

 <b>marketful</b> <small>fulfillment</small>	Documentación del archivo: cclass-dbcore.php
	Autor: Mauricio Alcala
	Fecha: 13 de agosto de 2018
	Versión: 1.01
	Autoriza: Adolfo Yanes

## Descripción general:

Son diferentes métodos que sirven para hacer queries y obtener resultados para las consultas.

## Funciones PHP

```

/*****
 * @Script
 * Se valida que exista la clase MKF_DBCore, en caso de que no exista
 * se ejecuta lo siguiente.
 */
if ( !class_exists( 'MKF_DBCore', false ) ):
/*****

/*****
 * @clase MKF_DBCore
 *
 * @atributos
 ** @var Plg_Instance
 ** @var Wpdb
 ** @var db_prefix
 ** @var plugin_prefix
 ** @var default_charset
 *
 */
class MKF_DBCore
{
    protected static $Plg_Instance = NULL;
    protected $Wpdb;
    private $db_prefix = "";
    private $plugin_prefix = "";
    private $default_charset = "utf8";
    private $default_collate = "utf8_general_ci";
/*****

/*****
 * @función __construct()
 *
 * Se valida que exista una variable con nombre PLUGIN_GNAME, en tal caso se le
 * otorga el valor de mkf_
 */
function __construct()
{
    $this->plugin_prefix = defined( 'PLUGIN_GSNAME' ) ? PLUGIN_GSNAME : 'mkf_';
}
/*****

/*****
 * @función getInstance()
 * Es una función estática que accede al objeto $instance y
 * busca que sea nulo, en cuyo caso crea el objeto.
 * En caso contrario retorna el objeto.
 */
public static function GetInstance()
{

```

```

        if ( is_null( self::$Plg_Instance ) )
        {
            self::$Plg_Instance = new self;
        }
        return self::$Plg_Instance;
    }
}
/*****

/*****
* @Función getWpdb()
*
* Se crea una variable global $wpdb y se retorna
*/
private function getWpdb()
{
    global $wpdb;
    return $wpdb;
}
/*****

/*****
* @función getUsersTableName()
*
* Se retorna el valor que retorna el valor que retorna
* el @método getPrefix() concatenado con la string 'users'
*/
public function getUsersTableName()
{
    return $this->getPrefix() . "users";
}
/*****

/*****
* @función getUserMetaTableName()
* Retorna el valor del @método getPrefix() concatenado
* con 'users'
*/
public function getUserMetaTableName()
{
    return $this->getPrefix() . "usermeta";
}
/*****

/*****
* @función getPostMetaTableName()
* Retorna el valor del @método getPrefix() concatenado
* con 'postmeta'
*/
public function getPostMetaTableName()
{
    return $this->getPrefix() . "postmeta";
}
/*****

/*****
* @función getTermTaxonomyTableName()
* Retorna el valor del @método getPrefix() concatenado
* con 'term_taxonomy'
*/
public function getTermTaxonomyTableName()
{
    return $this->getPrefix() . "term_taxonomy";
}
}

```

```

/*****

/*****
 * @función getTermRelationshipsTableName()
 * Retorna el valor del @método getPrefix() concatenado
 * con 'term_relationships'
 */
public function getTermRelationshipsTableName()
{
    return $this->getPrefix() . "term_relationships";
}
/*****

/*****
 * @función getTermsTableName()
 * Retorna el valor del @método getPrefix() concatenado
 * con 'terms'
 */
public function getTermsTableName()
{
    return $this->getPrefix() . "terms";
}
/*****

/*****
 * @función getPlgWpDB()
 * Retorna el valor del @método loadWpdb()
 */
public function getPlgWpDB()
{
    return $this->loadWpdb();
}
/*****

/*****
 * @función getPrefix()
 * Valida si el atributo db_prefix esta vacio.
 * En caso de que si, le asigna el valor prefix del
 * @método getPlgWpDB().
 * Despues retorna el valor del atributo db_prefix
 *
 */
public function getPrefix()
{
    if (empty($this->db_prefix))
    {
        $this->db_prefix = $this->getPlgWpDB()->prefix;
    }
    return $this->db_prefix;
}
/*****

/*****
 * @función getPlgPrefix()
 * Retorna el valor del atributo plugin_prefix.
 */
public function getPlgPrefix()
{
    return $this->plugin_prefix;
}
/*****

```

```

/*****
 * @Función getCharSet()
 *
 * Se define una variable local llamada $charset_collate y
 * se le asigna el valor de ''
 * Se valida si el atributo de la función esta vacio
 * En caso de que si, la variable $charset_collate toma el
 * valor del atributo default_charset
 * En caso contrario, se le asigna el valor charset del
 * @método getPlgWpDB()
 *
 * Despues se crea una validación que valida si el atri-
 * buto collate del @método getPlgWpDB() esta vacio.
 * En caso de que así sea a la variable $charset_collate
 * se le asigna el valor default.
 * En caso contrario se le asigna el valor del atributo del
 * @método getPlgWpDB, collate.
 *
 * Al finalizar se retorna el valor de $charset_collate
 */
public function getCharSet()
{
    $charset_collate = '';

    if (empty($this->getPlgWpDB()->charset))
    {
        $charset_collate = "DEFAULT CHARSET={$this->default_charset}";
    }
    else
    {
        $charset_collate = "DEFAULT CHARACTER SET {$this->getPlgWpDB()->charset}";
    }

    if (empty($this->getPlgWpDB()->collate))
    {
        $charset_collate .= " COLLATE {$this->default_collate}";
    }
    else
    {
        $charset_collate .= " COLLATE {$this->getPlgWpDB()->collate}";
    }

    return $charset_collate;
}
/*****

/*****
 * @función execute_custom_query(@string, @string = (false))
 *
 * Dentro de la variable $result_set se guardan el resultado
 * de ejecutar la Query.
 * Despues validamos que $force_row sea verdadera, en cuyo caso
 * retornamos el valor de result_set
 * En caso contrario retornamos la variable $result_set
 */
public function execute_custom_query($query, $force_row = false)
{
    $result_set = $this->loadWpdb()->get_results($query, OBJECT);

    if ($force_row)
    {
        return $this->get_rows_force($result_set);
    }
    return $result_set;
}
/*****

/*****
 * @función get_rows_force(@Result_Set)
 *

```

```
* Creamos un switch con el numero de resultados que tiene el
* result_set.
* En el primer caso (1) retorna el $result_set
* En el segundo caso (0) retorna falso.
* En ninguno de los casos anteriores retorna el $result_set
*/
protected function get_rows_force($result_set)
{
    switch (count($result_set))
    {
        case 1: return $result_set[0];
        case 0: return false;
        default: return $result_set;
    }
    return $result_set;
}
/*****/
```