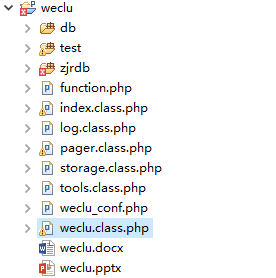
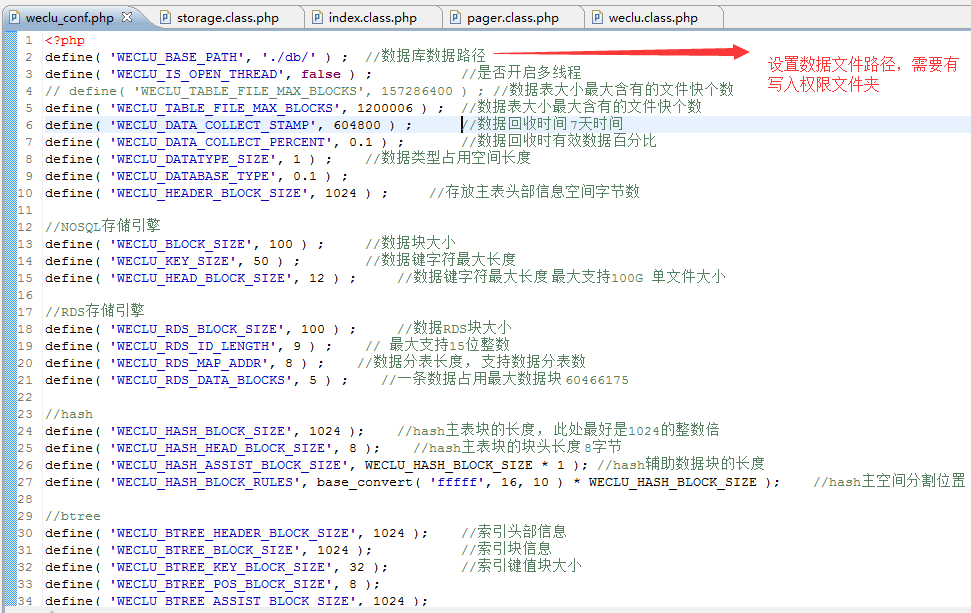
Weclu(维库)使用手册

## 开始

1. 引入 weclu.class.php



1. 设置数据文件路径



1. 初始化数据库

$weclu = new WECLU( $db );

* $db: 数据库名

1. 创建表

$weclu->create( $table, $engine ); //创建RDS存储引擎的表

* $table: 数据表
* $engine: 存储引擎，分别为RDS和HASH
* 返回值： bool, true 表示成功， false 表示失败

1. 获取所有的数据库

$weclu->show\_databases();

* 返回值： array( db1, db2, … );

1. 获取所有的数据表

* $weclu->show\_tables();
* 返回值： array( $table1, $table2, … );

1. 切换数据库

$weclu->use\_db( $db );

* $db: 数据库名
* 说明： 当该库已经存在时，直接切换， 不存在时，将先创建再切换

# 表操作

## Hash引擎表

1. Hash表操作查询

使用数据库中的第一张hash表时，首先需要初始化hash表空间

$weclu->init\_space();

1. $weclu->set( $table, $key, $val, $expire=0 );

* $table: 数据表;
* $key: 键
* $val: 值

$expire: 数据失效时间，<0 数据失效， =0 永久生效，>0 $expire秒后失效

* 返回值： $id

1. $weclu->sets( $table, $keyArr )；

* $table: 数据表;
* $keyArr: array( $key1 => $val1, $key2 => $val2, … ) 设置$key1, $key2，并且永久有效

或者 array( $key1 => array( $val1, 3600 ), $key2 => array( $val2 => array( $val2, 0 ) ) ) $key1 3600s(1小时)后失效，$key2 永久有效

* 返回值：array( $key1 => $id1, $key2 => $id2, … );

1. $weclu->get( $table, $key );

* $table: 数据表;
* $key: 键
* 返回值： $val

1. $weclu->gets( $table, $keys );

