

# Методы table

- `table.createItem()`
- `table.find()`
- `table.count()`
- `table.sum()`
- `table.max()`

# table.createItem()

## Создание записи

```
table.createItem(string $tableName, array|object $data)
```

### Описание:

Создает запись в таблице и возвращает значение этой записи, как объект.

### Атрибуты:

Имя	Тип	Описание
tableName	string	Наименование таблицы, в которой будет создана запись
data	array   object ( json{} )	Значения   значение записи

### Возвращает:

**ScriptCustomTableItem** - все значения полей записи в формате объекта.

### Пример:

Добавление данных в кастомную таблицу **regions**:

```
var region = {  
    "name": "Юго-Восточный",  
    "comment": "Создано через v8"  
};  
  
var newRegion = table.createItem('regions', region);
```

# table.find()

## Поиск записей

```
table.find(string $tableName, array|object $columns = [], array|object $where = [],  
array|object $orderBy = [], ?int $limit = null, ?int $offset = null): array
```

### Описание:

Ищет записи в таблице и возвращает их в виде массива объектов.

### Атрибуты:

Имя	Тип	Описание
tableName	string	Наименование таблицы, в которой будет найдена запись
columns	array   object ( json{} )	Столбцы которые будут возвращены в ответе
where	array   object ( json{} )	Условие, которое будет проверяться при поиске
orderBy	array   object ( json{} )	Устанавливает в какой последовательности будут возвращены выходные значения
limit	int	Устанавливает максимальное количество возвращенных значений
offset	int	Устанавливает смещение

### Возвращает:

**array** - массив объектов (найденных записей).

### Примеры:

Получение **всех** регионов (лимит 100):

```
var items = table.find('regions');  
var regions = [];
```

```
for(var i = 0; i < items.length; i++) { // добавление поочередно всех записей в объект regions
    var item = items[i];
    regions.push({"id": item.id, "name": item.name, "comment": item.comment});
}
```

Получение **трех** значений поля **region** с **id** **больше** **двух**:

```
var columns = ['region'];
var where = [{"id", ">", 2}];
var orderBy = [];
var limit = 3;
var offset = 0;

let items = table.find('regions', columns, where, orderBy, limit, offset);
```

# table.count()

## Подсчет количества записей

```
table.count(string $tableName, array|object $where = [])
```

### Описание:

Ищет записи соответствующие условию метода в таблице и возвращает их количество.

### Атрибуты:

Имя	Тип	Описание
tableName	string	Наименование таблицы, в которой будет создана запись
where	array   object ( json{ } )	Условие, которое будет проверяться при поиске

### Возвращает:

**int** - количество записей в формате числа.

### Примеры:

Получение количества **всех** записей в таблице:

```
var regionsCount = table.count("regions");
memory.setAttr('regionsCount', regionsCount);
```

Получение количества записей с **id больше 2**:

```
var countRegionIdsGreather2 = table.count("regions", [{"id", ">", 2}]);
memory.setAttr('countRegionIdsGreather2', countRegionIdsGreather2);
```

# table.sum()

## Подсчет суммы по полю

```
table.sum(string $tableName, string $column, array|object $where = [])
```

### Описание:

Ищет записи соответствующие условию метода в таблице и возвращает сумму их значений.

### Атрибуты:

Имя	Тип	Описание
tableName	string	Наименование таблицы, в которой будет создана запись
column	string	Наименование поля из которого будут браться значения
where	array   object ( json{ } )	Условие, которое будет проверяться при поиске

### Возвращает:

**int** - сумма всех значений поля в виде числа.

### Примеры:

Подсчет суммы **всех** значений записей поля **id** в таблице **regions**:

```
var sumRegionIds = table.sum("regions", 'id');  
memory.setAttr('sumRegionIds', sumRegionIds);
```

Подсчет суммы значений записей поля **id** в таблице **regions** с **id больше нуля**:

```
var sumRegionIds = table.sum("regions", 'id', [{"id", ">", 0}]);  
memory.setAttr('sumRegionIds', sumRegionIds);
```

# table.max()

## Поиск максимального значения по полю

```
table.max(string $tableName, string $column, array|object $where = [])
```

### Описание:

Ищет записи соответствующие условию метода в таблице и возвращает наибольшее по значению из них.

### Атрибуты:

Имя	Тип	Описание
tableName	string	Наименование таблицы, в которой будет создана запись
column	string	Наименование поля из которого будут браться значения
where	array   object ( json{ } )	Условие, которое будет проверяться при поиске

### Возвращает:

**mixed** - максимальное среди всех значений поля в том же типе, что в таблице

или

**null** - если максимальное значение не было найдено.

### Примеры:

Нахождение максимального значения в поле **id** в таблице **regions**:

```
var maxRegionId = table.max("regions", 'id');  
memory.setAttr('maxRegionId', maxRegionId);
```

Нахождение максимального значения в поле **id** в таблице **regions** с **id больше нуля**:

```
var maxRegionId = table.max("regions", 'id', [{"id", ">", 0}]);  
memory.setAttr('maxRegionId', maxRegionId);
```