A Research on Dynamic Forms Verification Technology Based on PHP + Ajax

Shengju Yang
Gansu Institute of Science & Technology Information   Lanzhou, China
e-mail: ysj@gsinfo.net.cn

***Abstract*—It is necessary to verify the input data to ensure the safety of high-quality software applications. The traditional WEB forms Verification under B/S model is characterized by the problems of restrictions by HTTP, refreshment of entire pages, long-time waiting of users, etc. In order to solve these problems, the paper proposes the Dynamic Forms Verification Technology Based on PHP + Ajax in order to establish asynchronous no-refreshment, fast response and more user-friendly forms verification technology. The results show that this method can greatly improve the response speed of WEB interface and provide users with a more natural browsing experience.**

Keywords-dynamic forms;PHP;ajax;verification

I. Introduction

Under the application environment of WEB, authentication is especially sensitive part. This is because there are multiple potential risks in the illegal data to destroy the application functions, and also they can even make the databases in the most sensitive application areas collapsed. The authentication on the imported data is one of the essential guarantees for the application of the high-quality security software.

The authentication on the imported data refers to checking whether the data imported by the users conforms to the rules which have been previously defined or not. All of these rules are established in accordance with the applied business standards. For example, the "09/9/22" is thought to be invalid if the formal date format is required to be "YYYY-MM-DD". This also goes for the E-mail address and phone numbers, which are necessary to be checked according to an effective format.

II. Web Forms Authentication

The WEB application systems, which are based on the browser/server (B/S) structure, make use of the forms to process the data entry in most situations. For this reason, the WEB forms authentication is one of the most important measures to guarantee the effectiveness of the data entry. Generally speaking, there are three methods of authenticating the WEB forms authentication, which are client-side authentication, server-side authentication, as well as client-side and server-side combined authentication respectively. First of all, the client-side authentication first implements the end authentication through JavaScript, Jscript and VBScript before submitting the forms to server, for the purpose of preventing the users from submitting the data to the server owing to the careless filling errors. This process generally includes the required-field authentication, the field-length authentication, the data authentication, and the authentication meeting a certain rule (usually a regular expression is used). For example, in the literatures [1]-[2], a general JavaScript client-side form authentication plan is put forward in the literature [3]. Second, the server-side authentication first takes advantage of the dynamic languages such as PHP, ASP and JSP to implement the validity of the authentication data in server side after submitting the forms to server, and then executes other operations. The contents in server-side authentication are similar to that in client-side authentication, such as the references [4]-[6]. At last, in the client-side and server-side combined authentication, the client-side authentication is a good supplement to the server-side authentication, while the server-side authentication is essential, such as the reference [7].

However, the above three WEB forms authentication technologies has the following problems [8]. First of all, the server-side authentication is restricted by HTTP, because HTTP is a protocol without a state. Second, the client authentication can't acquire the data by calling the backend database to authenticate the validation of the form data. However, the client-side authentication can be skipped. As a result, the forms can be submitted without an authentication as long as the active scripts are disabled. Finally, in the submission of a form webpage, the users need to wait for the loading of the entire page.

In accordance with the above-mentioned problems, the authors realize a method of applying the AJAX to authenticate the forms under the environment of PHP, and therefore the forms authentication technology which is more friendly and quicker in responses is established in this paper.

III. Key Technologies

A. PHP Technology

PHP (Hypertext Preprocessor) is a server-side HTML embedded scripting language. Its style is similar to the C language and has been applied widely. PHP mixes the C, Java and Perl grammar as well as its self-innovated grammars. It can execute the dynamic webpage more quickly than CGI or Perl. PHP executes the programs by embedding the programs into the HTML documents. Its execution efficiency is much higher than that of CGI, which completely generates the HTML labels. Besides, PHP can execute the compiled code. The compilation can encrypt and optimize the codes and also make them quicker to run. PHP can implement all the functions of CGI, and also supports almost all databases and operating systems.

B. Ajax Technology

AJAX (Asynchronous JavaScript and XML is a development technology, which can create the interactive web applications and also provides a client JavaScript technology acquiring data by calling the backend database. Unlike the traditional Web applications, Ajax makes use of the asynchronous interaction process. The workflow of the visitors to request supporting the AJAX web pages is as shown in the figure 1.

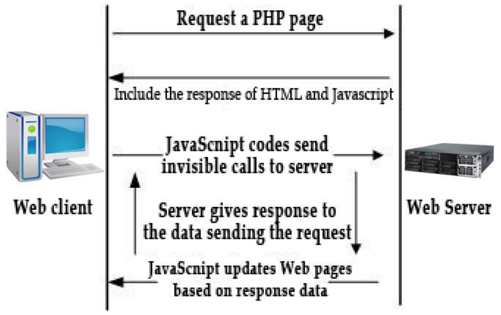

Request a PHP page
Include the response of HTML and Javascript
JavaScnipt codes send invisible calls to server
Web client
Server gives response to the data sending the request
Web Server
JavaScnipt updates Web pages based on response data

Figure 1: the Ajax Calling Workflow

The Ajax has three advantages in the following. First of all, its response speed is highly quicker. The Ajax applications can not only send the essential data to server and also can take back it from server. Therefore, the number of the data exchanged between server and browser decreases greatly. This can make an extreme improvement on system performance. Second, the Web can be refreshed partially. In other words, the data can be maintained under the condition that the entire page is not updated. This makes the WEB applications quicker to give responses to the actions of the users, and also prevents the sending of the information which has nothing changed in Internet. At last, the Ajax is with no need for the plug-ins of browsers. However, it is necessary for the users to allow the JavaScript to be executed in browsers.

