

## Слепая вера и здесь же Изобретение велосипеда

```
public static function allWithPagination($page, $taskLimit, $sort)
{
    $startId = $page * $taskLimit - $taskLimit;
    $query = Task::find()->setFirstResult($startId)->setMaxResults($taskLimit);

    if ($sort) {
        foreach ($sort as $key => $val) {
            if ($val === 'ASC' || $val === 'DESC') {
                $query = Task::find()->orderBy($key, $val)->setFirstResult($startId)->setMaxResults($taskLimit);
            }
        }
    }

    return static::all($query);
}
```

Ошибка: Не проверяю переданы ли аргументы, и корректны ли они. Ну реализация пагинации с сортировкой (эксклюзивчик от автора) :D

Решение: либо обернуть в try catch, либо сделать проверки аргументов с соответствующим выводом ошибок если необходимо. Ну а решение с пагинацией наверно надо делать в модели.

## Магические числа

```
public function index($page = 1, $taskLimit = 3)
{
    $auth = $this->request->session('auth');
    $sort = null;
    $countTasks = count(Task::all());

    $sort = $this->request->get();
    $data = Task::allWithPagination($page, $taskLimit, $sort);

    $pageLimit = ceil($countTasks / $taskLimit);
    return $this->render('index', [
        'data' => $data,
        'name' => $auth['login'],
        'pageLimit' => $pageLimit,
    ]);
}
```

Ошибка: собственно числа).

Решение: Передавать в виде констант, объявленных ранее в классе, либо вообще в файле конфигурации если возможно.

## Полтергейст

```
class PageController extends Controller
{
    public function index()
    {
        $this->request->redirect('/task');
    }
}
```

Ошибка: контроллер и метод созданы только для редиректа с главной страницы, т.к. посчитал что обрабатывать корневой запрос с помощью TaskController не корректно.

Решение: Может если какой то красивый способ это делать, но мне кажется лучше главную страницу сделать без вывода задач, и тогда можно метод нормально полноценно реализовать. А из навбара уже попадать на /task