Design and optimization of two-step verification scheme for form data

Wang Aihua
School of Information and Arts, Shandong Institute of Commerce and Technology, Jinan, Shandong, 250103, China
19940473@sict.edu.cn
Corresponding author’s e-mail: 19940473@sict.edu.cn

Abstract. In the era of big data, we submit a lot of data to the server through the web form interface every day. Data validation is an indispensable link in the process of submitting data. The current verification scheme has some problems, such as untimely verification data, heavy load on the server. This paper analyzes the defects in the single step method in the form data verification by case analysis. On this basis, the two-step scheme of data verification is proposed, the design and optimization ideas of the two-step scheme are given, and the process and principle of the two-step verification are described. The two-step method of form data validation provides an available solution for efficient and fast data validation.

1. Introduction

Form data validation is a function often used in many websites for registration, login or publishing information. The purpose of data validation is not to detect whether the data is correct, but to detect whether the data submitted by the user meets the data requirements. Data verification usually includes two aspects: one is to detect whether the number of characters in the data meets the requirements, the other is to detect whether the character types of the data meet the requirements.

The best way to verify the validity of data is to complete data verification by browser, but in many literatures or projects, the form data verification is completed by the server[3], this practice is not worthy of recognition and promotion. The reasons are explained as follows, if the data that need to be validated is submitted to the server for judgment, there are two main disadvantages:

- First, assume that there are thousands of user data waiting for the server to detect at the same time, when the server verifies, each browser is in the waiting state, the burden on the server will be greatly increased, this will seriously reduce the efficiency of the system;
- Second, after verification, if the data does not meet the requirements, the server needs to return the verification results to the browser, after the user sees the returned prompt information, re-enter the data and submit it to the server again for verification. So again and again, in addition to increasing the burden of the server, it also results in a large increase in Network traffic and prolongs the waiting state of the browser.

If the browser completes the data verification, on the basis of ensuring the legitimacy of the data, it can not only reduce the network traffic, but also reduce the burden of the server and the invalid waiting time of the browser.

The two-step verification method proposed in this paper refers to that for each data that needs to be input, when the input is completed, a verification is performed, and another verification is performed
when the data is submitted, through the double guarantee, it can not only ensure the legitimacy of data, but also increase the user friendliness.

Next, we will take a form verification example to analyze the necessity of two-step verification, the specific methods and the defects in the verification process, and propose solutions.

2. AN EXAMPLE OF FORM DATA VERIFICATION

2.1. Form structure that required data validation
Suppose you need to verify the data validity of the form shown in Figure 1.

![Form Data Verification Diagram](image)

Figure 1. Examples of forms that need to be validated.

For convenience of subsequent description, it is assumed that the name of the password input box is "psw", and the name of the confirmation password input box is "repsw".

2.2. Data validation requirement description
The data validation requirements for the form shown in Figure 1 are as follows:

- The user name must be input, and only alphanumeric characters can be input, and the number of characters must be between 6 and 20.
- Gender in radio button group and hobbies in check box group are required.
- Password must be input, and the number of characters should be between 6 and 18. In addition to characters such as letters and numbers, in order to increase the strength of the password, the specified special symbols are allowed to appear in the password string.
- The password strength is determined according to the character type entered by the user. There are four kinds of characters: uppercase English letters, lowercase English letters, numbers and
special characters. If only one character is input, the password strength is "very weak"; if two characters are input, the password strength is "weak"; if three characters are input, the password strength is "strong"; if four characters are input, the password strength is "very strong". The result of password strength is displayed below the password box.

- The confirmation password must be consistent with the password.
- The mobile phone number must conform to the format of the mobile phone number.

3. DEFECTS IN SINGLE STEP VERIFICATION OF FORM DATA

The method of verifying form data only when submitting can be called single step verification method, which is the most frequently used method.

Early single step verification can only be done by defining the submit event function of the form. The main idea of this function is: In the process of execution, if the submitted data of an input box does not meet the requirements, the function will stop execution and return "false", and prevent the data from being submitted to the server through the returned "false" value[1]. Since HTML5 added new attributes and new form elements, most of the form elements, including text, value, web address, email address, etc., can use HTML5 built-in methods to complete data validation, the number of form data that need to be verified in the submit event verification function is greatly reduced, the common ones are the verification of confirm password and the verification of radio button, check box and other selective elements.

