

Алгоритмизация и программирование

Лекция 5.2

Создание бэкенда сайта. Продолжение

Советы для проектирования интерфейса администратора

Проектирование приложения целесообразно начинать с построения структуры данных.

Исходя из задачи, приведённой в лекции 5.2, данные, формируемые и используемые приложением должны храниться в базе данных, поэтому построение структуры данных начинается с разработки таблиц.

Специально для этого проекта создана база данных **backend**.

На данном этапе проектирования в этой базе создана пока одна таблица **content**.

[127.0.0.1](#) » [backend](#)

Таблица	Строки	Тип	Размер	Комментарии	
content	3	InnoDB	16 КБ	Создание:	Ноя 15 2023 г., 12:56
1 таблица	3	--	16 КБ		

Структура таблицы content приведена на следующем рисунке:

[127.0.0.1](#) » [backend](#) » [content](#)

content

Поле	Тип	Null	По умолчанию	Комментарии	MIME
id	int(10)	Нет			
razdely	varchar(255)	Нет			
content	text	Нет			

Поле id задано, как уникальный первичный ключ PRIMARY AUTOINCREMENT. Его значения должны формироваться автоматически по мере наполнения таблицы.

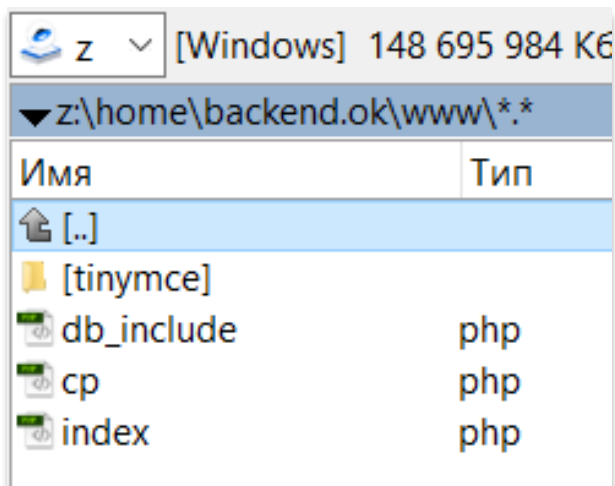
Индексы

Имя индекса	Тип	Уникальный	Упакован	Поле	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	id	3	A	Нет	

Поле `razdely` предназначено для хранения наименований разделов. Из значений этого поля будут формироваться меню администратора в бэкенде и пользовательское меню во фронтенде.

Поле `content` предназначено для хранения контента разделов. Контент будет формироваться с помощью редактора TinyMCE.

Файловая структура проекта



На данном этапе проектирования предполагается следующая файловая структура проекта:

- папка плагина `tinymce`;
- файл `db_include.php`;
- файл `cp.php`;
- файл `index.php`

Папка `tinymce` содержит в себе файлы, необходимые для работы редактора `tinymce`. Описание редактора приведено в лекции [«Лекция 5 Алгоритмизация и программирование. Применяем PHP»](#)

Файл `db_include.php` содержит код для соединения и подключения к базе данных. Его описание содержится в лекции [«Взаимодействие PHP и MySQL»](#).

Файл `cp.php` это наш бэкенд, административная часть. Название «`cp`» - аббревиатура Control Panel.

Файл `index.php` – фронтенд сайта.

Рекомендации по структуре и логике при разработке бэкенда (`cp.php`)

Файл состоит из двух логических разделов:

- Раздел `php` скрипта;
- Раздел HTML шаблона с вставками JavaScript кода и переменных `php`.

Раздел `php` скрипта включает в себя:

- Подключение файла `db_include.php`;
- Приём переменных, переданных сценарию методом GET и POST;
- Сценарий удаления раздела
- Формирование меню разделов;
- Обработчик ссылки создания нового раздела
- Сценарий создания нового раздела
- Сценарий подготовки раздела к редактированию
- Сценарий редактирования раздела

Раздел HTML шаблона включает в себя:

- JavaScript код подключения плагина `tinymce`;
- JavaScript код настройки работы плагина `tinymce`;
- Форма, содержащая поле типа `textarea`, поле `input` типа `text` и кнопку типа `submit` для отправки формы;
- Кнопки-ссылки «Удалить раздел» и «Создать раздел»
- Меню разделов

