

Министерство науки и высшего образования РФ
Государственное образовательное учреждение высшего
профессионального образования
Белгородский Государственный Технологический Университет им
В. Г. Шухова

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем.

Лабораторная работа №3 по дисциплине
«Технологии Web-программирования»

Тема: Серверное программирование. Yii2

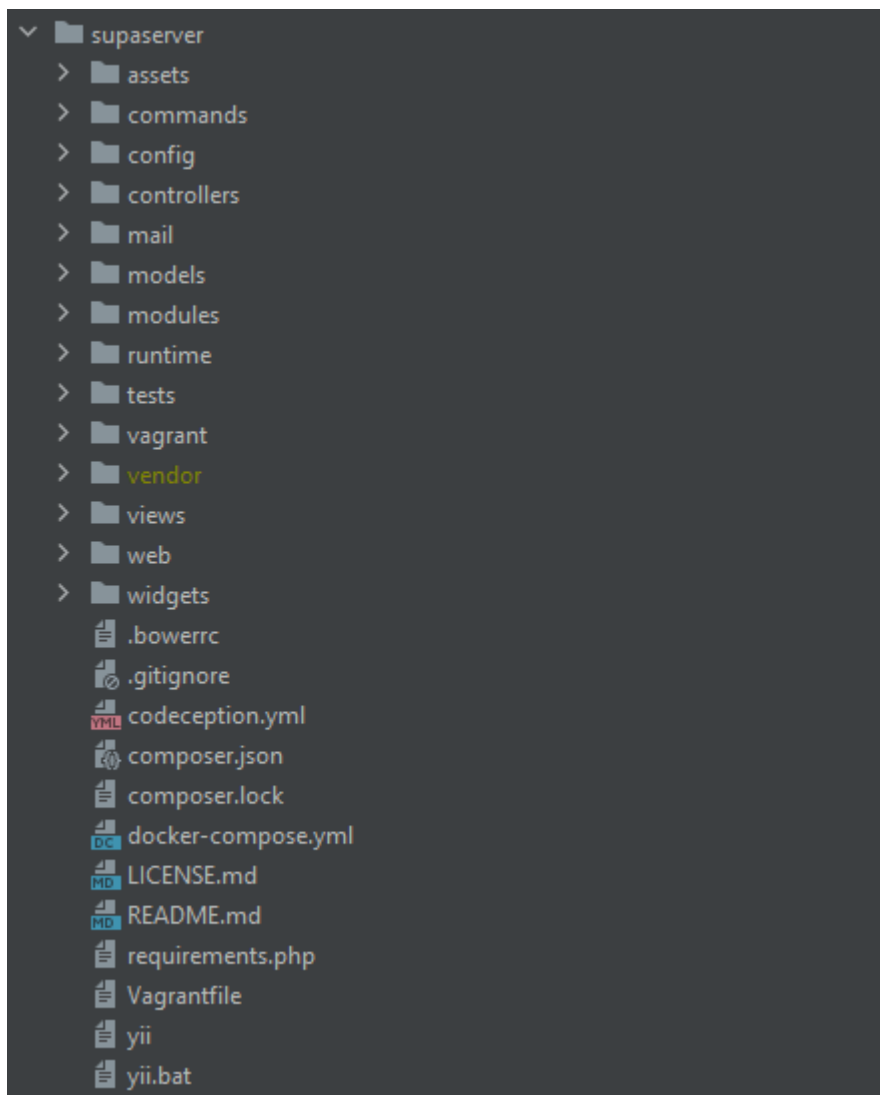
Выполнил:
студент группы ПВ-41
Колесников В.Д.
Проверил: Поляков В.К.

Белгород 2019 г.

Цель работы: познакомиться с основами backend разработки web-приложений. Научится писать скрипты на языке PHP. Познакомиться с основами работы docker. Познакомиться с фреймворком Yii2 и научиться разворачивать проект, производить его настройку. Научится работать с API в приложении Postman.

