

Лабораторные работы по курсу РНР: Проектирование и разработка сложных веб – проектов

Лабораторная работа 5.1

Работа с curl в консольном режиме

Упражнение 1. Создание и выполнение запроса через curl

1. Откройте меню **OpenServer** (нажмите флажок в нижней правой части экрана)
2. Выберите **Дополнительно** -> **Консоль**
3. Напишите запрос к **curl**, который вернет HTML-страницу <http://www.specialist.ru> вместе с HTTP-заголовками (<http://osxh.ru/terminal/command/curl>)
4. Запустите запрос и убедитесь, что в консоли отображаются HTTP-заголовки и содержимое страницы

Упражнение 2. Сохранение результатов запроса в файл

1. Измените предыдущий запрос так, чтобы результаты сохранялись в файл с расширением **html**
2. Уберите флаг показа HTTP-заголовков
3. Отправьте запрос к **curl**
4. Убедитесь, что появился HTML-файл с содержимым страницы

Лабораторная работа 5.2

Работа с curl из РНР

Упражнение 1. Создание и выполнение запроса к curl через РНР

1. Создайте файл **curl.php**
2. Инициализируйте ресурсный указатель
3. Задайте настройки для **curl**:
 - a. **CURLOPT_POST** должна идти вместе с **true**
 - b. **CURLOPT_POSTFIELDS** – содержать массив данных, отправляемых **POST**
 - c. Если в первом задании текущей лабораторной не назначили **URL**, установите **CURLOPT_URL** в значение **test_curl.php**
4. Создайте файл **test_curl.php**, который принимает значения из **POST**
5. Внутри **test_curl.php** напишите код обрабатывающий данные **POST**
 - a. Просто выводятся данные на экран
 - b. Вычисляющий некоторое значение на основе входных данных
6. Выполните запрос через **curl**, используя функцию **curl_exec()**

Упражнение 2. Проверка работы

1. Убедитесь, что в **curl.php** сформирован корректно
2. Проверьте работу **curl.php** запуском его в браузере
3. Исправьте ошибки

Лабораторная работа 6.1

Работа с регулярными выражениями в PHP

Упражнение 1. Использование функции замены текста

1. Создайте файл **pcre.php**
2. Напишите регулярное выражение, которое будет менять шаблон **@username@** на «John», а **@price@** на «100500»
3. Прочитайте содержимое файла **message.tpl**
4. Замените шаблоны при помощи регулярного выражения при помощи функции **preg_replace()**
5. Выведите на экран измененную строку
6. Убедитесь в правильности работы, запустив скрипт на своем сайте
7. Доп. задание: Перепишите пример с использованием функции **preg_replace_callback()**

Упражнение 2. Использование функции **preg_match_all()**

1. В файле **pcre.php** прочитайте содержимое сайта <http://www.specialist.ru>
2. Напишите регулярное выражение, которое найдёт все гиперссылки на сайте
3. Используя функцию **preg_match_all()** найдите и выведите все вхождения в прочитанном содержимом
4. Убедитесь в правильности работы скрипта

Лабораторная работа 8.1

Организация тестирования при помощи PHPUnit

Упражнение 1. Создание тестового класса

1. Создайте файл с тестовым классом в папке **work/tests/** (для тестирования можно выбрать любой класс, созданный на предыдущих занятиях)
2. Создайте у тестового класса два тестовых метода
3. Укажите зависимость одного метода от другого
4. Запустите тестирование командой
php phpunit.phar --bootstrap ВашКласс.php tests/ВашКлассTest.php
5. Убедитесь в том, что тест пройден, изучите результаты

Упражнение 2. (Для самостоятельной домашней работы) Создание остальных тестовых классов

1. Создайте тестовые классы для каждого класса курса
2. Запустите тестирование всех классов проекта (при запуске нужно указывать только название папки)
3. Проверьте результаты и исправьте ошибки в существующих классах

Лабораторная работа 8.1

Создание MVC-фреймворка

Упражнение 1: Создание поведения

1. Перенесите содержимое папки **C:\Users\Public\OpenServer\domains\mysite.local\mvc** на уровень выше, то есть в папку **mysite.local**