Описание работы сценариев

При запуске файла `sr.php` происходит подключение к базе данных backend.

```
include'db_include.php' ;
```

Потом запускается сценарий формирования меню разделов.

```
//Формирование меню разделов//
$query="SELECT id, razdely from content";
$result=mysql_query($query);
if($result)
{
    $menu='<table width="500" cellpadding="2" cellspacing="0"
border="1" bordercolor="#CCCCCC" align="center">
<tr>
    <th>ID</th>
    <th>Раздел</th>
</tr>
';
    while ($row = mysql_fetch_assoc($result))
    {
        $menu.='<tr>
            <td><a
href="?p=2&id='.$row['id'].'">'.$row['id'].'</a></td>
            <td>'.$row['razdely'].'</td>
        </tr>';
    }
    $menu.='</table>';
}
```

Этот сценарий отправляет запрос к таблице `content` базы данных на выборку записей значений, содержащихся в полях `id` и `razdely`. С помощью

цикла while происходит получение данных, которые обёртываются в табличную форму и присваиваются переменной \$menu. Эта переменная в результате работы сценария содержит полный HTML код меню разделов, упакованных в таблицу со столбцами ID и Раздел. При этом id оформляется в виде ссылки для отправки команды на редактирование разделов.

```
<a href="?p=2&id='.$row['id'].'">'.$row['id'].'</a>
```

Ссылка содержит параметр p=2 и параметр id, равный id раздела, которые передаются методом GET сценарию редактирования разделов.

При первом запуске атрибут action формы не содержит никаких параметров. Чтобы подготовить форму к созданию нового раздела, необходимо сформировать переменную \$action, которая в зависимости от задачи принимает значения:

```
$action='?ad=12'; для создания раздела;
```

```
$action='?ad=22&id='.$id; для редактирования раздела.
```

Для создания раздела необходимые параметры передаются при клике по кнопке «Создать раздел»

```
<a href="?p=1"><button>Создать раздел</button></a>
```

В этом случае запускается сценарий «Обработчик ссылки создания нового раздела». Он формирует переменную \$action для создания раздела.

```
///Обработчик ссылки создания нового раздела///
if($p==1)
{
    $action='?ad=12';
}
```

Эта переменная вставлена в html код формы:

```
<form method="post" action="<?=$action;?>">
```

После того, как будут заполнены поля формы и нажата кнопка submit в работу вступит сценарий создания нового раздела.

Этот сценарий готовит запрос на добавление и отправляет его. В результате в таблице content появляется новая запись.

```
//обработчик создания нового раздела/////
if($ad==12)
{
    $query="INSERT INTO `backend`.`content` (
        `id` ,
        `razdely` ,
        `content`
    )
    VALUES (
        NULL , '$razdely', '$incontent'
    )";

    $res=mysql_query($query);
}
```

```

        if (!$res)
        {
            echo $query.'<br>';
        }
        else
        header('location:?p=1');
    }

```

Для редактирования раздела необходимо в меню раздела кликнуть по ссылке на id раздела. При этом вступает в работу сценарий подготовки раздела к редактированию:

```

//обработчик подготовки раздела к редактированию
if($p==2)
{
    $action='?ad=22&id='.$id;
    ///Формируем запрос выборки по id раздела
    $query="SELECT * from content WHERE id=".$id;
    $res=mysql_query($query);
    if($res)
    {
        $row=mysql_fetch_assoc($res);
        $incontent=$row['content'];
        $id=$row['id'];
        $razdely=$row['razdely'];
    }
    else echo $query.'<br>';
}

```

Данный обработчик формирует переменную \$action для формы редактора \$action='?ad=22&id='.\$id; где сформирован запрос GET, передающий параметры ad=22 и id= id редактируемой записи.

Далее формируется запрос к базе данных на выборку из таблицы content единственной строки, соответствующей переданному в сценарий id.

Сформированный запрос отправляется в базу данных функцией mysql_query(), результат, возвращаемый базой данных записывается в переменную \$res.

Далее проверяется истинность переменной \$res и если запрос выполнен успешно, производится формирование массива \$row с помощью функции mysql_fetch_assoc() в параметр которой передан результат запроса \$res.

