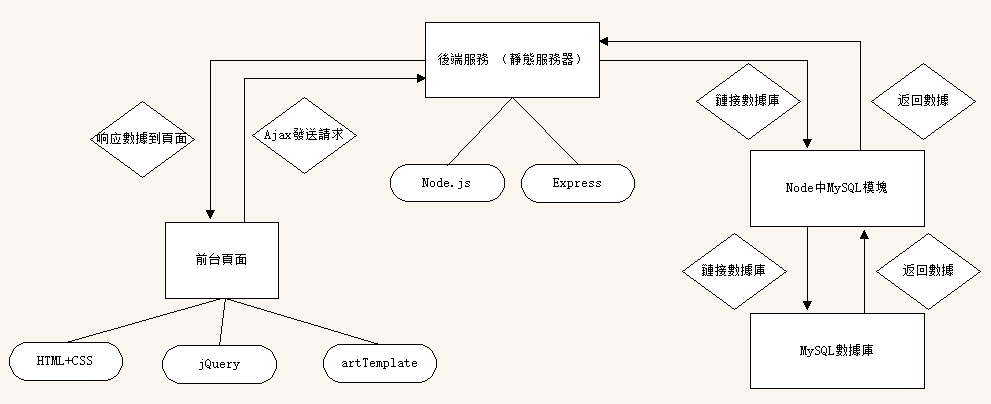
本階段目標：

1. 使用Node/Express搭建靜態服務器，提供後端服務
2. 知道Node.js不同模塊的作用
3. 使用MySQL數據庫存儲數據
4. 使用artTemplate模板引擎，高效、快速渲染數據
5. 使用Ajax請求後端接口，對數據進行增刪改查操作
6. Git項目版本管理工具
7. GitHub項目代碼託管平台

