

制用获取表数据接口

conn.php

```
<?php
header("Content-type: text/html; charset=utf-8");
$conn=@mysqli_connect("localhost","root","root","userdb");
if(!$conn){
    die("连接数据库失败:".mysqli_connect_error());
}
mysqli_query($conn,"set names utf8");
class Response{
    public static function json($code,$msg="", $data=array()){
        $res=array(
            'code'=>$code,
            'msg'=>$msg,
            'data'=>$data
        );
        echo json_encode($res);
        exit;
    }
}
```

getBookList.php

```
<?php
include("../conn.php");
$sql="select * from book";
$res=mysqli_query($conn,$sql);
$data=array();
//var_dump($res);
if($res){
    while($row=@mysqli_fetch_assoc($res)){
        //print_r($row);
        $data[]=$row;
    };
}
//mysqli_affected_rows($conn);
if(mysqli_affected_rows($conn)!=-1){
    Response::json(200,'返回数据成功',$data);
}else{
    Response::json(400,'返回数据失败');
}
?>
```

01.使用GET发起不带参数的请求

```

<body>
  <button id="btnGET">发起不带参数的GET请求</button>
  <script>
    $(function () {
      $('#btnGET').on('click', function () {
        $.get('localhost/api/getBookList', function (res) {
          console.log(res)
        })
      })
    })
  </script>
</body>

```

制作删除接口

```

<?php
include("../conn.php");
$id=$_GET["id"];
$sql="delete from book where id={$id}";
$res=mysqli_query($conn,$sql);
$data=array();
//var_dump($res);
if($res){
    while($row=@mysqli_fetch_assoc($res)){
        //print_r($row);
        $data[]=$row;
    };
}
//mysqli_affected_rows($conn);
if(mysqli_affected_rows($conn)!=-1){
    Response::json(200,'删除数据成功',$data);
}else{
    Response::json(400,'删除数据失败');
}
?>

```

02.使用GET发起带参数的请求

```

<body>
  <button id="btnGETINFO">发起单参数的GET请求</button>
  <script>
    $(function () {
      $('#btnGETINFO').on('click', function () {
        $.get('http://localhost/api/getbooks/api/getbooks', { id: 1 }, function (res) {
          console.log(res)
        })
      })
    })
  </script>
</body>

```

制作添加数据

03.使用POST提交数据

```
<body>
  <button id="btnPOST">发起POST请求</button>
  <script>
    $(function () {
      $('#btnPOST').on('click', function () {
        $.post('http://localhost/api/getbooks/api/addbook', { bookname: '水浒传', author: '施耐庵', publisher: '天津图书出版社' }, function (res) {
          console.log(res)
        })
      })
    })
  </script>
</body>
```

04.使用\$.ajax发起GET请求

```
<body>
  <button id="btnGET">发起GET请求</button>
  <script>
    $(function () {
      $('#btnGET').on('click', function () {
        $.ajax({
          type: 'GET',
          url: 'http://localhost/api/getbooks',
          data: {
            id: 1
          },
          success: function (res) {
            console.log(res)
          }
        })
      })
    })
  </script>
</body>
```

05.使用\$.ajax发起POST请求

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script src="./lib/jquery.js"></script>
</head>

<body>
  <button id="btnPOST">发起POST请求</button>

  <script>
    $(function () {
      $('#btnPOST').on('click', function () {
        $.ajax({
          type: 'POST',
          url: 'http://localhost/api/addbook',
          data: {
            bookname: '史记',
            author: '司马迁',
            publisher: '上海图书出版社'
          },
          success: function (res) {
            console.log(res)
          }
        })
      })
    })
  </script>
</body>

</html>

```

制作搜索接口

```

//封装搜索信息
$wherelist = array ();//定义一个封装搜索条件的数组变量
//判断书是否有值,若有则封装此搜索条件
if (!empty($_GET["bookname"])){
  $wherelist []="title like '%".$_GET ['bookname'] .%" ";
}
//判断作者是否有值,若有则封装此搜索条件
if (!empty($_GET["author"] )){
  $wherelist []="keywords like '%".$_GET ['author'] .%" ";
}
//判断出版社是否有值,若有则封装此搜索条件
if (!empty($_GET["publisher"] )){
  $wherelist []="keywords like '%".$_GET ['publisher'] .%" ";
}
//组装搜索条件
if(count($wherelist)>0){
  $where = " where " .implode (" and " , $wherelist);
}

```

```
echo $where;
```

06.图书管理

```
<body style="padding: 15px;">
  <!-- 添加图书的Panel面板 -->
  <div class="panel panel-primary">
    <div class="panel-heading">
      <h3 class="panel-title">添加新图书</h3>
    </div>
    <div class="panel-body form-inline">
      <div class="input-group">
        <div class="input-group-addon">书名</div>
        <input type="text" class="form-control" id="iptBookname" placeholder="请输入书名">
      </div>
      <div class="input-group">
        <div class="input-group-addon">作者</div>
        <input type="text" class="form-control" id="iptAuthor" placeholder="请输入作者">
      </div>
      <div class="input-group">
        <div class="input-group-addon">出版社</div>
        <input type="text" class="form-control" id="iptPublisher" placeholder="请输入出版社">
      </div>
      <button id="btnAdd" class="btn btn-primary">添加</button>
    </div>
  </div>
  <!-- 图书的表格 -->
  <table class="table table-bordered table-hover">
    <thead>
      <tr>
        <th>Id</th>
        <th>书名</th>
        <th>作者</th>
        <th>出版社</th>
        <th>操作</th>
      </tr>
    </thead>
    <tbody id="tb"></tbody>
  </table>

