

Jiayuan Yan 1904780

by Jiayuan Yan

FILE	17866_JIAYUAN_YAN_JIAYUAN_YAN_1904780_242847_893779515.DOCX (30.17K)		
TIME SUBMITTED	17-APR-2020 11:11PM (UTC+0100)	WORD COUNT	3008
SUBMISSION ID	124380154	CHARACTER COUNT	15535

L7 - Web Development (ACCA7004)

Assignment

Guide teacher: Mike Dacey

<Jiayuan Yan>

<1904780>

Abstract:

This paper is mainly divided into four parts: 1. Discuss the term server's code, and explains the MVC design method. 2. Compare the client-side and server-side script. 3. The study Nodejs framework, and critically evaluate the strengths and weaknesses of using it. 4. Discuss the significance of AJAX. In the second and the fourth part two I will through the upload code form to their corresponding example.

Keywords: MVC design pattern, Nodejs framework, AJAX

Contents

1. “server's code”and MVC design method.....	4
1.“server's code”	4
2. the strengths of the (MVC) design method.....	5
2. client-side and server-side scripting.....	7
1. Client script:.....	7
2.Server script:.....	8
3.Differences between server and client scripts:.....	9
3. the benefits and drawbacks of using Nodejs framework	10
4. the importance of AJAX Error! Bookmark not defined.	
5.References	19

1. “server-side code”and MVC design pattern

1.“server-side code”

First, I introduce the following server-side and client-side concepts:

The role of the client in web development is to send requests, or requirements, to the server, which then processes them. The server side is responsible for receiving the request sent by the client side, and then processing it, and then feedback the existence of the object after processing.

For example, when we visit the web site, use the browser or application request, and we retrieve the site (such as sina, Google) they make corresponding feedback, finally a new page to us, this is a server side and client a process of interaction, among them, the browser, application is the client, Google sites is the server.

Server-side programming, on the other hand, is writing code that runs on the server side of a dynamic web site. Server-side code is code that retrieves database information, USES database memory, or responds to user needs.

Next, I will introduce the interaction process between the server and the client:

When you click a link, submit a form, or perform a search on a web page, the browser sends an HTTP request to the server.

Request contains a URL that specifies the location of the site and the resource to be (modified). This may be followed by a URL parameter to

transmit the information you want to GET. This often contains the following information:

One method: what do you want to do with the resource, GET GET, POST create, HEAD, PUT update, etc.

Subject: to the resources of the POST, cookies contain session data related to customers.

After the server analyzes the request, it extracts the required resources from the file or database, and then returns them in response.

The response includes a HEADER: there are: the response status code 200 OK or 404 ERROE, and the format of the response time, the main content and time, using the kind of character set to coding, information about the server, etc.

Main body: returns the HTML.

2. the benefits of the Model View Controller (MVC) design pattern.

The concept of the MVC pattern :

MVC design pattern, which divides the software development system into model, view and controller, is a unique software structure pattern in software development engineering^[5].

View:

There is no actual processing going on in the view, whether the data is stored online or a list of employees, but as a view it is simply a way to output the data and allow the user to manipulate it.

Model:

Models represent enterprise data and business rules. Of the three parts of MVC, the model has the most processing tasks. For example, it might use component objects like EJBs and ColdFusionComponents to work with databases. The data returned by the model is neutral, meaning that

the model is not related to the format of the data, so a model can provide data for multiple views. Because the code applied to the model can be reused by multiple views only once, this reduces code duplication.

Controller:

When a user clicks a connection in a web page and sends a request, the controller does not give any feedback or perform any processing, it just receives the request from the user, decides which model to call for processing, and then decides which view to use to view the returned data.

Based on the concept of MVC pattern, it has the following advantages:

1. Low coupling
2. High reusability and applicability
3. Low life cycle cost
4. Deploy quickly
5. Maintainability
6. Conducive to software engineering management^[8].

2. client and server scripting

First, let me introduce the client-side and server-side scripts:

1. Client script:

Client scripts, such as the browser, are responsible for interpreting and executing HTML documents and scripts fed back from the web server, and they are also responsible for keeping the page up to date, as well as processing the scripts, all on the client side^[3].

