

# ORM в 1С-Битрикс

Выборка данных

# Старый добрый getList

Метод ЕДИН для всех и подчиняется одним законам.

Принимает следующие параметры в виде массива:

`select` - имена полей, которые необходимо получить в результате

`filter` - описание фильтра для `WHERE` и `HAVING`

`group` - явное указание полей, по которым нужно группировать результат

`order` - параметры сортировки

`limit` - количество записей

`offset` - смещение для `limit`

`runtime` - динамически определенные поля

getList ВСЕГДА возвращает объекта класса DB\Result. Из него можно получить данные с помощью:

fetch() - используем в цикле для получения данных одного элемента

fetchAll() - получаем ВСЕ данные

```
$result = BookTable::getList($parameters);  
//Извлекаем данные так  
while ($row = $result->fetch()) {  
    $rows[] = $row;  
}  
//или так  
$rows = $result->fetchAll();
```

# select

Параметр `select` определяется в виде массива с именами полей сущности:

```
//Пример использования параметра select
BookTable::getList(['select' => ['ISBN', 'TITLE', 'PUBLISH_DATE']]);
//В результирующем массиве название поля PUBLISH_DATE будет заменено на PUBLICATION
BookTable::getList(['select' => ['ISBN', 'TITLE', 'PUBLICATION' => 'PUBLISH_DATE']]);
//Выбираем ВСЕ поля
BookTable::getList(['select' => ['*']]);
```

# filter

Если мы не указываем =, то по умолчанию будет выполнен LIKE.

```
// WHERE ID = 1
BookTable::getList(['filter' => ['=ID' => 1]]);
// WHERE TITLE LIKE 'Patterns%'
BookTable::getList(['filter' => ['%=TITLE' => 'Patterns%']]);

// WHERE ID = 1 AND ISBN = '9780321127426'
// 'LOGIC' => 'AND', // по умолчанию элементы склеиваются через AND
BookTable::getList([
    'filter' => [
        '=ID' => 1,
        '=ISBN' => '9780321127426'
    ]
]);
// WHERE (ID=1 AND ISBN='9780321127426') OR (ID=2 AND ISBN='9781449314286')
BookTable::getList([
    'filter' => [
        'LOGIC' => 'OR',
        ['=ID' => 1, '=ISBN' => '9780321127426'],
        ['=ID' => 2, '=ISBN' => '9781449314286']
    ]
]);
```

## group и order

В параметре `group` перечисляются поля для группировки:

```
BookTable::getList(['group' => ['PUBLISH_DATE']]);
```

Параметр `order` позволяет указать порядок сортировки:

```
BookTable::getList(['order' => ['PUBLISH_DATE' => 'DESC', 'TITLE' => 'ASC']]);  
//Направление по умолчанию - ASC  
BookTable::getList(['order' => ['ID']]);
```

# offset и limit

Параметры `offset` и `limit` помогут ограничить количество выбираемых записей или реализовать постраничную выборку:

```
// 10 последних записей
BookTable::getList([
    'order' => ['ID' => 'DESC'],
    'limit' => 10
]);

// 5-я страница с записями, по 20 на страницу
BookTable::getList([
    'order' => ['ID'],
    'limit' => 20,
    'offset' => 80
]);
```

# runtime

Позволяет использовать вычисляемые поля. В секции `runtime` можно регистрировать не только Expression поля, но и поля любых других типов. Механизм `runtime` работает таким образом, что к сущности добавляется новое поле, будто оно было описано в ней изначально в методе `getMap`. Но такое поле находится в зоне видимости только в рамках одного запроса - в следующем вызове getList такое поле уже будет недоступно, потребуется заново его зарегистрировать.

```
// выбрать дни, в которые выпущено более 5 книг
// SELECT PUBLISH_DATE, COUNT(*) AS CNT FROM my_book GROUP BY PUBLISH_DATE HAVING COUNT(*) > 5
BookTable::getList([
    'select' => ['PUBLISH_DATE'],
    'filter' => ['>CNT' => 5],
    'runtime' => [new Entity\ExpressionField('CNT', 'COUNT(*)')]
]);

// SELECT MAX(DATEDIFF(NOW(), PUBLISH_DATE)) AS MAX_AGE FROM my_book
BookTable::getList(['select' => [new Entity\ExpressionField('MAX_AGE', 'MAX(%s)', ['AGE_DAYS'])]]);
```

# Кеширование выборки

Для кеширования выборки указываем ключ `cache` в `getList`. По умолчанию выборки с JOIN не кешируются. Сброс кеша происходит в любом методе `add/update/delete`.

```
//Пример кеширования выборки
$res = \Bitrix\Main\GroupTable::getList(["filter"=>["=ID"=>1], "cache"=>["ttl"=>3600]]);
//Кешировать JOIN выборки
$res = \Bitrix\Main\GroupTable::getList(["filter"=>["=ID"=>1], "cache"=>["ttl"=>3600, "cache_joins"=>true]]);
//Очистка кеша
\Bitrix\Main\UserTable::getEntity()->cleanCache();
```

# Короткие вызовы

В дополнении к GetList существует еще ряд методов, которые позволяют в более короткой форме получить определенные данные:

`getById($id)` - производит выборку по первичному ключу;

`getByPrimary($primary, array $parameters)` - выборка по первичному ключу с возможностью указания дополнительных параметров;

`getRowById($id)` - производит выборку по первичному ключу, но возвращается массив данных (не нужно использовать `fetch`/`fetchAll`);

`getRow(array $parameters)` - производит выборку не по первичному ключу, а по каким-то другим параметрам. При этом возвращается только одна запись.