* $table: 数据表;
* $keys: 键数组 array( $key1, $key2, … )
* 返回值： array( $key1 => $val1, $key2 => $val2, … )

## RDS引擎表：查询条件中至少包含一个索引，否则无法查询数据，\_id 是一直存在的主键索引，由系统自动生成，无需手动添加，可作为查询条件

1. 创建需要查询的索引

$weclu->create\_index( $table, $indexs );

* $table: 数据表
* $indexs: 要创建的索引数组 array( $index1, $index2, $index3, … );

1. 删除索引

$weclu->create\_index( $table, $indexs );

* $table： 数据表
* $indexs: 索引数组，array( $index1, $index2, … );
* 返回值： 创建成功的索引， 数组类型， array( $index1, $index2, … )

1. 查询数据操作

$weclu->select( $table, $filed, $where, $sort=array(), $page=0, $prepage=1000 );

* $table: 数据表
* $filed: 字段，数组类型，或字符串类型，例： ‘name, age, birth’, array( ‘name’, ‘age’, ‘birth’ )

读取多维度，需要一维一维指定,中间用“.”隔开，例： ‘country.province.city’, 这样就能读取city维度下面的所有数据

* $where: 查询条件，数组类型，遵循 同一数组中并且关系，相邻数组数据或者关系，看例子
  + - * 1. array( ‘name’ => ‘zjr’, ‘age’ => 200 ) 查询名为zjr 并且年龄为200
        2. array( ‘name’ => ‘zjr’, ‘age’ => array( ‘>=’ => 18, ‘<=’ => 30 ) ) 查询名为zjr, 年龄在18岁到30岁之间 的数据行
        3. array( array( ‘name’ => ‘zjr’ ), array( ‘age’ => 200 ) ) 查询名为zjr或者年龄为200 的数据行
        4. array( ‘name’ => array( ‘zjr’, ‘zs’, ‘ls’, ‘ww’ ), ‘age’ => array( array( ‘<20’ ), array( ‘>’ => 60 ) ) ) 查询年龄小于20 或大于60 并且名字在’zjr, zs, ls, ww’中的任意一个 的数据行
        5. array( ‘name’ => array( ‘like’ => ‘zj%’ ) ) 查询名字为zj开头的所有数据行， 模糊查询， 只支持后模糊， 不支持前模糊查询。
        6. array( ‘name’ => array( ‘==’ => ‘zjr’ ) ) 等价于 array( ‘name’ => ‘zjr’ )
        7. array( ‘name’ => array( ‘in’ => array( ‘zjr’, ‘zs’ ) ) ) 等价于 array( ‘name’ => array( ‘zjr’, ‘zs’ ) )
        8. array( ‘country.province.city’ => ‘shenzhen’ ) 查询城市为深圳的数据行
        9. array( ‘\_id’ => 123 ) 查询数据主键Id为123的数据行
      * $sort: 为空时默认以主键id升序排序，不为空时 例

array( ‘\_id’ => ‘desc’ ) 以主键倒序

array( ‘name’ => ‘asc’, ‘age’ => ‘desc’ ) 先通过name升序，再通过age倒序

* + - * $page: 默认为0 表示所有， $page不为0 时表示指定的页
      * $prepage: 默认为1000, 指定时，返回指定的最大数据行数
      * 返回值：数组类型， 以filed为字段的数据

1. 插入操作

$weclu->insert( $table, $data );

* + - * $table: 数据表
      * $data: 数组， 支持无限多维
      * 返回值： 插入成功的ID

1. 修改操作

$welcu->update( $table, $data, $where );

* + - $table: 数据表
    - $data: 修改的字段数据，修改多维需指定维度，例 array( ‘country.province’ => array( ‘shenzhen’, ‘nanchang’ ) )
    - $where: 参考上面的where条件
    - 返回值：修改成功的IDs, 数组类型

1. 删除操作

$weclu->delete( $table, $where );

* + - $table: 数据表
    - $where: 参考上面的where条件
    - 返回值： 删除成功的IDs，数组类型