技術點之間的聯繫



Ajax介紹

## 1 客戶端與服務器

在上網過程中，存放和對外提供資源的電腦，叫做服務器。

服務器本質是一台電腦，只不過其性能要比個人電腦高很多。

在上網過程中，負責獲取和消費資源的電腦，叫做客戶端。個人電腦中，可以通過安裝瀏覽器的形式，訪問服務器對外提供的各種資源。

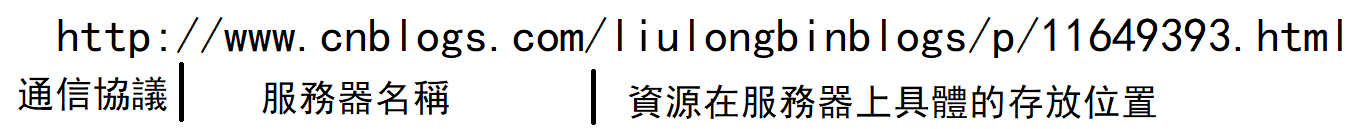
## 2 URL地址

URL英文全稱是Uniform Resource Location，中文叫統一資源定位符，用於標識互聯網上每個資源的唯一存放位置。

瀏覽器只有通過URL地址，才能正確定位資源的存放位置，從而成功訪問到對應的資源。

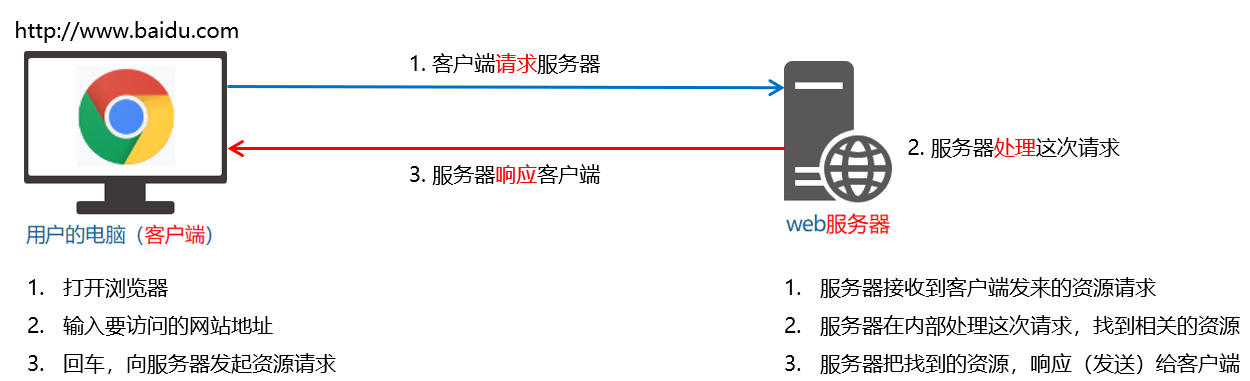
URL地址組成部分：

1. 客戶端與服務器之間的通信協議
2. 存有該資源的服務器名稱
3. 資源在服務器上具體的存放位置



## 3 分析網頁的打開過程

### （1）圖解這個過程



注意：

① 客戶端與服務器之間的通信過程，分為請求-處理-響應三個步驟。

② 網頁中的每一個資源，都是通過請求-處理-響應的方式從服務器獲取回來的。

### （2）用瀏覽器的開發者工具分析通信過程

① 打開Chrome瀏覽器

② 按Ctrl + Shift + I 或者F12打開Chrome的開發者工具

③ 切換到Network面板

④ 選中Doc標籤頁

⑤ 刷新頁面，分析客戶端與服務器的通信過程

## 4 服務器提供的資源

### （1）務器對外提供了哪些資源

除了文字內容，image圖片，audio音頻，video視頻以外，網頁中的數據也是服務器對外提供的一種資源，例如股票數據、個行業排行榜等。

數據是網頁的靈魂。

HTML是網頁的骨架。

CSS是網頁的顏值

JavaScript是網頁的行為。

骨架、顏值、行為都為數據服務。

### （2）網頁中如何請求數據？

如果要在網頁中請求服務器上的數據資源，需要用到XMLHttpRequest對象。

XMLHttpRequest（簡稱xhr）是瀏覽器提供的js成員，通過它，可以請求服務器上的數據資源。

最簡單的用法 var xhrObj = new XMLHttpRequest()

