- **参数：**

From：String 类型，必须。用以指定需要处理的图片的路径，值为相对于应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径，**以“/”开始。**

To：String 类型，必须。用以指定将新的图片存储到何处，值为相对于应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径，**以“/”开始。**

Width：Int 类型，可选，默认值为 NULL（空）。用以指定需要生成的图片的宽度，单位像素。当未指定 Width 而指定了 Height 时，生成的图片将是高为 Height 像素的等比缩放图片。

Height：Int 类型，可选，默认值为 NULL（空）。用以指定需要生成的图片的高度，单位像素。当未指定 Height 而指定了 Width 时，生成的图片将是宽为 Width 像素的等比缩放图片。

Scale：Float 类型，可选，默认值为 1.0。用以指定图片的缩放比例，仅在 Width 和 Height 都为空时有效。

Word：**String 类型，可选。**用以指定文字水印内容。当未定义或传入的值为空时，将不会为图片添加水印。

WordColor：**String 类型，可选，默认值为“#F5F5F5”。**用以指定文字水印颜色，值为形如#000000 的 16 进制的 RGB 色值。

- **返回**：**String 类型。**

  返回的字符串为文件相对应用根目录（即 Core、Source 文件夹所在的目录，通常是站点根目录）的路径。

- *特别注意：*

  务必开启 PHP 的 GD 库。

Tool

简述：一些实用的小功能。

配置：Tool.php。

$HtmlTag 用于指定哪些 HTML 标记可以被还原。

$HtmlMediaTag 用于指定哪些 HTML 中的多媒体标记可以被还原。

$HtmlMediaEndTag 用于指定哪些 HTML 中的多媒体标记中的结束标记可以被还原。

方法：

Random()：

- **语法：#[变量=]<Tool@Random*(Mode, StringLength)*>**
- **说明：**生成一个随机字符串。
- **参数：**

  Mode：String 类型，可选，默认为“AaN”。用以指定所生成字符串的字符构成方式。

  1、Mode 中含有“A”时，生成的字符串将含有大写的 A-Z；

  2、Mode 中含有“a”时，生成的字符串将含有小写的 a-z；

  3、Mode 中含有“N”时，生成的字符串将含有数字 0-9。

  StringLength：Int 类型，可选，默认为 32。用以指定随机字符串的长度。
- **返回：String 类型。**
- *特别注意：*

虽然字符串为随机生成，但仍有极小概率重复，因此不可直接使用本方法生成 UUID，您可以加上时间戳使其成为唯一的字符串。

Token()：

- **语法：#[变量=]<Tool@ *Token()*>**
- **说明：**种植 Token，Token 将存储于 Session 中。
- **参数：无。**
- **返回：String 类型。**

  将返回一个 32 位长度的随机字符串，通常植入于表单页，在提交数据时加入字段 Token（大小写敏感），值为该字符串，并开启 Receive 模块中的 Token 验证功能，则 Receive 模块中会自动获取来自 HTTP-GET 或 HTTP-POST 方式提交的数据中的 Token 字段，并进行验证。
- *特别注意：*

  Token 可防止跨站请求伪造（CSRF）攻击，建议应用于所有的表单页。

Html()：

- **语法：#[变量=]<Tool@Html*(Str, Tag_media, Tag_a, Tag_div)*>**
- **说明：**还原经过 Receive 模块中 SafeCheck()方法转换的字符串。
- **参数：**

  ***Str*：String 类型，必须。**用以指定待还原的字符串。

  ***Tag_media*：Bool 类型，可选，默认为 TRUE。**用以指定是否还原多媒体标记，TRUE 为还原，FALSE 为不还原。

  ***Tag_a*：Bool 类型，可选，默认为 TRUE。**用以指定是否还原 a 标记，TRUE 为还原，FALSE 为不还原。

Tag_div：**Bool 类型，可选，默认为 FALSE。**用以指定是否还原 div 标记，TRUE 为还原，FALSE 为不还原。

- **返回：String 类型。**