It mainly includes the following two aspects:

1. `<SCRIPT type=text/javascript> </SCRIPT>`(It is used to define the code in it)
2. `<script type="text/javascript"src="***.js"></script>` (Defined in an external file, and then referenced with this expression, the call is typically handled by an event)

Client-side scripts have the following characteristics:

1. Client scripts can only be interpreted in the client browser.
2. As you can generally see, the source code is public to the users.
3. If you think of the browser as an application, all of its operations are performed by client-side scripts, and it also controls the user's interaction with the browser.
4. When the client script is executed, the web page cannot interact with the server, so the client script cannot connect to the database to update the data.
5. Client-side scripts often pop up a dialog box in the form of "alert()", which is different from server-side scripts.

Advantages: it not only improves the speed of instant feedback from the web page, but also reduces the load on the server.

Weakness: differences in the presentation of web pages may occur due to browser differences.

Development language environment: support JavaScript, JScript and VBScript languages.

2.Server script:

As the name implies, all of its scripts are processed in the server side, after the user sent HTTP request through the browser, the web server began to run the script, and then the results of the script run and web page HTML sent to the client browser, this is the server side script.

Server scripts are generally "<%%>" surrounded by code, or there is , when an HTML control such as runat="Server", is a control that can be run on the Server side, that is, the web control, so is also equivalent to the scriptrun on the Server side^[4].

It has the following characteristics:

1. Server-side scripts are only used to generate web code (including HTML, CSS, JS, etc.).
2. It is just a result generated page of execution, and the page sent to the client does not contain the code itself.
3. Response.write (" "), is no dialog box pop-up.

Advantage: reduce the difference of the result that the browser brings, improve the stability of the page.

Weakness: increased server load and slow client response.

3.Differences between server and client scripts:

1. Server scripts are generally closed to the public and handled by the web server itself, whereas client-side scripts are relatively open to users.

2. The server-side control can only call the server-side script. If the client-side script is forced to be called, there will inevitably be an error and the compilation error cannot be found because of the different compilation methods. Client-side controls, however, have free access to client-side and server-side scripts because everything on the page is sent to the client.

3. Server-side scripts are used to generate web code, while client-side scripts are mainly used to control the interaction between users and clients.

Note: I will show some examples in "Code Submission" to illustrate their respective features.

3.the benefits and drawbacks of using node.js framework

1.What is the Nodejs framework

Nodejs is a development platform that USES event-driven and non-blocking I/O models to make JavaScript a scripting language on a par with server-side languages like PHP and Python, essentially a process of encapsulating Chrome V8.

In short, Nodejs has the following characteristics:

1. It USES the JavaScript scripting language and provides the environment for JavaScript to run.
2. It explains the code based on Chrome V8.
3. It adopts a non-blocking, event-driven I/O model, and has lightweight, extensible properties. At the same time, it is also suitable for real-time interactive data applications.

Application scenario of Nodejs: Web development: Express + EJS + Mongoose/MySQL, REST development: Restify, Front-end package management platform: bower.js, Browser environment tool: browserify, Command line programming tool: Commander, Client application tool: node-webvit, etc^[6].

1. the benefits and drawbacks of using node.js framework

Benefits:Nodejs can solve the following two problems:

1. Node.js can solve the problem of concurrent connections: instead of generating a new OS thread (and allocating some matching memory) for each connection, Node.js can emit (emit) an Event running in the Node.js engine process and place it in the Event queue by changing the way it connects to the server.

2. Node.js can solve the problem of I/O blocking: Node.js will create a thread to execute the I/O event, and then the main thread will continue to execute. Therefore, the action of taking profile triggers an I/O event, and the action of taking timeline will be executed immediately. After their I/O operation is completed, an event is emitted, profile and timeline, and the event agent receives and continues to execute the following logic, which is the characteristic of Node.js non-blocking I/O.

So the biggest advantage of Node.js is 1. High concurrency 2. Suitable for I/O intensive applications.

Drawbacks:

1. Node.js is not suitable for CPU-intensive applications because of its single-threaded feature. When it runs for a long time, it will be unable to release CPU time slices and launch subsequent I/O after calculation.