Выполнение

Было развёрнуто базовое приложение Yii2 с помощью php-композера командой `composer create-project --prefer-dist yiisoft/yii2-app-basic basic`.



Далее был настроен файл конфигурации `docker-compose.yml`

```
version: '3.1'
services:
  php:
    image: yiisoft/yii2-php:7.3-apache
    volumes:
      - ~/.composer-docker/cache:/root/.composer/cache:delegated
      - ../app:delegated
    ports:
      - '1199:80'
    extra_hosts:
      - "visiticelandbeauty.com:127.0.0.1"
```

```
mysql:
  image: mysql
  command: --default-authentication-plugin=mysql_native_password --character-set-server=utf8 --collation-server=utf8_general_ci
  environment:
    - MYSQL_ROOT_PASSWORD=verysecret
    - MYSQL_DATABASE=db
    - MYSQL_USER=user
    - MYSQL_PASSWORD=password
```

Был добавлен API модуль версии 1 со следующими контроллерами и моделями:

User.php

```
<?php
```

```
namespace app\models;
```

```
class User extends \yii\base\BaseObject implements \yii\web\IdentityInterface
{
    public $id;
    public $username;
    public $password;
    public $authKey;
    public $accessToken;

    private static $users = [
        '100' => [
            'id' => '100',
            'username' => 'admin',
            'password' => 'admin',
            'authKey' => 'test100key',
            'accessToken' => '100-token',
        ],
        '101' => [
            'id' => '101',
            'username' => 'demo',
            'password' => 'demo',
            'authKey' => 'test101key',
            'accessToken' => '101-token',
        ],
    ];

    /**
     * {@inheritdoc}
     */
    public static function findIdentity($id)
    {
        return isset(self::$users[$id]) ? new static(self::$users[$id]) : null;
    }

    /**
     * {@inheritdoc}
     */
    public static function findIdentityByAccessToken($token, $type = null)
    {
        foreach (self::$users as $user) {
            if ($user['accessToken'] === $token) {
                return new static($user);
            }
        }

        return null;
    }

    /**
     * Finds user by username
     *
     * @param string $username
     * @return static|null
     */
    public static function findByUsername($username)
    {
        foreach (self::$users as $user) {
            if (strcasecmp($user['username'], $username) === 0) {
                return new static($user);
            }
        }
    }
}
```

```

        }
    }

    return null;
}

/**
 * {@inheritdoc}
 */
public function getId()
{
    return $this->id;
}

/**
 * {@inheritdoc}
 */
public function getAuthKey()
{
    return $this->authKey;
}

/**
 * {@inheritdoc}
 */
public function validateAuthKey($authKey)
{
    return $this->authKey === $authKey;
}

/**
 * Validates password
 *
 * @param string $password password to validate
 * @return bool if password provided is valid for current user
 */
public function validatePassword($password)
{
    return $this->password === $password;
}
}

```

ApiController.php

```
<?php
```

```

namespace app\modules\v1\controllers;

use yii\filters\auth\CompositeAuth;
use yii\filters\ContentNegotiator;
use yii\filters\Cors;
use yii\filters\RateLimiter;
use yii\filters\VerbFilter;
use yii\rest\Controller;
use yii\web\Response;

class ApiController extends Controller
{
    public function behaviors()
    {
        return [
            'contentNegotiator' => [
                'class' => ContentNegotiator::className(),
                'formats' => [
                    'application/json' => Response::FORMAT_JSON
                ],
            ],
            'verbFilter' => [
                'class' => VerbFilter::className(),
                'actions' => $this->verbs(),
            ],
            'authenticator' => [
                'class' => CompositeAuth::className(),
            ],
            'rateLimiter' => [
                'class' => RateLimiter::className(),
            ],
            'corsFilter' => [
                'class' => Cors::class
            ],
        ];
    }
}