  <script>
    $(function () {
      // 获取图书列表数据
      function getBookList() {
        $.get('localhost/api/getbooks', function (res) {
          if (res.status !== 200) return alert('获取数据失败！')

          var rows = []
          $.each(res.data, function (i, item) {

            rows.push('<tr><td>' + item.id + '</td><td>' + item.bookname + '</td><td>' +
```

```

item.author + '</td><td>' + item.publisher + '</td><td><a href="javascript:;" class="del" data-
id="' + item.id + '">删除</a></td></tr>')
    })
    $('#tb').empty().append(rows.join(''))
  })
}

getBookList()

/* $('#del').on('click', function () {
    console.log('ok')
}) */

// 通过代理的方式为动态添加的元素绑定点击事件
$('tbody').on('click', '.del', function () {
    var id = $(this).attr('data-id')
    $.get('localhost/api/delbook', { id: id }, function (res) {
        if (res.status !== 200) return alert('删除图书失败!')
        getBookList()
    })
})

$('#btnAdd').on('click', function () {
    var bookname = $('#iptBookname').val().trim()
    var author = $('#iptAuthor').val().trim()
    var publisher = $('#iptPublisher').val().trim()
    if (bookname.length <= 0 || author.length <= 0 || publisher.length <= 0) {
        return alert('请填写完整的图书信息!')
    }

    $.post('localhost/api/addbook', { bookname: bookname, author: author, publisher:
publisher }, function (res) {
        if (res.status !== 201) return alert('添加图书失败!')
        getBookList()
        $('#iptBookname').val('')
        $('#iptAuthor').val('')
        $('#iptPublisher').val('')
    })
})
})
</script>

```

机器人

```

<div class="wrap">
  <!-- 头部 Header 区域 -->
  <div class="header">
    <h3>小思同学</h3>
    
  </div>
  <!-- 中间 聊天内容区域 -->
  <div class="main">

```

```

<ul class="talk_list" style="top: 0px;" id="talk_list">
  <li class="left_word">
     <span>嗨，最近想我没有？</span>
  </li>
  <!-- <li class="right_word">
     <span>你好哦</span>
  </li> -->
</ul>
<div class="drag_bar" style="display: none;">
  <div class="drager ui-draggable ui-draggable-handle" style="display: none; height:
412.628px;"></div>
</div>
</div>
<!-- 底部 消息编辑区域 -->
<div class="footer">
  
  <input type="text" placeholder="说的什么吧..." class="input_txt" id="ipt" />
  <input type="button" value="发 送" class="input_sub" id="btnSend" />
</div>
</div>

<!-- 注意：只要为 audio 指定了新的 src 属性，而且指定了 autoplay，那么，语音就会自动播放 -->
<audio src="" id="voice" autoplay style="display: none;"></audio>

<script type="text/javascript" src="js/scroll.js"></script>
<script>
$(function() {
  // 初始化右侧滚动条
  // 这个方法定义在scroll.js中
  resetui()

  // 为发送按钮绑定鼠标点击事件
  $('#btnSend').on('click', function() {
    var text = $('#ipt').val().trim()
    if (text.length <= 0) {
      return $('#ipt').val('')
    }
    // 如果用户输入了聊天内容，则将聊天内容追加到页面上显示
    $('#talk_list').append('<li class="right_word"> <span>' + text
+ '</span></li>')
    $('#ipt').val('')
    // 重置滚动条的位置
    resetui()
    // 发起请求，获取聊天内容
    getMsg(text)
  })

  // 获取聊天机器人发送回来的消息
  function getMsg(text) {
    $.ajax({
      method: 'GET',

      url: ' http://www.liulongbin.top:3006/api/robot',

```

```

data: {
  spoken: text
},
success: function(res) {
  // console.log(res)
  if (res.message === 'success') {
    // 接收聊天消息
    var msg = res.data.info.text
    $('#talk_list').append('<li class="left_word"> <span>' +
msg + '</span></li>')
    // 重置滚动条的位置
    resetui()
    // 调用 getVoice 函数, 把文本转化为语音
    getVoice(msg)
  }
}
})
}

// 把文字转化为语音进行播放
function getVoice(text) {
$.ajax({
  method: 'GET',
  url: 'http://www.liulongbin.top:3006/api/synthesize',
  data: {
    text: text
  },
  success: function(res) {
    // console.log(res)
    if (res.status === 200) {
      // 播放语音
      $('#voice').attr('src', res.voiceUrl)
    }
  }
})
}

// 为文本框绑定 keyup 事件
$('#ipt').on('keyup', function(e) {
  // console.log(e.keyCode)
  if (e.keyCode === 13) {
    // console.log('用户弹起了回车键')
    $('#btnSend').click()
  }
})
})
</script>

```