IV. PRINCIPLE OF AJAX FORMS AUTHENTICATION

The attribute value is null when the client-side pages are loaded just for a while. When the users fill in some items of a form completely and simultaneously remove the cursors, field values are sent to server, and the system can return the authentication results to the users in real time through the Ajax. If the authentication fails, a friendly message notifying the errors will be displayed, so as to report the authentication results of this field to the users. Then, the users will continue to fill in the other items of the form and authenticate each item of the form by executing the call of the backend data; the authentication results on each item will be returned to the users in real time. At this moment, the attribute value is not null, and therefore the re-loading of the form is prevented owing to the authentication errors. Each item of the form is submitted to the form executive server to be authenticated after being authenticated by the Ajax in real time, and the session variable is used to save the data imported by the users at the same time, so as to prevent the failure of authentication and the refreshing of the form. The users can directly access to the follow-up operations in the system if the authentication is successful. However, a message to notify the errors will be displayed if the authentication is failed, and also the cursor is necessary to be positioned at errors so the users can import the data again and the system can be easy to execute the Ajax authentication steps. The Ajax forms authentication flow is as shown in figure 2.

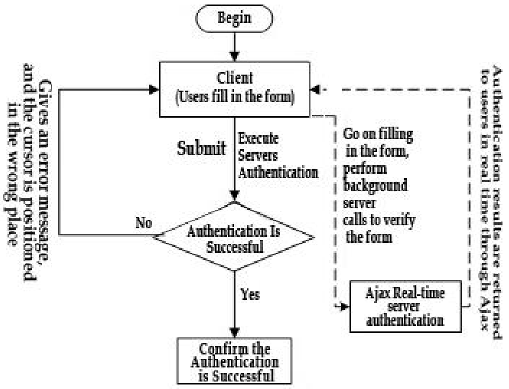

Begin
Client
(Users fill in the form)
Submit
Execute Servers Authentication
Go on filling in the form, perform background server calls to verify the form
No
Authentication Is Successful
Yes
Ajax Real-time server authentication
Confirm the Authentication is Successful
Gives an error message, and the cursor is positioned in the wrong place
Authentication results are returned to users in real time through Ajax

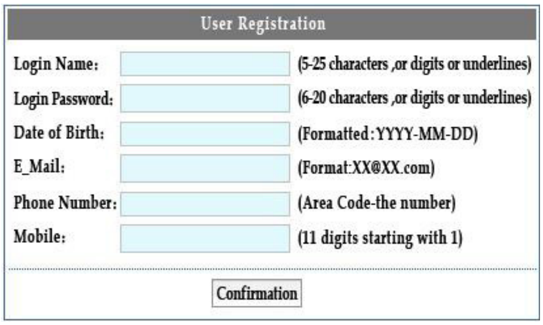
Figure 2: The Ajax Forms Authentication Flow

The Ajax forms authentication realizes not only applying the server side to authenticate the form data after the forms are submitted and also is successful on the client side. In this whole authentication process, the activities of the users are never interrupted. Therefore, it is a reasonable authentication.

V. IMPLEMENTATION OF THE AJAX FORMS AUTHENTICATION

A. Authentication Rules

Take the user registration information in a system for an example. The username, password, date of birth, E-mail, telephone number and mobile phone number and other elements, which are used the most frequently in the forms authentication, can be selected to be authenticated. The imported data is checked according to the following rules. First, the username, which has existed in the database, can't be used, and have to be composed of 5-25 characteristics, digits or underlines. Second, the password can't be null, and have to be composed of 6-20 characteristics, digits or underlines. Third, the date of birth has to be valid, and is formatted as "YYYY-MM-DD". Fourth, the E-mail has to be formatted as xx@xx.xx. Firth, the phone number has to be formatted as "area code-phone number", and should be seven digits at least. Finally, the mobile phone number has to be the eleven digits starting with 1. The user registration interface is as shown in the figure 3.



User Registration

Login Name:		(5-25 characters ,or digits or underlines)
Login Password:		(6-20 characters ,or digits or underlines)
Date of Birth:		(Formatted:YYYY-MM-DD)
E_Mail:		(Format:XX@XX.com)
Phone Number:		(Area Code-the number)
Mobile:		(11 digits starting with 1)

Figure 3 The User Registration Interface

B. *Implementation Of Codes*

An onblur event of the input elements will be generated when the users remove these elements. It can trigger the JavaScript function "validate ()—feild value and feild ID" through two parameters. This function sends an asynchronous HTTP request to the script "validate.php", so as to process the Ajax authentication.

