

# 1 Delo z Yii

## 1.1 Postavitev strani

Postavitve strani (*layout*), so definirane v direktoriji `~/basic/views/layouts`. Po namestitvi najdemo datoteko `main.php`. Spletno stran razdeli na 3 funkcionalne dele:

- vrstico za krmarjenje (*navigation bar*)
- osrednji del spl. strani
- nogo (*footer*)

Vsebina navigacijske vrstice in noge je zapisna v tej datoteki. Spremenimo napise v meniju in nogi, namesto besedila z imenom podjetja (levi zgornji kot) bomo prikazali sliko z logotipom.

```
...
<?php
    $logo = Yii::$app->request->baseUrl . '/images/logo.png';
    NavBar::begin([
        // 'brandLabel' => 'My Company', izvorna koda
        'brandLabel' => Html::img($logo),
        'brandUrl' => Yii::$app->homeUrl,
        'options' => [
            'class' => 'navbar-inverse navbar-fixed-top',
        ],
    ]);
    echo Yii::$app->request->baseUrl;
    echo Nav::widget([
        'options' => ['class' => 'navbar-nav navbar-right'],
        'items' => [
            ['label' => 'Domov', 'url' => ['/site/index']],
            ['label' => 'O nas', 'url' => ['/site/about']],
            ['label' => 'Kontakt', 'url' => ['/site/contact']],
            Yii::$app->user->isGuest ? (
                ['label' => 'Prijava', 'url' => ['/site/login']]
            ) : (
                '<li>'
                . Html::beginForm(['/site/logout'], 'post')
                . Html::submitButton(
                    'Odjava (' . Yii::$app->user->identity->username .
                ')',
                ['class' => 'btn btn-link logout']
                )
                . Html::endForm()
                . '</li>'
            )
        ],
    ]);
    NavBar::end();
?>
...
```

Po potrebi v meni dodamo nove vnose, oz. obstoječe brišemo. Metoda `Yii::$app->request->baseUrl` vrne direktorij Yii aplikacije, tj. `~/basic/web`. Tu je priporočeno kreirati direktorije za slike in ostale gradnike spletne strani, v našem primeru `images`. Podobno še na nogi

spremenimo napis na dnu datoteke.

```
<footer class="footer">
    <div class="container">
        <p class="pull-left">&copy; ŠC Ptuj, Elektro in računalniška šola
            <?= date('Y') ?></p>
        <p class="pull-right"><?= Yii::powered() ?></p>
    </div>
</footer>
```

Vsebine osrednjega dela strani je shranjena na direktoriju basic/views/site.

Datoteka s slogi site.css se nahaja na /basic/web/css. Barve navigacijske vrstice ne najdemo, če smo pozorno pogledamo datoteko main.php, obnašanje te vrstice je določeno z:

```
...
'class' => 'navbar-inverse navbar-fixed-top',
...
```

Vrednost navbar-inverse zamenjamo z barvo, ki jo dodamo v site.css, primer:

```
...
.nav-vrstica {
    background-color: #28166f;
}
```

### 1.1.1 Naslov strani

Naslov strani določimo v datotekah vsebine (direktorijski views/site), naslov je namreč prikazan v vsebini strani, npr. datoteka about.php naj ima naslov "O nas".

```
<?php
...
$this->title = 'O nas';
$this->params['breadcrumbs'][] = $this->title;
?>
...
```