Из сформированного ассоциативного массива \$row мы получаем значения полей, используя в качестве ключей имена полей. Значения присваиваем переменным:

```

$row=mysql_fetch_assoc($res);
    $incontent=$row['content'];
    $id=$row['id'];
    $razdely=$row['razdely'];

```

Переменные `$incontent` и `$razdely` подставлены в форму в соответствующие поля в качестве значений по умолчанию:

```
<textarea name="incontent"><?=$incontent;?></textarea>
<input name="razdely" type="text" size="120" value="<?=$razdely;?>">
```

Значения этих переменных отобразятся в полях формы и станут доступны для редактирования.

После редактирования данных и нажатия кнопки Submit форма отправит эти данные в сценарий редактирования раздела:

```
/////обработчик редактирования раздела/////
if($ad==22)
{
    $query="UPDATE `backend`.`content` SET `content` =
'".$incontent."', `razdely` = '".$razdely.'"
    WHERE `content`.`id` = ".$id;
    $result=mysql_query($query);
    if($result)
    {
        header('location:?p=2&id='.$id);
    }
    else echo $query.'  
';
    echo $query.'  
';
}
```

Данный сценарий формирует запрос к базе данных на изменение.

Запрос отправляется в базу данных, результат возвращается в переменную `$result`.

```
$result=mysql_query($query);
```

В случае успешного результата управление передаётся на сценарий подготовки раздела к редактированию с помощью заголовка `header()`:

```
header('location:?p=2&id='.$id);
```

В заголовке передаётся `id` редактируемого раздела. И форма снова готова для редактирования.

Для удаления созданного раздела необходимо кликнуть по кнопке «Удалить раздел» Эта кнопка оформлена в виде ссылки:

```
<a href="?del=1&id=<?=$id;?>"><button>Удалить раздел</button></a>
```

Ссылка методом GET передаёт параметры `del=1` и `id` удаляемого раздела, хранящегося в переменной `$id`.

При этом запускается сценарий удаления раздела:

```
/////Удаление раздела//////////
if($del==1)
{
    $query="DELETE from content where id=".$id;
    $res=mysql_query($query);
    if(!$res) echo $query.'  
';
}
```

```
}
```

Этот сценарий формирует т передаёт запрос к базе данных на удаление записи, которой соответствует переданный id.

Все переменные, значения которых передаются методом GET и POST принимают значения в блоке приёма переменных, переданных сценарию методом GET и POST:

```
/////Приём переменных/////
$p=$_GET['p'];
$ad=$_GET['ad'];
$razdely=$_POST['razdely'];
$incontent=$_POST['incontent'];
$id=$_GET['id'];
$del=$_GET['del'];
$action='';
```

Описание html шаблона

```
<form method="post" action="<?=$action;?>">
  <textarea name="incontent"><?=$incontent;?></textarea>
  <table width="80%" border="0" cellspacing="0" cellpadding="5">
    <tr>
      <td colspan="2"><div align="right">Раздел:</div></td>
      <td><input name="razdely" type="text" size="120"
value="<?=$razdely;?>"></td>
    </tr>
    <tr>
      <td><input type="submit" value="Сохранить"></td>
    </tr>
  </table>

</form>
<table width="200" border="0" cellspacing="0" cellpadding="5">
  <tr>
    <td ><a href="?del=1&id=<?=$id;?>"><button>Удалить
раздел</button></a></td>
    <td><a href="?p=1"><button>Создать раздел</button></a></td>
  </tr>
</table>

<?
  echo $menu;

?>
<div>
  <? echo $incontent; ?>
</div>
```

В разделе body html шаблона размещается код формы, часть элементов которой обернута в таблицу.

В элементы формы встроены php переменные, в которые описанными выше сценариями передаются подставляемые значения.

Ниже формы размещены ссылки-кнопки, обёрнутые в теги `<button></button>` и встроенные для красоты в таблицу.

Ниже встроена php переменная, в которую сценарием формирования меню разделов передаётся html код меню разделов.

Завершает html шаблон демонстрация содержимого переменной `$incontent`, в которой содержится контент раздела.

Прочие рекомендации

При формировании запросов SQL рекомендуется во избежание ошибок синтаксиса использовать конструктор запросов, предоставляемый сервисом phpMyAdmin.