

07_03 Writing the Server Side Data Validation Script

In this exercise, you will write the server side data validation script of the update form. You will work in the same page as in the previous video (`profile.cfm`).

1. Use `<cfif>` and `structKeyExists()` to detect the form submission.

At the very top of the 'profile.cfm' page locate the `<!--Form processing begins here-->` and the `<!--Form processing ends here-->` comments. In between these 2 comments, use `<cfif>` and `structKeyExists()` to detect the form submission.

```
<!--Form processing begins here-->
  <cfif structKeyExists(form, 'form.fld_editUserSubmit')>
  </cfif>
<!--Form processing ends here-->
```

2. Create the `aErrorMessages` array.

In the body of the `<cfif>`, use `<cfset>` and `arrayNew()` to create the `aErrorMessages` array. Don't hesitate to add your own meaningful comments in the code.

```
<!--Form processing begins here-->
  <cfif structKeyExists(form, 'form.fld_editUserSubmit')>
    <!--Server side form validation-->
    <cfset aErrorMessages = arrayNew(1) />
  </cfif>
<!--Form processing ends here-->
```

3. Create the 'formSubmitComplete' flag and set it to 'false'

This variable will help you track the successful completion of the update process.

```
<!--Form processing begins here-->
  <cfif structKeyExists(form, 'form.fld_editUserSubmit')>
    <!--Server side form validation-->
    <cfset aErrorMessages = arrayNew(1) />
    <cfset variables.formSubmitComplete = false />
  </cfif>
<!--Form processing ends here-->
```

- In this context the word 'flag' is a substitute for 'boolean variable'. It means that a flag can only have 2 values : 'true' or 'false'. You set the flag to 'false' right now, but you'll set it to 'true' later in the process.
- The value of this variable (`false`) is NOT surrounded by quotes. Remember that only the values of type 'string' have to be surrounded by quotes. In this case, the value is not of type 'string' but of type 'boolean' and 'false' is a keyword that ColdFusion is able to understand natively.

4. Validate the presence of the required fields as well as the Email address pattern.

The required fields are the first Name (`fld_userFirstName`), the last name (`fld_userLastName`), the email address (`fld_emailAddress`) and both password fields (`fld_userPassword` and `fld_userPasswordConfirm`).

```
<!--Server side form validation-->
<cfset aErrorMessages = arrayNew(1) />
<cfset variables.formSubmitComplete = false />
<!--Validate firstName-->
<cfif form.fld_userFirstName EQ ''>
    <cfset arrayAppend(aErrorMessages,'Please provide a first name') />
</cfif>
<!--Validate lastName-->
<cfif form.fld_userLastName EQ ''>
    <cfset arrayAppend(aErrorMessages,'Please provide a last name') />
</cfif>
<!--Validate Email-->
<cfif form.fld_userEmail EQ '' OR NOT isValid('email',form.fld_userEmail)>
    <cfset arrayAppend(aErrorMessages,'Please provide a valid email
        address')/>
</cfif>
<!--Validate Password-->
<cfif form.fld_userPassword EQ '' >
    <cfset arrayAppend(aErrorMessages,'Please provide a password ')/>
</cfif>
<!--Validate Password confirmation-->
<cfif form.fld_userPasswordConfirm EQ '' >
    <cfset arrayAppend(aErrorMessages,'Please confirm your password')/>
</cfif>
```

Remember that if the user submits an empty text field, the field does exist in the form scope of the action page but is passed as an empty string. This is the reason why `isDefined()` and `structKeyExists()` are not appropriate in this case.

5. Make sure both password fields have the same value.

```
<!---Validate Password confirmation--->
<cfif form.fld_userPasswordConfirm EQ '' >
    <cfset arrayAppend(aErrorMessages,'Please confirm your password')/>
</cfif>
<!---validate password and password confirmation Match --->
<cfif form.fld_userPassword NEQ form.fld_userPasswordConfirm >
    <cfset arrayAppend(aErrorMessages,'The password and the password
        confirmation do not match')/>
</cfif>
```

6. Continue form processing if the aErrorMessages array is empty

If the aErrorMessages array is empty, it means that the submitted data successfully passed all the validation tests. In this case, you can safely move on to the actual user update in the database.

```
<!---Continue if the aErrorMessages array is empty--->
<cfif arrayIsEmpty(aErrorMessages)>
</cfif>
```

The actual update action will be developed in the next video.

7. Display the errorMessages if the aErrorMessages array is not empty.

To wrap up the server side form validation, you'll return to the form and output the error messages to the user if the following 2 conditions are met :

- The aErrorMessages array exists.
- The aErrorMessages array is not empty.

First, scroll down the page and locate the <!---Output error messages if any---> comment.

Right below the comment write the proper <cfif> tag block.

```
<!---Output error messages if any--->
<cfif isDefined('aErrorMessages') AND NOT arrayIsEmpty(aErrorMessages)>
</cfif>
```

In the body of the <cfif>, create a <cfoutput> tag block.

```
<cfif isDefined('aErrorMessages') AND NOT arrayIsEmpty(aErrorMessages)>
    <cfoutput>
    </cfoutput>
</cfif>
```

Next, create a `<cfl loop>` tag block that loops over the 'aErrorMessages' array. Make the current item of the loop available in the 'message' variable.

```
<cfif isDefined('aErrorMessages') AND NOT arrayIsEmpty(aErrorMessages)>
    <cfoutput>
        <cfl loop array="#aErrorMessages#" index="message">
        </cfl loop>
    </cfoutput>
</cfif>
```

In the body of the `<cfl loop>`, use a `<p>` tag block with the 'errorMessage' CSS class to output the value of the 'message' variable.

```
<cfif isDefined('aErrorMessages') AND NOT arrayIsEmpty(aErrorMessages)>
    <cfoutput>
        <cfl loop array="#aErrorMessages#" index="message">
            <p class="errorMessage">#message#</p>
        </cfl loop>
    </cfoutput>
</cfif>
```

8. Save and run the page.

Once the page has loaded in the browser, change the password and confirm password fields so that they do not match, then, submit the form. When the page refreshes, your error message should be displayed at the top of the form.



Update your profile

Your profile

The password and the password confirmation do not match

First Name

Test