2. Node.js can only support single-core CPUs, so it cannot make full use of the CPU, which is its limitation in terms of CPU utilization.

3. Once a part of Node.js code crashes, the entire system will be affected or even crash^[7].

4. The significance of AJAX

1. what is AJAX:

AJAX has the ability to interact between the front and back end, which is the tool that our client sends messages to the server and the tool that receives responses. It is not a new programming language, but a new way to use existing standards. AJAX is a technique for exchanging data with a server and updating portions of a web page (without reloading the entire page). AJAX also has the capability of a default asynchronous execution mechanism, which is divided into synchronous and asynchronous.

At the same time, AJAX is a combination of several existing technologies, specifically reflected in:

1. Use CSS and XHTML for presentation.
2. Use the DOM model for interactive and dynamic display.
3. Use XMLHttpRequest to communicate asynchronously with the server.
4. Bind and invoke using javascript.

At the heart of AJAX is the XMLHttpRequest object.

Different browsers create XMLHttpRequest objects in different ways.

Such as: Internet explorer USES ActiveXObject, while other browsers use a built-in JavaScript object called XMLHttpRequest

2. The operating principles and programming steps of AJAX

Ajax works by adding an intermediate layer (an Ajax engine) between the user and the server, making the user's actions and the server's responses asynchronous. Not all user requests are submitted to the server. Data validation and data processing, for example, are left to the Ajax engine, which submits requests to the server only when it is certain that new data needs to be read from the server.

To understand this process and principle, I must have some understanding of XMLHttpRequest.

XMLHttpRequest is the core mechanism of ajax, which was first introduced in IE5. Is a technique that supports asynchronous requests. Simply put, javascript can make requests and process responses to the server in a timely manner, rather than blocking the user and resulting in no refresh.

The following are three common properties of the XMLHttpRequest object

1. The onreadystatechange property

The onreadystatechange property holds the function that handles the server response.

The following code defines an empty function to set the onreadystatechange property:

```
xmlHttp.onreadystatechange = function() {  
    //We need to write some code here  
}
```

2. The readyState attribute

The readyState property holds the status information for the server response. The onreadystatechange function is executed whenever the readyState changes.

The following table shows the possible values of the readyState property:

state	describe
0	Request not initialized (before calling open())
1	The request is made (before calling send())
2	The request has been sent (here you can usually get the content header from the response)
3	Request processing (some data is usually available in the response, but the server has not completed the response)
4	The request is complete (you can access the server response and use it)

So we add an If statement to the onreadystatechange function to test whether our response is complete (meaning data is available) :

```
xmlHttp.onreadystatechange = function() {  
    if (xmlHttp.readyState == 4) {  
        //Get the data from the response of the server  
    }  
}
```

3. The responseText property

The data returned by the server can be retrieved through the `responseText` property.

In our code, we will set the value of the time text box to equal to `responseText`:

```
xmlHttpRequest.onreadystatechange = function() {  
    if (xmlHttpRequest.readyState == 4) {  
        document.myForm.time.value = xmlHttpRequest.responseText;  
    }  
}
```

Two common methods of `xmlHttpRequest`:

1. The `open ()` method

`Open ()` takes three arguments. The first parameter defines the method used to send the request, the second parameter specifies the URL of the server-side script, and the third parameter specifies that the request should be processed asynchronously.

```
xmlHttpRequest.open("GET","test.php",true);
```

2. The `send ()` method

The `send()` method sends the request to the server. If we assume that HTML files and PHP files are in the same directory, the code looks like this:

```
xmlHttpRequest.send(null);
```

I just briefly introduced the Ajax programming steps here:

1. Create an XMLHttpRequest object.
2. Set the request mode.
3. Call the callback function.
4. Send the request.

Note: I'll detail the programming steps for using AJAX in the [uploaded code](#).

3. the importance of AJAX:

Advantages:

1. No update data

The great advantage of Ajax is the ability to maintain communication with the server without refreshing the entire page

2. Asynchronous communication with the server

Communicate with the server asynchronously without interrupting the user

3. Load balancing between front end and back end

Move some back-end work to the front end, reducing the burden on servers and bandwidth

4. Specification based is widely supported

You don't need to download a browser plug-in or applet, but you do need the customer's permission for JavaScript to execute on the browser.

