


Тема 4. «Реализация функционала пользователя»

Грушевский Ю.В.



Контроль доступа к методам

Authorization -> Access Control Filter

```
use yii\web\Controller;
use yii\filters\AccessControl;

class SiteController extends Controller
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::class,
                'only' => ['login', 'logout', 'signup'],
                'rules' => [
                    [
                        'allow' => true,
                        'actions' => ['login', 'signup'],
                        'roles' => ['?'],
                    ],
                    [
                        'allow' => true,
                        'actions' => ['logout'],
                        'roles' => ['@'],
                    ],
                ],
            ],
        ];
    }
    // ...
}
```



```
use yii\filters\AccessControl;

class SiteController extends Controller
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::class,
                'only' => ['special-callback'],
                'rules' => [
                    [
                        'actions' => ['special-callback'],
                        'allow' => true,
                        'matchCallback' => function ($rule, $action) {
                            return date('d-m') == '31-10';
                        },
                    ],
                ],
            ],
        ];
    }

    // Колбек сработал! Эта страница может быть отображена только 31-ого октября
    public function actionSpecialCallback()
    {
        return $this->render('happy-halloween');
    }
}
```

Доступ к регистрации только гостю

```
UserController.php
22 public function behaviors()
23 {
24     return array_merge(
25         parent::behaviors(),
26         [
27             'access' => [
28                 'class' => AccessControl::class,
29                 'only' => ['create'],
30                 'rules' => [
31                     [
32                         'actions' => ['create'],
33                         'allow' => true,
34                         'roles' => ['?'],
35                     ],
36                 ],
37             ],
38             'verbs' => [
39                 'class' => VerbFilter::className(),
40                 'actions' => [
41                     'delete' => ['POST'],
42                 ],
43             ],
44         ];
45 }
46 }
```



Реализация функции «Страница заявлений»

CRUD для заявок

CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for a data model.

Model Class

Search Model Class

Controller Class

View Path

Base Controller Class

My Application [Home](#) [About](#) [Contact](#) [Регистрация](#) [Login](#)

[Главная](#) / Requests

Requests

Create Request

#	ID	User ID	Status ID	Auto Number	Text
Ничего не найдено.					

Доступ к личному кабинету

```
RequestController.php x
16 class RequestController extends Controller
17 {
18     /** {@inheritdoc} ... */
21     public function behaviors()
22     {
23         return [
24             'access' => [
25                 'class' => AccessControl::className(),
26                 'only' => ['*'],
27                 'rules' => [
28                     [
29                         'actions' => ['index', 'create'],
30                         'allow' => true,
31                         'roles' => ['@'],
32                     ],
33                 ],
34             ],
35             'verbs' => [
36                 'class' => VerbFilter::className(),
37                 'actions' => [
38                     'delete' => ['POST'],
39                 ],
40             ],
41         ];
42     }
}
```

Перенаправления и ссылки

```
SiteController.php x
79 public function actionLogin()
80 {
81     if (!Yii::$app->user->isGuest) {...}
84
85     $model = new LoginForm();
86     if ($model->load(Yii::$app->request->post()) && $model->login()) {
87         if (!Yii::$app->user->identity->isAdmin()) {
88             return $this->redirect(['url' => '/request']);
89         }
90     }
91     return $this->goBack();
92 }
93
94 $model->password = '';
95 return $this->render('view:');
97 }
```

```
main.php x
43 ['label' => 'Contact', 'url' => ['/site/contact']],
44 ['label' => 'Регистрация', 'url' => ['/user/create'], 'visible'=>Yii::$app->user->isGuest],
45 ['label' => 'Личный кабинет', 'url' => ['/request/index'], 'visible'=>!Yii::$app->user->isGuest],
46
47 Yii::$app->user->isGuest ? (
48     ['label' => 'Вход', 'url' => ['/site/login']]
49 ) : (
50     '<li>'
51     . Html::beginForm(['/site/logout'], method: 'post')
52     . Html::submitButton(
53         content: 'Выход (' . Yii::$app->user->identity->username . ')',
```


