

## Практическая работа №6

**Цель работы:** создание личного кабинета администратора.

**Формируемые компетенции:** ПК 5.4 «Производить разработку модулей информационной системы в соответствии с техническим заданием»

Для создания личного кабинета администратора необходимо выполнить следующие задачи:

**Задача 1:** С помощью генератора кода gii произвести генерацию CRUD контроллера:

- CRUD построить на основе модели с заказами, с использованием модели для поиска (AdminSearch).
- Название контроллера установить согласно заданию (если в задании определено по какому адресу должен быть личный кабинет пользователя), либо AdminController.

**Задача 2:** С помощью генератора кода gii произвести генерацию CRUD контроллера (CategoryAdminController) для управления категориями, с использованием модели для поиска (CategoryAdminSearch).

**Задача 3:**

Реализация ограничения доступа к контроллеру администратора.

Ниже представлены 2 варианта ограничения доступа к контроллеру администратора:

- вариант №1 – через фильтр доступа AccessControl;
- вариант №2 – через метод beforeAction контроллера.

Правильно использовать вариант №1, но если данная реализация вызывает затруднения, то использовать вариант №2.

Вариант №1.

Используя поведение behaviors и фильтр доступа AccessControl, настроить доступ к контроллеру панели администратора только зарегистрированного пользователя - администратора. Доступ определять через вызов matchCallback для правил и denyCallback.

<https://www.yiiframework.com/doc/guide/2.0/ru/security-authorization>

<https://itreviewchannel.ru/yii2-filtry-kontrolya-dostupa-access-control-filters/>

Пример реализации доступа:

```
public function behaviors()
{
    return array_merge(
        parent::behaviors(),
        [
            'access' => [
                'class' => AccessControl::class,
                'rules' => [
                    [
                        'allow' => true,
                        'roles' => ['@'],
                        'matchCallback' => function ($rule, $action) {
                            return Yii::$app->user->identity->isAdmin;
                        },
                    ],
                    [
                        'denyCallback' => function ($rule, $action) {
                            $this->goHome();
                        },
                    ],
                ],
            ],
            'verbs' => [
                'class' => VerbFilter::class,
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],
        ],
    );
}
```

Вариант №2.

Используя метод `beforeAction` контроллера, настроить доступ к контроллеру панели администратора только зарегистрированного пользователя - администратора. Метод должен возвращать булевый результат проверки, что пользователь является администратором.

[https://www.yiiframework.com/doc/api/2.0/yii-base-controller#beforeAction\(\)-detail](https://www.yiiframework.com/doc/api/2.0/yii-base-controller#beforeAction()-detail)

Пример реализации доступа:

```
public function beforeAction($action)
{
    if (!parent::beforeAction($action)) {
        return false;
    }

    return Yii::$app->user->identity->isAdmin;
}
```

**Задача 4:** С помощью генератора кода `gii` произвести генерацию CRUD контроллера:

- CRUD построить на основе модели с товарами, с использованием модели для поиска (`ProductAdminSearch`).
- Название контроллера установить `ProductAdminController`.

**Задача 5:** На главной странице лк администратора создать две кнопки:

- «Управление категориями» с url - `'admin/category-admin/index'`;
- «Управление товарами» с url - `'admin/product-admin/index'`;

**Задача 6:** В файле конфигурации приложения (`config/web.php`), в секции «`urlManager`», параметра «`rules`» создать правило для доступа в контроллеры «админки»:

```
'admin/<controller:(url контроллера админа для управления категориями|
url контроллера админа для управления товарами)>/<action>' =>
'<controller>/<action>'.
```

<https://www.yiiframework.com/doc/guide/2.0/ru/runtime-routing#parameterizing-routes>

Пример правил:

```
'urlManager' => [
    'enablePrettyUrl' => true,
    'showScriptName' => false,
    'rules' => [
        'admin/<controller:(category-admin|product-admin)>/<action>' => '<controller>/<action>',
    ],
],
```

**Задача 7:** Со всех страниц панели администратора удалить `breadcrumbs` («хлебные крошки»).

Проверить работоспособность приложения.