```

```

    ]
  };
}
}

```

EmotesController.php

```
<?php
```

```

namespace app\modules\v1\controllers;

class EmotesController extends ApiController {
    public function actionHome() {
        $emotes = [
            [
                'id' => 1,
                'name' => 'Okayga',
                'description' => 'FeelsOkayMan + Pepega.',
                'image' => '@@/assets/images/Okayga.png',
            ],
            [
                'id' => 2,
                'name' => 'PeepoClown',
                'description' => 'Because twitch viewers are often clowns.',
                'image' => '@@/assets/images/PeepoClown.png',
            ],
            [
                'id' => 3,
                'name' => 'forsenCD',
                'description' => 'D OMEGALUL C emote.',
                'image' => '@@/assets/images/forsenCD.jpg',
            ],
            [
                'id' => 4,
                'name' => 'FeelsAmazingMan',
                'description' => 'When you are totally happy.',
                'image' => '@@/assets/images/FeelsAmazingMan.png',
            ],
            [
                'id' => 5,
                'name' => 'Pepega',
                'description' => 'Absolute retardness.',
                'image' => '@@/assets/images/Pepega.png',
            ],
            [
                'id' => 6,
                'name' => 'forsenE',
                'description' => 'The face of Twitch.',
                'image' => '@@/assets/images/forsenE.jpg',
            ],
            [
                'id' => 7,
                'name' => 'KKomrade',
                'description' => 'Fellow russians emote',
                'image' => '@@/assets/images/KKomrade.png',
            ],
            [
                'id' => 7,
                'name' => '5Head',
                'description' => 'Intellectual.',
                'image' => '@@/assets/images/5Head.png',
            ]
        ];
        return $emotes;
    }
}

```

UserController.php

```
<?php
```

```

namespace app\modules\v1\controllers;

use app\modules\v1\models\User;
use yii\rest\Controller;

class UserController extends ApiController {
    public function actionLogin() {
        // login stuff
    }
}

```

```

public function actionRegister() {
    // register stuff
}

public function actionAll() {
    $user = new User();
    return $user -> getUsers();
}
}

```

Дале были проверены запросы в Postman:

The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:1199/api/v1/user/all`. The 'Query Params' section is empty. The 'Body' tab is selected, showing a JSON array of three user objects. The 'Headers' tab shows 7 headers.

KEY	VALUE
Key	Value

```

1  [
2    {
3      "id": 1,
4      "login": "pepega",
5      "password": "forsan"
6    },
7    {
8      "id": 2,
9      "login": "okayge",
10     "password": "cock"
11   },
12   {
13     "id": 3,
14     "login": "bruceu",
15     "password": "zulul"
16   }
17 ]

```

GET
http://127.0.0.1:1199/api/v1/emotes/home

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings

Query Params

KEY	VALUE	DESCR
Key	Value	Descri

Body
Cookies
Headers (7)
Test Results

Pretty
Raw
Preview
Visualize
JSON

```

19  },
20  {
21      "id": 4,
22      "name": "FeelsAmazingMan",
23      "description": "When you are totally happy.",
24      "image": "@assets/images/FeelsAmazingMan.png"
25  },
26  {
27      "id": 5,
28      "name": "Pepega",
29      "description": "Absolute retardness.",
30      "image": "@assets/images/Pepega.png"
31  },
32  {
33      "id": 6,
34      "name": "forsenE",
35      "description": "The face of Twitch.",
36      "image": "@assets/images/forsenE.jpg"
37  },
38  {
39      "id": 7,
40      "name": "KKomrade",
41      "description": "Fellow russians emote",
42      "image": "@assets/images/KKomrade.png"
43  },
44  {
45      "id": 7,
46      "name": "5Head",
47      "description": "Intellectual.",
48      "image": "@assets/images/5Head.png"

```

Вывод: В результате выполнения работы были изучены основы backend-разработки web-приложений и скрипты на языке PHP. Также были рассмотрены основы работы с docker и фреймворком Yii2 а также приобретены навыки работы с API в приложении Postman.