Отображение только своих заявок

Enter your query

- Data Caching
- Data Formatting
- Data Providers
- Data widgets
- Database Access Objects
- Database Migration
- Dependency Injection Con...
- Entry Scripts
- Error Handling
- Events
- Extending ActiveForm on t...
- Extensions
- Filters
- Fixtures
- Fragment Caching
- Functional Tests
- Generating Code with Gii

Working with Data Keys

When using the data items returned by a `DataProvider`, you may want to use the custom `yii\data\DataProviderInterface::getModels()` method to filter the data by user ID.

```
use yii\data\ActiveDataProvider;

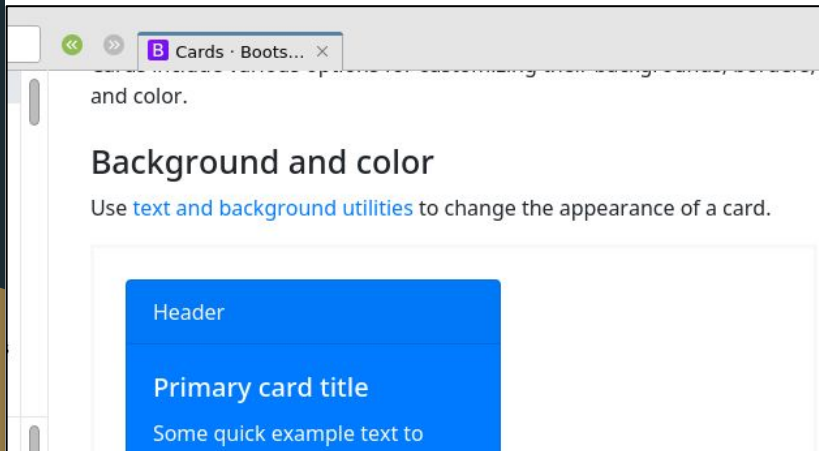
$query = Post::find()->where(['user_id' => Yii::$app->user->identity->getId()]);

$provider = new ActiveDataProvider([
    'query' => $query,
]);
```


RequestController.php

```
50  */
51  public function actionIndex()
52  {
53      $dataProvider = new ActiveDataProvider([
54          'query' => Request::find()->where(['user_id' => \Yii::$app->user->identity->getId()]),
55
56          'pagination' => [
57              'pageSize' => 5
58          ],
59          'sort' => [
60              'defaultOrder' => [
61                  'created_at' => SORT_DESC,
62              ]
63          ],
64      ]);
65
66
67      return $this->render( view: 'index', [
68          'dataProvider' => $dataProvider,
69      ]);
70  }
```

Исправление отображения



```
index.php x
21 <?php
22 foreach ($dataProvider->models as $model){
23
24     $classCard = match ($model->status->code) {
25         'approve' => 'text-white bg-success',
26         'rejected' => 'text-white bg-danger',
27         'new' => 'bg-light',
28     }
29
30     ?>
31     <div class="card mb-2 <?= $classCard ?>" >
32         <div class="card-body">
33             <h5 class="card-title"><?= $model->auto_number ?></h5>
34             <h6 class="card-subtitle mb-2 text-muted"><?= $model->status->name ?></h6>
35             <p class="card-text"><?= $model->text ?></p>
36         </div>
37     </div>
38 <?php }
39
40 echo LinkPager::widget([
41     'pagination' => $dataProvider->pagination,
42 ]);
43
44 ?>
45
```



Реализация функции «Добавления заявления»

Правка отображения формы

```
_form.php x
1 <?php
2
3 use ...
4
5
6 /** @var yii\web\View $this */
7 /** @var app\models\Request $model */
8 /** @var yii\widgets\ActiveForm $form */
9 ?>
10
11 <div class="request-form">
12
13     <?php $form = ActiveForm::begin(); ?>
14
15     <?= $form->field($model, attribute: 'auto_number')->textInput(['maxlength' => true]) ?>
16
17     <?= $form->field($model, attribute: 'text')->textarea(['rows' => 6]) ?>
18
19     <div class="form-group">
20         <?= Html::submitButton( content: 'Подать заявку', ['class' => 'btn btn-success']) ?>
21     </div>
22
23     <?php ActiveForm::end(); ?>
24
25 </div>
```