Using the new attributes pattern and required of HTML5 form elements, we can easily verify the data validity of user name, password, mobile phone number and other elements[2]. For example, to verify the validity of the user name, you can add the application of "pattern = '[a-z-a-z0-9] {6,20}' required ="required "" in the user name input box. The former sets regular expression to limit the type and number of characters that can be input, while the latter requires that it cannot be empty. To verify the validity of the password, you can add the application of "pattern = '[a-z-a-z0-9! @ # $% ^ & *] {6,18}' required ="required "" in the password box. To verify the validity of the mobile phone number, you can add "pattern ='[3456789] [0-9] {9}" in the mobile phone number input box, etc.

Using single-step verification method to verify the validity of form data, especially using HTML5 built-in method, although the implementation method is very simple, there are the following defects.

One side, after inputting data, you must wait until you click the submit type button to verify the data, that is, you cannot verify the data immediately after inputting. In this way, if the user enters a password that does not meet the requirements, the system will not be able to detect it in time. There is no doubt that when the user enters the confirmation password according to the password characters, the confirmation password does not meet the requirements, but the system is still unable to detect it in time. Until the user submits the data, the system can find that the password does not meet the requirements. At this point, the user has to change the two passwords at the same time.

Another side, the problem is more serious for the account information that needs server-side duplicate checking. Account duplicate checking needs Ajax technology. If the initial input account is submitted to the server by Ajax for duplicate checking, the feedback result is that the account is not duplicate, only when the data is submitted, it is found that the data itself does not conform to the input specification. You can only re-enter and re-check the account number. It causes the waste of time and resources, so the user friendliness is poor.

4. THE NECESSITY OF USING TWO-STEP METHOD TO VERIFY FORM DATA

In order to make up for the defects of single step verification of form data, so that the user can see whether the data is legal for each data input, so as to remind the user to modify the data in a timely manner in the shortest time, the scheme provided is to add an instant verification link after data input for all non selective data. For example, user name, password, confirmation password, mobile phone number, etc. should be added with instant verification. In the process of data input, there is a close relationship between password and confirmation password, while there is no relationship between user name, mobile phone number and other data, and their instant verification method is very similar to that
of password, therefore, this paper only describes the verification scheme of password and confirm password, and on this basis, puts forward the existing loopholes and solves them.

4.1. Complete the validation of the password

4.1.1. HTML5 validation used when submitting data: In the password input box, add "pattern = '[a-zA-Z0-9! @ # $% ^ & *]{6,18}' required = '', when the user clicks the submit button, the system verifies the validity of the password, If the data meets the requirements, it can be submitted to the server. If it does not meet the requirements, the system will prevent the submission of data[4].

4.1.2. Instant verification using JavaScript functions: Define the anonymous function corresponding to the change event for the password box element, use JavaScript regular expression and test method to judge whether the input password characters meet the requirements. When the user enters the password and the cursor leaves the password box, the element change event will be triggered and the function will be executed. If you want to increase the judgment of password strength, you need to complete it in the anonymous function. Because it has no impact on the verification process, it is not explained here. The function flow is shown in Figure 2.

In this way, the validation of the password will be completed in two different ways at two different time points. The initial verification is performed immediately after the password is entered, however, if this is the only verification process, the problems are as follows: If the user enters a password that does not meet the requirements, the system will prompt the user that the data does not meet the requirements, but it cannot prevent the submission of the data. In other words, if the user ignores the information prompted, does not change the password to meet the requirements of the form, but forcibly submits it. This single verification method that only uses instant verification cannot be prevented. Therefore, it is necessary to add the verification method of HTML5 and verify it again before submitting data.
4.2. Complete the validation of the confirmation password
To verify the validity of the confirmation password, because it involves the data in the password and confirmation password input boxes, HTML5 verification method is not applicable. JavaScript functions are used in both verification steps.

In the first step, when the user enters the confirmation password, the system will verify it immediately. Defines an anonymous function for the element's change event. Get the password and confirmation password in the function, and judge whether they are consistent. If they are inconsistent, a message box will pop up to prompt the user. After the user enters the confirmation password and moves the cursor out of the input box, the change event is triggered, and the system executes the function. The function flow is shown in Figure 3.