5. Separate the interface from the application

Ajax separates the interface from the application, that is, the data from the presentation

Disadvantages:

1. Ajax kills off the Back and History functionality, which destroys the browser mechanism

With dynamic page updates, the user cannot return to the previous page state because the browser can only remember static pages in the history

2. Security issues

AJAX technology not only brings good user experience to users, but also brings new security threats to IT enterprises. AJAX technology is like establishing a direct channel for enterprise data. This allows developers to inadvertently expose more data and server logic than ever before.

3. Weak support for search engines

4. Break the exception handling mechanism of the program

5. Violating the original purpose of URL and resource location

6. Client side hypertrophy, too much client code resulting in development costs^[2]

Conclusion

we have been exposed to or applied to ajax in our daily development. When it comes to the significance of ajax technology, we focus most on improving the user experience. If we combine the future of computers and the Internet, we see that ajax technology represents this trend in some ways. Computer may abandon bulky hard drive in the future, but directly from the Internet to get data and services, if so, which still has a lot of problems need to solve, the main problem is the Internet connection is not stable, and AJAX may handle this problem, it's just between the server and the client ACTS as a buffer, let users mistook service without interruption^[1]. To be precise, ajax doesn't make downloading data from the server faster, it just makes the wait less frustrating. But that was enough to make a huge impact and shock, and it actually made a huge impact on desktop software. So, in the future, maybe this technology will have a big impact on the typical desktop software.

5. References

- [1]Sophocles, *Ajax*. Bristol: Classical Press, 1981.
- [2]Sophocles, *Ajax*. Warminster: Aris & Phillips, 1998.
- [3]S. Bernard 和 S. Manuel, 《An interpreter for server-side hop》, *ACM SIGPLAN Notices*, 卷 47, 期 2, 页 1 - 12, 2012.
- [4]K. Kalata, *Internet programming with VBScript anf JavaScript*. Boston, Mass. ; United Kingdom: Course Technology, 2001.
- [5]M. Anzures-Garcia, L. A. Sanchez-Galvez, M. J. Hornos 和 P. Paderewski-Rodriguez, 《MVC Design Pattern Based-Development of Groupware》, 2016, 页 71 - 80.
- [6]Mihailo Vasiljević, Aleksandar Manasijević, Aleksandar Kupusinac, Ćamil Sukić 和 Dragan Ivetić, 《One Solution of Component Based Development in NodeJS for Modularization of gRPC Services and Rapid Prototyping》, *SAR Journal*, 卷 2, 期 4, 页 181 - 185, 2019.
- [7]U. Şimşek, E. Kärle 和 D. Fensel, 《RocketRML - A NodeJS implementation of a use-case specific RML mapper》, 20190312, 见于: 4 月 18, 2020.
- [8]Chunna Zhang 和 Wenxuan Shen, 《The studies on integrated development approach based on the architecture and MVC design pattern》, 2012, 页 578 - 580.
- [9]E. Bronson, 《The Myth of Resilience: Ajax Undressed》, *Mosaic: a Journal for the Interdisciplinary Study of Literature*, 卷 52, 期 4, 页 109 - 125, 2019.
- [10]Y. Wan-Ju, 《以前端瀏覽器為中心之雲端運算服務 模型研究》, *Journal of Library & Information Science*, 卷 41, 期 2, 页 031 - 047, 2015, doi: [10.6245/JLIS.2015.412/655](https://doi.org/10.6245/JLIS.2015.412/655).

FINAL GRADE

60/100

GENERAL COMMENTS

Instructor

Server-side/MVC (25) : 10

Client/Server Scripting (25): 15

Node.js (25): 17

AJAX (25): 18

Feedback:

=====

MVC discussion was very rushed and the benefits needed to be described in much more detail

Script discussion highlighted the main differences between server and client side scripting. The example client side script was short and simple, while the server-side script was confusing since one example was a C networking program??

Node.js section does describe the main points but it should have provided more detail. Likewise the AJAX discussion which should have provided more evaluation of the applications of AJAX.

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19