Validate.js can realize the client-side functions. In the Validate.js, the authentication function sends an Ajax request to server. The specific process of initializing an Ajax request is shown in the following. First of all, it is necessary to establish a XMLHttpRequest case. Second, it is necessary to establish a remote server address. At last, it is necessary to initialize the buffering of the authentication request, and then make use of the function to process the authentication of each form feild. The main codes are displayed in the following.

```
    function validate(inputValue, fieldID)
    {
    // connect the server
    try
    {
    //Only going on when the XMLHttpRequest object is
idle and the buffering is not null
    if ((xmlHttp.readyState == 4 || xmlHttp.readyState == 0)
&& cache.length > 0)
    {
    var cacheEntry = cache.shift();
    //gain a new parameter set from buffering
    xmlHttp.open("POST", serverAddress, true);
    //build a service request to authenticate the acquired data
xmlHttp.setRequestHeader("Content-Type",
    "application/x-www-form-urlencoded");
    xmlHttp.onreadystatechange =
handleRequestStateChange;
    xmlHttp.send(cacheEntry);
    }
    }
    }
```

First of all, it is necessary to monitor the change of the server-side state. The attribute readyState of xmlHttp represents the the server-side state. If the state is 0 or 4, it means a request without activities. At this moment, a new request can be sent again. When the new request is being sent, the data read from the buffering can be utilized, and has included the formative asking strings. The function of processing the server responses is the handleRequestStateChange, and successively calls the readResponse() when receiving the response from the server.

Validate. PHP deals with the Ajax calls of the users, and also process the authentication when the forms are submitted. The Ajax authentication is executed through the ValidateAjax method, and its results are used to compose a XML document, which will be returned to the users. The class, which supports the authentication function, is named as "validate", and is embodded into the sctript (validate.class.php) to be processed. The most improtant codes are as shown in the following.

```
    . <?php
    // support Ajax authentication class
    class Validate
    {
    // support Ajax authentication, authenticate a single
value
    public function ValidateAJAX($inputValue, $fieldID)
    {
    switch($fieldID)
    {
    //check which feild should be authenticated and execute
the authentication
    case 'txtPhone':
    //check whether phone number is valid or not
    return $this->validatePhone($inputValue);
    break;
    ………..
    }
    }
    private function validatePhone($value)
    {
    // authenticate phone number, the format: ####-#######
    return (!eregi('^[0-9]{4}--*[0-9]{7}$', $value)) ? 0 : 1;
    }
    ………..
    }
    ?>
```

VI. Conclusion

The development mode, which is based on the browser/server (B/S), has changed into the mainstream of the application system developments at the present time. In the application systems, the function, which is seen by people the most frequently, is the operation of a form validation. Generally speaking, the traditional WEB form authentication programs have a great number of problems such as the interface flashing, the restriction from HTTP, the refreshing of all pages and too long time for a user to wait. Under the environment of PHP, the Ajax form authentication technology takes advantage of the Ajax performance characteristics such as the asynchronization with no refreshes, the quick response and the saving of network bandwidth to greatly improve the response speed of the WEB interfaces, and also provides more natural browsing experience for users.

ACKNOWLEDGMENT

Funded Project: This Paper is supported by Natural Science Foundation of Gansu Province (No. 096RJZA021)

REFERENCES

[1] Wang Jianfeng. The Application of JavaScript to Form Authentication. Computer Programming Skills & Maintenance, (7),pp:29-34,2006.

[2] Zhang Rui, &Gao Zhang. The Application of JS Files to the Optimization of Website Form Authentication and Code Submission. Computer Programming Skills & Maintenance, (11),pp:85-86,2008.

[3] Xu Weiying. Design and Application of a General Scheme for Validating Form Based on JavaScript. Computer Era, (4),pp:43-45, 2010.

[4] Chang Zhonghua. Applying asp to enhance the Implementation of Form Authentication Program. Computer Programming Skills & Maintenance, (4),pp:27-29, 2005.

[5] Yu Tian'an. Detialed Interpretation on PHP Cases. Beijing: Tsinghua University Press.(7),pp: 111-116, 2007.

[6] Yang Huiqin, &Yao Wenlin. Design and Implementation of a reusable Form Validation Class. Financial Computer of Huanan, (3),pp:104-107, 2007.

[7] Tu Xiaoyun. Study on Verification Modes based on Struts. Fujian Computer, (11),pp:51-52, 2008.

[8] [Rumania] Cristian Darie, Bogdan Brinzarea, et al. AJAX and PHP Web Development. Translated by Wang Demin, Wang Xinying, et al. Beijing: The People's Posts and Telecommunications Press, 2007, 4.

[9] Yin Xu, Liao Bin, Zhang Zhihua, &Tian Ying'ai. Research and Application of Ajax-Based Workflow Form. Computer Applications and Software, 27 (3),pp:75-77, 2010.