



CMER

Centre for Mobile Education and Research

Form Handling, Validation and Functions



Form Handling

- Forms are a graphical user interfaces (GUIs) that enables the interaction between users and servers
- Form controls are grouped inside the `<form>` element
- Some common controls: buttons, text boxes, checkboxes, etc.



Form Handling (continued)

- In order to manipulate data from forms, a server must be equipped with a technology for data handling
- Some common technologies include ColdFusion, Java Server Pages (JSP), ASP.NET, Common Gateway Interface (CGI), etc.



Form Attributes

- There are two form attributes

action

This attribute specifies the URL that is intended to receive the data from form elements

Method

This attribute specifies the type of HTTP method by which data is sent to the server



Action Attribute

- This attribute is essential in any form
 - It indicates the location to which form data will be sent

Example: `<form action="CollectData.cfm" ... >`

- When submitting a form, the location of the Webpage must be specified



Method Attribute

- There are two common form methods

GET

Data sent by the form is appended to the URL specified in the Action attribute

POST

Data sent by the form is not appended to the URL specified in the Action attribute



Method Attribute Examples

GET

<http://www.google.com/search?hl=en&q=BlackBerry>

Form data

POST

<http://www.google.com/search>

Data is transmitted part of the HTTP Request instead of the URL



Form Tags

- **Submit & Reset** button tags have two attributes in this example:
 - **TYPE**: tells the browser what kind of button to create
 - **VALUE**: specifies the message that is to be displayed on the button



Input Form Tags

- Input tags are placeholders for information data that is transmitted to the action URL

Example

```
<form action="someURL.cfm">  
  Your First name: <input name="FirstName" type="text" /><br />  
  Your Last name: <input name="LastName" type="text" />  
</form>
```



Your First name:
Your Last name:



Form Example in BlackBerry

- Assume that we would like to collect user data in a form including:
 - Name (**text box**)
 - BlackBerry device type (**drop down menu**)
 - Email address (**text box**)
 - Newsletter option (**checkbox**)
- Save the file as **FormExample.cfm**



Form Example in BlackBerry (continued)

- HTML code

```
<form action="processData.cfm" method="POST">
  Name: <input name="firstName" type="text" /><br />
  BlackBerry Device: <select name="deviceType">
    <option value="8800">BlackBerry 8800</option>
    <option value="8820">BlackBerry 8820</option>
    <option value="Pearl">BlackBerry Pearl</option>
    <option value="Curve">BlackBerry Curve</option>
  </select><br />
  Email: <input name="emailAddress" type="text" /><br />
  Subscribe to Newsletter: <input type="radio" name="newsLetter" value="Yes" checked="checked" />Yes
    <input type="radio" name="newsLetter" value="No" />No <br />
</form>
```

action

method

input text box

drop down menu

input text box

radio button



Form Example in BlackBerry (continued)

- Browser output

Name:
BlackBerry Device:
Email:
Subscribe to Newsletter: Yes No

- One important item is missing

– the “**submit**” button

Submit my info



Form Example in BlackBerry (continued)

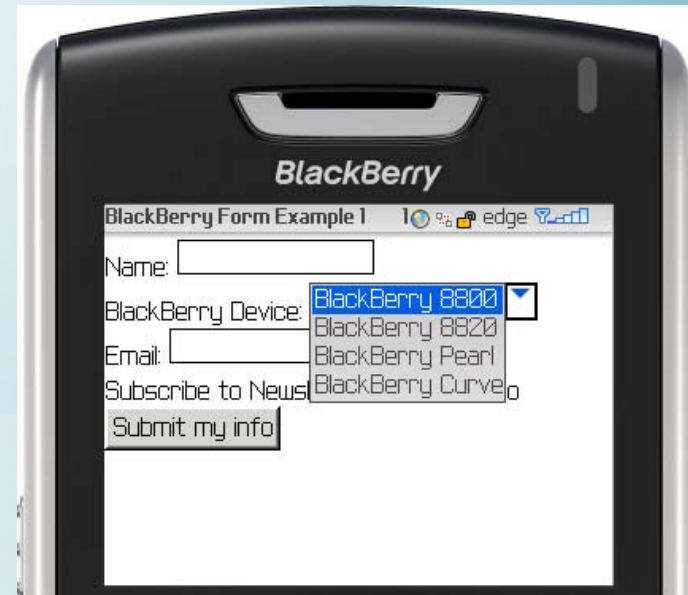
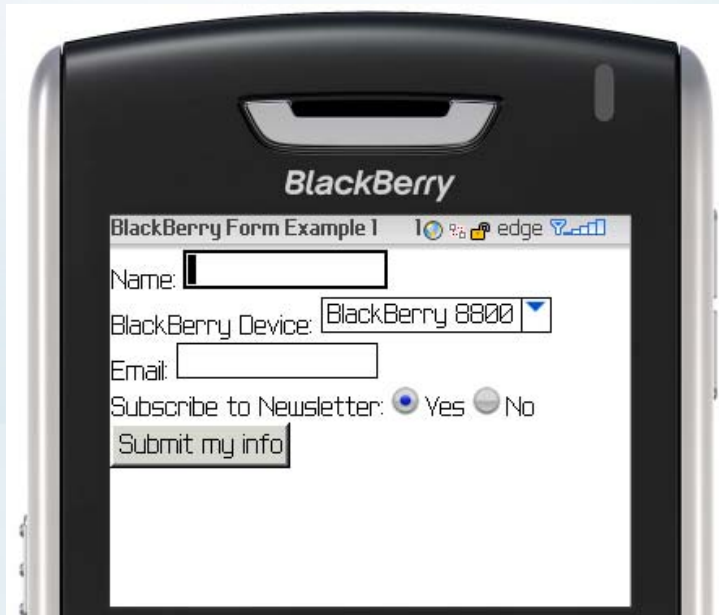
- updated HTML code

```
<form action="processData.cfm" method="POST">
  Name: <input name="firstName" type="text" /><br />
  BlackBerry Device: <select name="deviceType">
    <option value="8800">BlackBerry 8800</option>
    <option value="8820">BlackBerry 8820</option>
    <option value="Pearl">BlackBerry Pearl</option>
    <option value="Curve">BlackBerry Curve</option>
  </select><br />
  Email: <input name="emailAddress" type="text" /><br />
  Subscribe to Newsletter:   <input type="radio" name="newsLetter" value="Yes" checked="checked" />Yes
                             <input type="radio" name="newsLetter" value="No" />No <br />
  <input type="submit" value="Submit my info" />
</form>
```