### （3）資源的請求方式

客戶端請求服務器時，請求的方式有很多種，最常見的兩種請求方式分別為get和post請求。

* get請求通常用於獲取服務端資源（向服務器要資源）

如根據URL地址，從服務器獲取HTML文件、CSS文件、JS文件、圖片文件、數據資源等。

* post請求通常用於向服務器提交數據（往服務器發送資源）

如登錄時向服務器提交的登錄信息、註冊時向服務器提交的註冊信息、添加用戶時向服務器提交的用戶信息等各種數據的提交操作。

## 5 了解Ajax

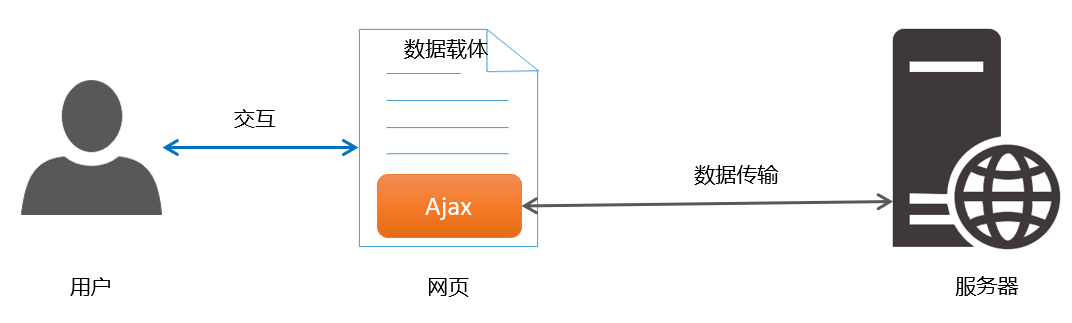
### （1）什麼是Ajax

Ajax全稱是Asynchronous JavaScript And XML （異步JavaScript和XML）。

通俗的理解：在網頁中利用XMLHttpRequest對象和服務器進行數據交互的方式，就是Ajax。

### （2）為什麼要學Ajax

之前所學的技術，只能把網頁做得更美觀漂亮，或者添加一些動畫效果。而Ajax可以讓我們輕鬆實現網頁與服務器之間的數據交互。

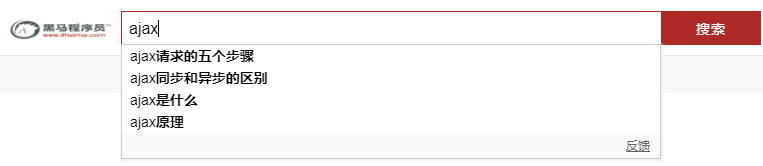


### （3）Ajax的典型應用場景

用戶名檢測：註冊用戶時，通過Ajax的形式，動態檢測用戶名是否被佔用。



搜索提示：當輸入搜索關鍵字時，通過Ajax的形式，動態加載搜索提示列表。



數據分頁顯示：當點擊頁碼值的時候，通過Ajax的形式，根據頁碼值動態刷新表格的數據。



數據的增刪改查：數據的添加、刪除、修改、查詢操作，都需要通過Ajax的形式來實現數據的交互。



## 6 jQuery中的Ajax

### （1）概述

瀏覽器中提供的XMLHttpRequest用法比較複雜，所以jQuery對XMLHttpRequest進行了封裝，提供了一系列Ajax相關的函數，極大地降低了Ajax的使用難度。

jQuery中發起Ajax請求最常用的三個方法如下：

$.get() 向服務器拿數據

$.post() 向服務器提交數據

$.ajax() 既可以向服務器拿數據又可以向服務器提交數據

### （2）$.get()

jQuery中$.get()函數的功能單一，專門用來發起get請求，從而將服務器上的資源請求到客戶端來進行使用。

語法：

$.get(url, [data], [callback])

第1個參數是必填，後面兩個參數被中括號[ ]括起來，是選填的。

|  |  |  |  |
| --- | --- | --- | --- |
| 參數名 | 參數類型 | 是否必填 | 說明 |
| url | string | 是 | 要請求的資源地址 |
| data | object | 否 | 請求資源期間要攜帶的參數 |
| callback | function | 否 | 請求成功時的回調函數 |

發起不帶參數的請求

    <script>

      $.get("http://www.liulongbin.top:3006/api/getbooks", function (*res*) {

        console.log(res); //res是服務器返回的數據

      });

    </script>



發起帶參數的請求

    <script>

      $.get(

        "http://www.liulongbin.top:3006/api/getbooks",

        { id: 1 },

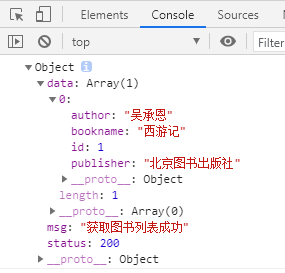
        function (*res*) {

          console.log(res); //res是服務器返回的數據

        }

      );

    </script>



### （3）$.post()

jQuery中$.post()函數的功能單一，專門用來發起post請求，從而向服務器提交數據。

$.post()函數的語法如下：

$.post(url, [data], [callback])

其中三個參數各自代表的含義如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 參數名 | 參數類型 | 是否必填 | 說明 |
| url | string | 是 | 要提交數據的地址 |
| data | object | 否 | 要提交的數據 |
| callback | function | 否 | 數據提交成功時的回調函數 |

    <script>

      $.post(

        "http://www.liulongbin.top:3006/api/addbook", //請求的URL地址

        {

          bookname: "星之繼承者",

          author: "詹姆斯·P·霍根",

          publisher: "新星出版社",

        }, //提交的數據

        function (*res*) {

          //回調函數

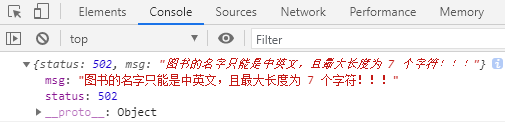
          console.log(res); //res是服務器返回的數據

        }

      );

    </script>







### （4）$.ajax()

相比于$.get()和$.post()，$.ajax()是一個功能比較綜合的函數，它允許我們對Ajax請求進行更詳細的配置。

$.ajax()函數的基本語法如下：

$.ajax({

type: ‘’, //請求的方式，例如GET或者POST （全部大寫或者全部小寫都正確）

url: ‘’ , //請求的URL地址

data: { }, //請求要攜帶的數據

success: function(res) { } //請求成功之後的回調函數

})

    <script>

      $.ajax({

        type: "GET", //請求的方式，例如GET或者POST

        url: "http://www.liulongbin.top:3006/api/getbooks", //請求的URL地址

        data: { id: 1 }, //請求要攜帶的數據。如果要返回全部數據，這個data可省略

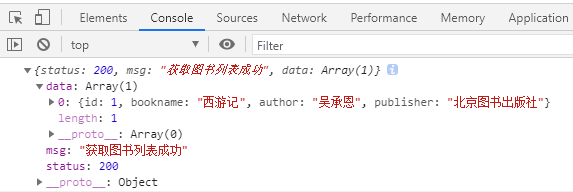
        success: function (*res*) { //請求成功之後的回調函數

          console.log(res); //res是服務器返回的數據

        },

      });