Автоматическое назначение автора

```
Request.php x
53 public function rules()
54 {
55     return [
56         [['user_id', 'status_id'], 'integer'],
57         [['user_id', 'default', 'value' => \Yii::$app->user->identity->getId()],
58         [['auto_number'], 'required'],
59         [['text'], 'string'],
60         [['created_at'], 'safe'],
61         [['auto_number'], 'string', 'max' => 255],
62         [['auto_number'], 'unique'],
63         [['status_id'], 'exist', 'skipOnError' => true,
64             'targetClass' => Status::class, 'targetAttribute' => ['status_id' => 'id']],
65         [['user_id'], 'exist', 'skipOnError' => true,
66             'targetClass' => User::class, 'targetAttribute' => ['user_id' => 'id']],
67     ];
68 }
```

Руссификация

```
create.php x
1 <?php
2
3 use yii\helpers\Html;
4
5 /** @var yii\web\View $this */
6 /** @var app\models\Request $model */
7
8 $this->title = 'Оставить новое заявление';
9 $this->params['breadcrumbs'][] = $this->title;
10
11 <div class="request-create">
12
13     <h1><?= Html::encode($this->title) ?></h1>
14
15     <?= $this->render( view: '_form', [
16         'model' => $model,
17     ]) ?>
18
19 </div>
```

```
Request.php x
52
53 no usages
54 public function attributeLabels()
55 {
56     return [
57         'id' => 'ID',
58         'user_id' => 'User ID',
59         'status_id' => 'Status ID',
60         'auto_number' => 'Номер автомобиля',
61         'text' => 'Описание нарушения',
62         'created_at' => 'Created At',
63     ];
64 }
```

Редирект

```
RequestController.php x
90 public function actionCreate()
91 {
92     $model = new Request();
93
94     if ($this->request->isPost) {
95         if ($model->load($this->request->post()) && $model->save()) {
96             return $this->redirect(['index']);
97         }
98     } else {
99         $model->loadDefaultValues();
100     }
101
102     return $this->render( view: 'create', [
103         'model' => $model,
104     ]);
105 }
```



Добавление уведомлений

Уведомление об успешной регистрации

```
UserController.php
78 public function actionCreate()
79 {
80     $model = new User();
81
82     if ($this->request->isPost) {
83         if ($model->load($this->request->post()) && $model->save()) {
84             Yii::$app->getSession()->setFlash( key: 'success', value: 'Поздравляем! Вы успешно зарегистрировались!');
85             return $this->redirect(['site/login']);
86         }
87     } else {
88         $model->loadDefaultValues();
89     }
90
91     return $this->render( view: 'create', [
92         'model' => $model,
93     ]);
94 }
```




Добавление анимаций

Пользовательский js



```
AppAsset.php x
11
12  /**
13   * Main application asset bundle.
14   *
15   * @author Qiang Xue <qiang.xue@gmail.com>
16   * @since 2.0
17   */
18  class AppAsset extends AssetBundle
19  {
20      public $basePath = '@webroot';
21      public $baseUrl = '@web';
22      public $css = [
23          'css/site.css',
24      ];
25      public $js = [
26          'js/main.js'
27      ];
28      public $depends = [
29          'yii\web\YiiAsset',
30          'yii\bootstrap5\BootstrapAsset'
31      ];
32  }
```

Реализация анимаций

 .animate() x

To animate the opacity, left offset, and height of the image simultaneously:

```
1 | $( "#clickme" ).on( "click", function() {  
2 |   $( "#book" ).animate({  
3 |     opacity: 0.25,  
4 |     left: "+=50",  
5 |     height: "toggle"  
6 |   }, 5000, function() {  
7 |     // Animation complete.  
8 |   });  
9 | });
```

 main.js x

```
1 | $.alert().animate( prop: {  
2 |   opacity: 0.25,  
3 | }, speed: 1000, easing: function () : void {  
4 |   $.alert().hide('show');  
5 | });  
6 |
```

Задание (45 мин)

Добавление уведомлений и анимаций

- Добавьте уведомления на все активные действия пользователя
- Подберите и интегрируйте другую анимацию для уведомлений на сайте