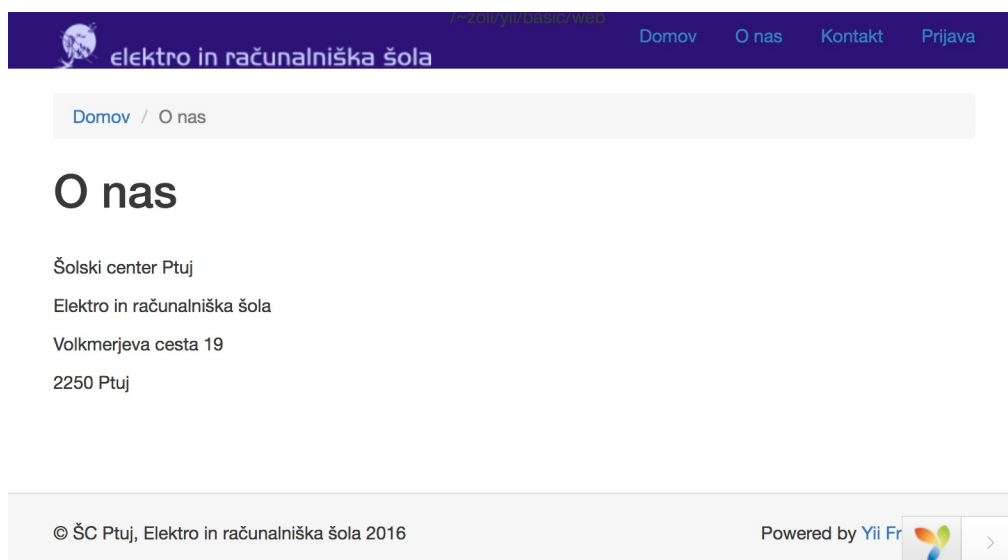
**Pomembno!** Naslov strani mora biti HTML zakodirano v nadaljevanju:

```
...
<h1><?= Html::encode($this->title) ?></h1>
...
```

Pogosto je koristno, da se na spletni strani prikaže tudi trenutna pot (*Breadcrumbs*), po kateri lahko tudi navigiramo. Začetek trenutne poti kaže na domači url, označen je z napisom "Home". Napis spremenimo v kontrolniku Breadcrumbs v datoteki main.php:

```
<?= Breadcrumbs::widget([
    'homeLink' => ['label' => 'Domov', 'url' => Yii::$app->getHomeUrl()],
    'links' => isset($this->params['breadcrumbs']) ?
        $this->params['breadcrumbs'] : [],
]) ?>
<?= $content ?>
```

Slika 1 prikazuje stran "O nas" po ureditvi vsebine datoteke about . php.



Slika 1: Stran "O nas"

## 2 Delo z obrazci

Z obrazci pridobivamo podatke od uporabnikov psletnih strani. Poleg pogledov in akcij bomo pri delu z obrazci izdelali še model. Model uporabimo za predstavitev uporabnikov podatkov. Izdelava strani z obrazci poteka v treh korakih:

- izdelava modela in določitev pravil za validacijo podatkov
- izdelava akcije
- izdelava pogleda s HTML obrazci

### 2.1 Izdelava modela

Podatke uporabnikov (ime, priimek, el. naslov) predstavimo z razredom `EntryForm`, shranimo v datoteko `models/EntryForm.php`:

```
<?php

namespace app\models;
use yii\base\Model;

class EntryForm extends Model
{
    public $ime;
    public $priimek;
    public $email;

    public function rules()
    {
        return [
            [['ime', 'priimek', 'email'], 'required', 'message' => 'To
polje je zahtevano'],
            [['email', 'email'], 'email', 'message' => 'Elektronski naslov ni
veljavne oblike'],
        ];
    }
}
```

Razred `EntryForm` deduje lastnosti osnovnega razreda `yii\base\Model`, ki se običajno uporablja za predstavitev podatkov. Vsebuje tri javne razredne spremenljivke, v katera se bodo shranjevali podatki vnešene s strani uporabnikov in metodo `rules()`, ki vrača množico pravil za validacijo podatkov, te so:

- vsi trije podatki so zahtevani – lastnost `required`
- elektronski naslov mora biti pravilne oblike – lastnost `email`

Če podatki niso veljavni v skladu s pravili, se v brskalniku izpiše sporočilo podano z lastnostjo `message`. Če ta parameter izpustimo se izpiše privzeto sporočilo.