2. В папке **application\controllers** создайте файл **UserController.php**
3. В файле **UserController.php** создайте класс **UserController**, который наследует интерфейс **IController**
4. В классе **UserController** создайте действие (action) **hello**, то есть метод **helloAction**
5. Метод должен быть отработан при получении URL вида **/user/hello/name/john**
6. Получите необходимые параметры и передайте их модели **FileModel**
7. Представление для данного действия находится в папке **views**, имя файла - константа **8. USER_DEFAULT_FILE**
9. Сохраните файл **application\controllers\UserController.php**
10. Запустите браузер
11. В адресной строке наберите **http://mysite.local/user/hello/name/john**
12. Вы должны получить строку **Hello, john!**
13. Проверьте других пользователей

Упражнение 2: Создание действия и представления для вывода списка пользователей

1. В классе **UserController** создайте действие (action) **list**
2. Метод должен быть отработан при получении URL вида **/user/list/**
3. Задача: получить и вывести список пользователей
4. Получите необходимые параметры и передайте их модели **FileModel**
5. Представление для данного действия находится в папке **views**, имя файла - константа **USER_LIST_FILE**
6. Сохраните файл **application\controllers\UserController.php**
7. В тестовом редакторе откройте файл **application\views\user_list.php**
8. Выведите список пользователей в виде HTML-списка
9. Сохраните файл **application\views\user_list.php**
10. Запустите браузер
11. В адресной строке наберите **http://mysite.local/user/list**
12. Вы должны увидеть список пользователей в виде HTML-списка

Упражнение 3: Создание действия и представления для вывода роли пользователя

1. В классе **UserController** создайте действие (action) **get**
2. Метод должен быть отработан при получении URL вида **/user/get/role/john/**
3. Задача: получить и вывести роль пользователя
4. Получите необходимые параметры и передайте их модели **FileModel**
5. Проверьте, есть ли данный пользователь в списке
6. Представление для данного действия находится в папке **views**, имя файла - константа **USER_ROLE_FILE**
7. Сохраните файл **application\controllers\UserController.php**
8. В тестовом редакторе откройте файл **application\views\user_role.php**
9. Если пользователь существует, то выведите строку **User имя_пользователя is роль_пользователя**
10. Если пользователь в списке отсутствует, то выведите строку **Unknown user имя_пользователя**
11. Сохраните файл **application\views\user_role.php**
12. Запустите браузер
13. В адресной строке наберите **http://mysite.local/user/get/role/john**
14. Вы должны увидеть строку **User john is admin**

15. Проверьте других пользователей
16. Введите имя, которое отсутствует в списке

Упражнение 4: Создание действия и представления для добавления пользователя в список

1. В классе `UserController` создайте действие (action) **add**
2. Метод должен быть отработан при получении URL вида
/user/add/name/vasya/role/admin
3. Задача: добавить переданное имя пользователя и его роль в общий список
4. Получите необходимые параметры, добавьте нового пользователя в список и передайте параметры модели **FileModel**
5. Представление для данного действия находится в папке **views**, имя файла - константа **USER_ADD_FILE**
6. Сохраните файл **application\controllers\UserController.php**
7. В тестовом редакторе откройте файл **application\views\user_add.php**
8. Выведите строку **New user имя_пользователя : роль_пользователя**
9. Сохраните файл **application\views\user_role.php**
10. Запустите браузер
11. В адресной строке наберите **http://mysite.local/user/add/name/vasya/role/admin**
12. Вы должны увидеть строку **New user vasya : admin**
13. В адресной строке наберите **http://mysite.local/user/list**
14. Вы должны увидеть нового пользователя в общем списке

Лабораторная работа 10.1

Работа с REST-фреймворком

Упражнение 1. Работа с каркасным приложением на Slim (Skeleton)

1. Откройте страницу пошаговое руководство по созданию вебприложения на Slim
<http://www.slimframework.com/docs/tutorial/first-app.html>
2. Найдите ссылку на Skeleton-приложение на github
3. Перейдите по ссылке и изучите код основных файлов
4. Установите это приложение и запустите в браузере главную страницу приложения

Упражнение 2. Старт работы с Zend Framework 2

1. Повторите действия первого упражнения для ZF2

Упражнение 3. * Старт работы с ...

1. Выберите произвольный PHP-фреймворк
2. Установите фреймворк через composer

Спасибо, что занимались на курсе PHP4!

Пожалуйста, пришлите свои пожелания по улучшению курса на мой емейл atarasov@specialist.ru или в закрытую группу ВК <http://vk.com/phpspec>

Удачной работы!