    </script>



    <script>

      $.ajax({

        type: "POST", //請求的方式，例如GET或者POST

        url: "http://www.liulongbin.top:3006/api/addbook", //請求的URL地址

        data: {

          //要提交給服務器的數據

          bookname: "星之繼承者",

          author: "詹姆斯·P·霍根",

          publisher: "新星出版社",

        },

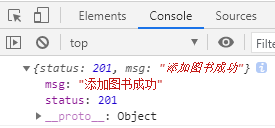
        success: function (*res*) {

          console.log(res); //res是服務器返回的數據

        }, //請求成功之後的回調函數

      });

    </script>



## 7 接口

### （1）接口的概念

使用Ajax請求數據時，被請求的URL地址，就叫做數據接口（簡稱接口）。同時，每個接口必須有請求方式（GET或者POST）。

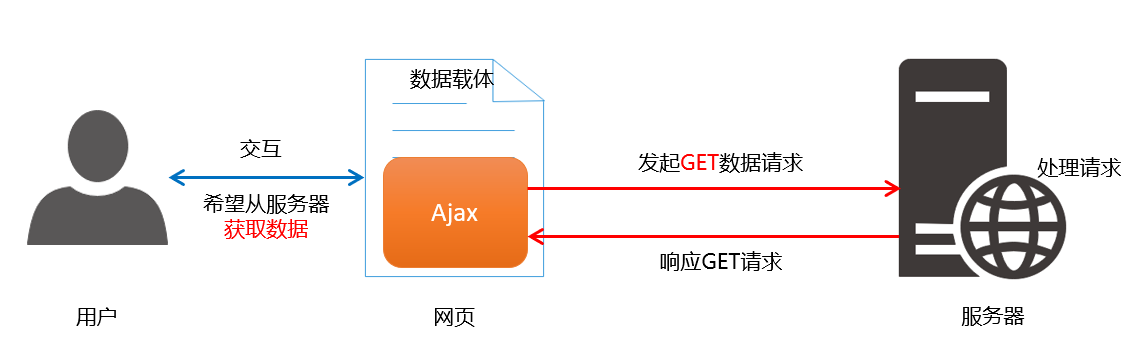
例如：

<http://www.liulongbin.top:3006/api/getbooks> 獲取圖書列表的接口（GET請求）

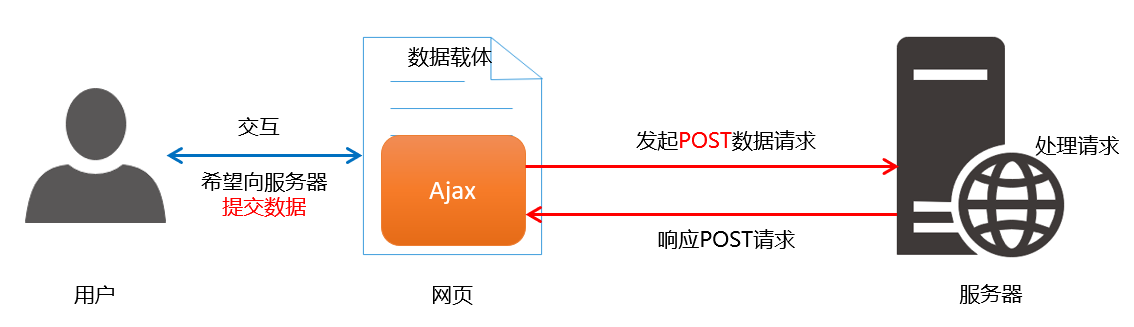
<http://www.liulongbin.top:3006/api/addbook> 添加圖書的接口（POST請求）