## 2.2 Izdelava akcije

V `controllers/SiteController.php` dodamo akcijo `actionEntry()`, za podrobnosti glej poglavje Napaka: vira sklicevanja ni mogoče najti. Po uporabnikovem vnosu in validaciji bo prikazal podatke, ki jih je uporabnik vnesel, oz. sporočila ob neveljavnih podatkih.

```
<?php
namespace app\controllers;
...
use app\models\EntryForm;      // vključimo model, ki smo ga prejle
definirali

...
public function actionEntry()
{
    $model = new EntryForm();
    if ($model->load(Yii::$app->request->post()) && $model-
>validate())
    {
        // podatki v $model so veljavni
        return $this->render('entry-confirm', ['model' => $model]);
    } else {
        // vsaj eno polje ni bilo pravilno izpolnjeno
        return $this->render('entry', ['model' => $model]);
    }
}
...
```

## 2.3 Izdelava pogledov

Izdelamo še dva pogleda, `entry` bo vseboval obrazec za nos podatkov. Shranimo v `views/site/entry.php`:

```
<?php
use yii\helpers\Html;
use yii\widgets\ActiveForm;
?>
<?php $form = ActiveForm::begin(); ?>
    <?= $form->field($model, 'ime') ?>
    <?= $form->field($model, 'priimek') ?>
    <?= $form->field($model, 'email')->label('El. naslov') ?>
    <div class="form-group">
        <?= Html::submitButton('Pošlji', ['class' => 'btn btn-
primary']) ?>
    </div>
<?php ActiveForm::end(); ?>
```

Napis vnosnih polj Yii vzame iz imena lastnosti, če želimo drug napis, storimo to z metodo `label` kot je v primeru za polje `email`.

V pogledu `views/site/entry-confirm` prikažemo vnešene podatke, če so le ti prestali validacijo določene v modelu.

```

<?php
use yii\helpers\Html;
?>

<p>Vnesel si sledeče podatke:</p>
<ul>
    <li><label>Ime</label>: <?= Html::encode($model->ime) ?></li>
    <li><label>Priimek</label>: <?= Html::encode($model->priimek) ?></li>
    <li><label>El. naslov</label>: <?= Html::encode($model->email) ?></li>
</ul>

```

## 2.4 Preizkus aplikacije

Delovanje aplikacije preizkusim, tako da v brskalnik vnesemo URL s pogledom entry:

```
http://hostname/index.php?r=site/entry
```

Prikaže stran z obrazcem, ki smo ga izdelali.

elektro in računalniška šola

Domov O nas Kontakt Prijava

Ime

Priimek

El. naslov

Pošlji

© ŠC Ptuj, Elektro in računalniška šola 2016

Powered by Yii Framework

Slika 2: Stran z obrazcem

Če podatki niso prestali validacijo, se prikažejo sporočila, katere podatke moramo dopolniti.

Slika 3: Sporočila za dopolnitev in korekcijo podatkov

Po vnosu pravih podatkov se prikaže pogled `entry-confirm`.

Slika 4: Stran po pravilno izpolnjem obrazcu

Za zaključek obrazec dodamo še v navigacjsko vrstico, da bi bila lažje dosegljiv. V dodatki `views/layouts/main.php` v kontrolnik `Nav::widget` dodamo nov element.

```
...
echo Nav::widget([
    'options' => ['class' => 'navbar-nav navbar-right'],
    'items' => [
        ['label' => 'Domov', 'url' => ['/site/index']],
        ['label' => 'O nas', 'url' => ['/site/about']],
        ['label' => 'Kontakt', 'url' => ['/site/contact']],
        ['label' => 'Vnos', 'url' => ['/site/entry']],
    ]
]);
...
```

### 3 Literatura in viri:

- [1] The Definitve Guide to Yii 2.0 - <http://www.yiiframework.com/doc-2.0/guide-start-workflow.html>
- [2] 10 PHP Frameworks For Developers – Best Of <http://www.hongkiat.com/blog/best-php-frameworks/>
- [3] When & Why to Use PHP Framework? – <http://www.templatemonster.com/blog/php-frameworks-when-and-why-to-use-them/>
- [4] How to choose a PHP framework, <https://opensource.com/business/16/6/which-php-framework-right-you>