The second step is to verify the data when it is submitted.
After verifying the confirmation password by using the flow chart in Figure 3, if the user ignores the prompted information and submits the data forcibly, the system will not prevent the submission process, so there will still be two inconsistent passwords. In order to ensure the consistency between the submitted confirmation password and the password, we must consider that we can verify the legitimacy of the confirmation password when submitting data. The solution provided is to define a validation function for the submit event of the form. When the user clicks the submit button to submit data, the submit event of the form is triggered, and the function is executed to complete the verification process. The specific methods are as follows.

To reduce code duplication and improve execution efficiency, our solution is to define a global variable at the beginning of script code. This variable is used to indicate whether the confirmation password is consistent with the password. If it is inconsistent, the value is set to "false". If it is consistent, the value is set to "true". Based on this requirement, the initial value of the variable is set to "false", which means that the confirmation password is inconsistent with the password at the beginning.

Assuming that the global variable name is "repswflag", the next step is to improve the function flow shown in Figure 3. When the two passwords are inconsistent, set the value of the global variable to "false", that is, add the code "repswflag = false;" in the last position of the original "if" clause. When the two passwords are the same, to set the value of the global variable to "true", you need to add an "else" clause and add the code "repswflag = true" inside it. Then define the submit function of the form, if the value of the global variable in the function is "false", the system will prompt the user again that the passwords of the two times are inconsistent, and return "false" to prevent data submission.

5. VULNERABILITY ANALYSIS AND SOLUTION IN VERIFYING CONFIRMATION PASSWORD

5.1. Vulnerability analysis in verifying confirmation password
After verifying the confirmation password according to the above two-step scheme, we will test it.

The first step is to input the password "sdw123". After the cursor leaves the input box, there is no prompt, indicating that the password characters meet the requirements;

The second step is to enter the confirmation password "sdw111". After the cursor leaves the input box, it will prompt "two passwords are inconsistent, please re-enter". Set the global variable value to "false". The user does not modify it in time after clicking the "OK" button, and keep the global variable value to "false";

In the third step, when the user clicks the "submit" button, the value of "repswflag" is judged to be "false", and the user is prompted again that "two passwords are inconsistent, please re-enter", and the data submission is prevented. At this time, the user must modify the confirmation password. After changing it to "sdw123", judge that the confirmation password is consistent with the password, and change the value of the global variable to "true".

In the fourth step, after entering the correct confirmation password, the user changes the password to "sdw123%" because of the password strength or other reasons. After changing the password, there is no problem with the password data itself, however, since the confirmation password is not re entered, the
verification process of confirming password is not performed again. Therefore, the value of global variable "repswflag" still retains the last result "true". When the user clicks the "submit" button, the system judges that "the two passwords are the same", the result is that the two passwords submitted to the server are inconsistent.

5.2. Solution and principle description provided

In order to solve the problem of data verification vulnerability caused by skipping to re-enter the confirmation password after re-entering the password, we need to modify the function flow of password validation in Figure 2: set "repswflag" to "false" at the beginning of the function. This is because at any time, as long as the password is changed, the confirmation password and the password must be different. If the user changes the password without changing the confirmation password, the value of the variable "repswflag" must be "false". When the user clicks the "submit" button again, the data submission process can be successfully prevented.

6. Conclusion

Form data validation is a function that many websites need to use. The two-step data verification scheme proposed in this paper can verify the data immediately after input, prompt the user to modify the data that does not meet the requirements at the first time, and also verify the data in the last step before all the data are submitted to the server. The dual guarantee blocks all unqualified data on the browser side, ensuring that the data submitted to the server side must meet the requirements. It can not only provide a good user experience, but also effectively reduce the burden of the server. It is a better data verification scheme.

References

[1] YANG H Y. (2019) JavaScript Technology Uses Regular Expressions to Verify Forms. J. Computer Knowledge and Technology., 15(24): 22-24.

[2] LUO H Y. (2019) Regular Expression Discussion in Java Language. J. Computer Knowledge and Technology., 15(32): 95-98.

[3] Guo L, (2017) Validation of form data validity in Struts2 framework. J. Information & communications., 02: 117-118.

[4] Zhang Y S, (2020) Web form client verification skills. J. Computer programming skills and maintenance., 03: 172-175.