### （2）分析接口的請求過程

通過GET方式請求接口的過程



通過POST方式請求接口的過程



### （3）接口測試工具

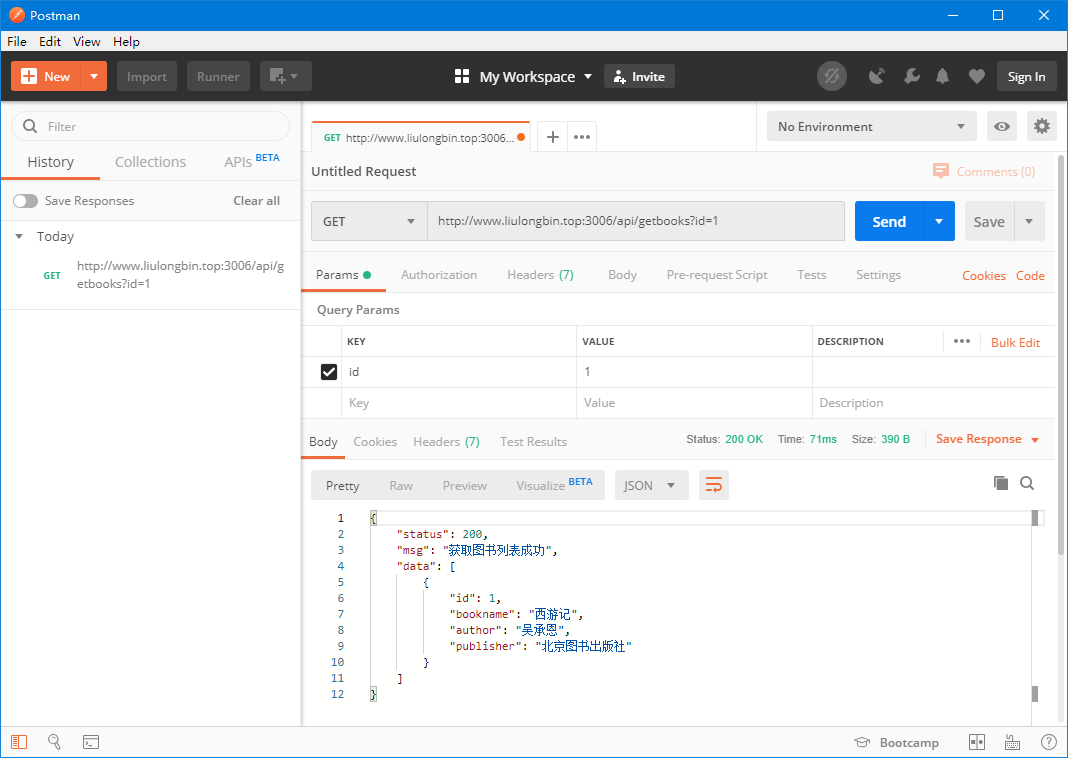
為了驗證接口能否被正常訪問，我們常常需要使用接口測試工具來對數據接口進行檢測。

好處：接口測試工具讓我們在不寫任何代碼的情況下，對接口進行調用和測試。

PostMan是個非常優秀的接口測試工具

訪問PostMan的官方下載網址：<https://www.getpostman.com/downloads/> ，下載所需的安裝程序後，直接安裝即可。

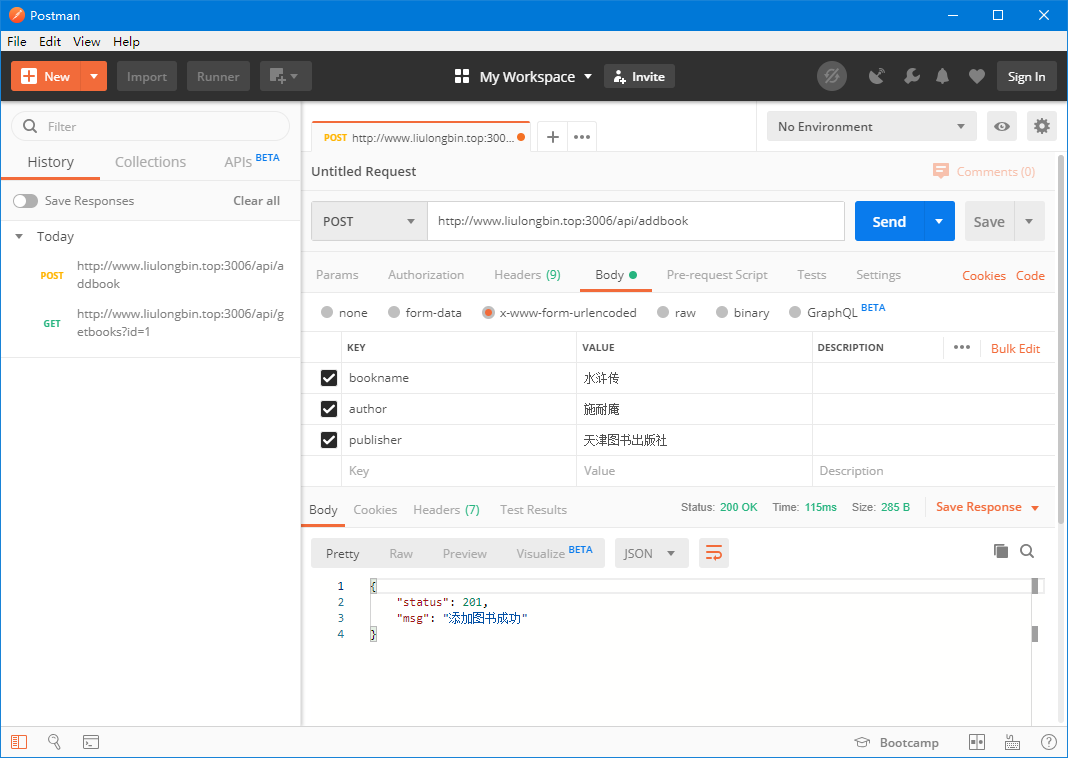
使用PostMan測試GET接口



步驟：

1. 選擇請求方式GET
2. 填寫URL地址
3. 填寫參數
4. 點擊Send按鈕發起GET請求
5. 查看服務器響應的結果

使用PostMan測試POST接口



1. 選擇請求方式POST
2. 填寫URL地址
3. 選擇Body面板，勾選數據格式x-www-form-urlencoded
4. 填寫要發送到服務器的數據
5. 點擊send按鈕發起post請求
6. 查看服務器響應的結果

### （4）接口文檔

接口文檔，即接口說明文檔，它是我們調用接口的依據。好的接口文檔包含了對接口URL、參數以及輸出內容的說明。我們參照接口文檔就能方便地知道接口的作用，以及如何調用接口。

接口文檔的組成部分：

