storage

Description

Core\Storage **storage**()

Our storage function allows you to store information in the DB, without the need to create custom database tables on a clients set. Its based on a key/value system that allows for custom queries and ordering plus the ability to set and return multiple values for a specific key.

> **Notice**
>
> The storage system is not a cache. All data sets are permanent, unless deleted. Use the cache function if you wish to simply cache data.

Examples

```php
<?php

route('/foo', function() {

   /**
    * Set a string value "bar" in the DB using the key "foo"
    */
   storage()->set('foo', 'bar');

   // Print the value for "foo"
   $object = storage()->get('foo');

   echo $object->value;



   /**
    * Storing arrays
    */
   storage()->set('foo_array', [
      'hello' => 'world'
   ]);

   // Print the value for "foo_array"
   $object = storage()->get('foo_array');

   // Prints the object
   print_r($object->value);

   // Print the "hello" value in our array, which will output "world"
   echo $object->value->hello;



   /**
    * Storing multiple values
    */
   storage()->set('multiple', '1');
   storage()->set('multiple', '2');
   storage()->set('multiple', '3');

   // Get all objects based on key
   $objects = storage()->all('multiple');
   foreach ($objects as $object) {
      // Will print 1,2,3
      echo $object->value . ',';
   }
});
```

Methods

Method	Description	Usage	Returns

get(string $key [, int $id = null])
 	
$key
 	
Unique storage key.
 	
$id
 	
If you store multiple data in a key
you can pass a unique ID to return a
specific item from that data set.
	Get data from the DB based on a unique key and/or ID.	`$object = storage()->get ('foo');`
		
		`echo $object->value;`
			On success it returns Core\Storage\Object, null on failure.

getById(int $id)
 	
$id
 	
Unique auto incremented ID.
	Get a specific data set based on the unique auto incremented key created by Core\Storage\Object.	`$object = storage()->getById (1);`
		
		`echo $object->value;`
			On success it returns Core\Storage\Object, null on failure.

all(string $key)
 	
$key
 	
Unique storage key.
	Returns all data sets based on a unique key.	`$objects = storage()->all ('multiple');`
		
		`foreach ($objects as $object) {`
		`  echo $object->value;`
		`}`
			Array of values on success, empty array on failure.
			
			Note: Each value from the array is an object from Core\Storage\Object

set(string $key, mixed $value , int $id = 0])
 	
$key
 	
Unique storage key.
 	
$value
 	
Value to set for this entry. Can be a string, array, int or boolean.
 	
$id
 	
(optional) Include a custom unique ID for this data set.
	Sets data into the DB based on your key.
	
	Notice
	
	This method will set multiple entries even if the key already exists.
	If you only want one entry for a key make sure to check if it exists and use the update method.
		`// Set a string`
		`storage()->set ('foo', 'bar');`
		
		`// Set an array`
		`storage()->set ('foo_array', [`
		`  'hello' => 'world'`
		`]);`
		
		`// Set multiple values`
		`storage()->set ('foo', '1');`
		`storage()->set ('foo', '2');`
		`storage()->set ('foo', '3');`
			Returns unique auto increment **ID** on success, **false** on failure.

update(string $key, mixed $value) $key Unique storage key. $value Value to set for this entry. Can be a string, array, int or boolean.	Updates the value of a data set that already exists. If the value is an array, it will merge any new data and overwrite any existing data in the value.	`storage()->update ('foo', 'bar');`	On success it returns the updated Core\Storage\Object, **false** on failure.
del(string $key [, int $id = null]) $key Unique storage key. $id If you stored multiple data in a key you can pass a unique ID to delete a specific entry.	Deletes a value from storage based on the key.	`storage()->del ('foo');`	null

Core\Storage\Object

When we return a data value it returns it as an object. Any value to store can always be accessed via the **value** property.

Property	Type	Description
id	int	Unique auto incremented storage ID.
key	string	Custom key you pass when accessing the storage methods.
value	mixed	Returns any values you stored in storage.
order	int	Order ID of a data set in storage.