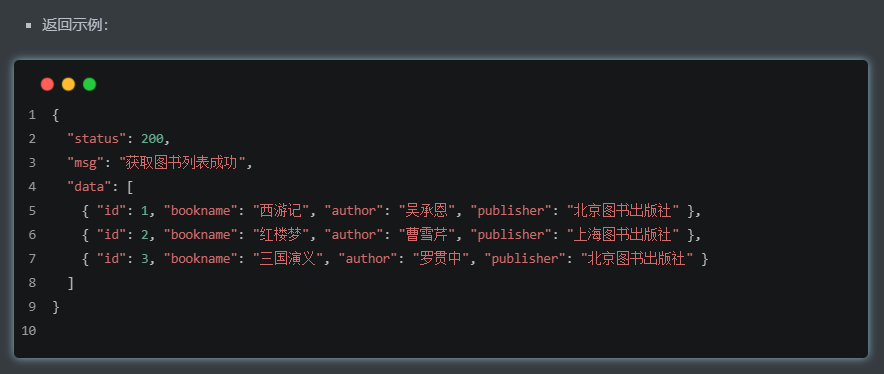
接口文檔可以包含很多信息，也可以按需進行精簡。一個合格的接口文檔，應該包含以下6項內容：

1. 接口名稱：用來標識各個接口的簡單說明，如登錄接口，獲取圖書列表接口等。
2. 接口URL：接口的調用地址。
3. 調用方式：如GET，POST。
4. 參數格式：接口需要傳遞的參數，每個參數必須包含參數名稱、參數類型、是否必選、參數說明這4項內容。
5. 響應格式：接口的返回值的詳細描述，一般包含數據名稱、數據類型、說明3項內容。
6. 返回示例（可選）：通過對象的形式，例舉服務器返回數據的結構。

接口文檔示例：







## 8 案例-圖書管理

### （1）渲染UI結構



### （2）案例用到的庫個插件

用到的CSS庫：bootstrap.css

用到的JavaScript庫：jquery.js

用到的vs code 插件：Bootstrap 3 Snippets 或者 Bootstrap 4 Snippets

## 9 案例-聊天機器人

分析

實現步驟：

1. 梳理案例的代碼結構
2. 將用戶輸入的內容渲染到聊天窗口
3. 發起請求獲取聊天消息
4. 將機器人的聊天內容轉換為原因
5. 通過<audio>播放語音
6. 使用回車鍵發送消息

梳理案例的代碼結構：

1. 梳理頁面的UI佈局
2. 將業務代碼抽離到chat.js中
3. 了解resetui()函數的作用：向聊天區域發送了新內容後，滾動條可以被自動拉到頁面最底部。