submit button



Form Example in BlackBerry (continued)

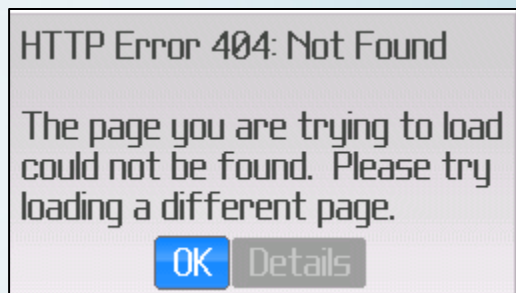




Collecting Data from Form Elements

- Now that the form is ready to send data; the URL in the action attribute must be able to collect the values from the form elements
- The URL then can process the data and return a response message to the user
 - *If you attempt to click on the “Submit my info” button, you will get an error because the action does not exist yet*

BlackBerry error message





Data Collection from Forms

- Create a file called “processData.cfm”
 - *Modify the action attribute in the form to this URL*

```
<form action="processData.cfm" method="POST">
```

- To collect values of form variables in ColdFusion, the form variable is used

- **Syntax: <cfset form.var_name>**

- **Example:**

```
<!--- Retrieve form values --->  
<cfset fname="#form.firstName#">  
<cfset device="#form.deviceType#">  
<cfset email="#form.emailAddress#">  
<cfset subscription="#form.newsLetter#">
```




Data Collection from Forms (continued)

- Action URL code

```
<!-- Retrieve form values -->
<cfset fname=#form.firstName#>
<cfset device=#form.deviceType#>
<cfset email=#form.emailAddress#>
<cfset subscription=#form.newsLetter#>

<head>
<title>Thank you <cfoutput>#fname#</cfoutput></title>
</head>

<body>
Thank you <cfoutput>#fname#</cfoutput> for submitting your information. <br />
Your device is: <cfoutput>#device#</cfoutput> and email address is: <cfoutput>#email#</cfoutput>. <br />
Subscription: <cfoutput>#subscription#</cfoutput>.
</body>
</html>
```



Data Collection from Forms (continued)

- You can also use the output command to display the values of more than one variable at once

Example

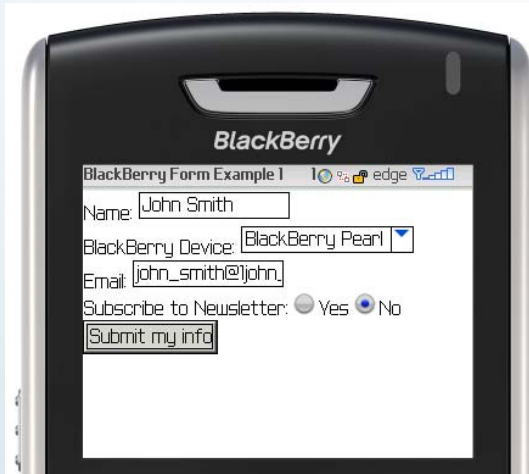
```
<!-- Retrieve form values -->
<cfset fname = #form.firstName# >
<cfset device = #form.deviceType# >
<cfset email = #form.emailAddress# >
<cfset subscription = #form.newsLetter# >

<head>
<title>Thank you </title>
</head>

<body>
<cfoutput>
Thank you #fname# for submitting your information. <br />
Your device is: #device# and email address is: #email#. <br />
Subscription: #subscription#.
</cfoutput>
</body>
</html>
```



Data Collection



Name:

BlackBerry Device:

Email:

Subscribe to Newsletter: Yes No



Thank you John Smith for submitting your information.
Your device is: Pearl and email address is: john_smith@john_smith_com.
Subscription: No.



Validating Forms

- **Form validation is sometimes necessary when completing forms**
 - It ensures that users enter data into a field
 - Check the validity of data entered
- **Three types of form validation**
 - Server-side (using CFML)
 - Client-side (using JavaScript)



Server-Side CFML Validation

- **These controls are usually hidden**
 - **Prebuilt: CFML contains validation hidden fields that are predefined**
 - **Example: required data entry**

```
<input name="name" type="text" />  
<input name="name_required" type="hidden" value="Please enter your name" />
```

Value of the name attribute must match the form control plus the special suffix

input type hidden is used

value attribute contains the error message



Adding Form Validation to FormExample.cfm

- Modify FormExample.cfm to validate user input and checks for missing data fields

```
<form action="processData.cfm" method="POST">
  Name: <input name="firstName" type="text" /><br />
  <input type="hidden" name="firstName_required" value="You must enter your name" />
  BlackBerry Device: <select name="deviceType">
    <option value="8800">BlackBerry 8800</option>
    <option value="8820">BlackBerry 8820</option>
    <option value="Pearl">BlackBerry Pearl</option>
    <option value="Curve">BlackBerry Curve</option>
  </select><br />
  Email: <input name="emailAddress" type="text" /><br />
  <input type="hidden" name="emailAddress_required" value="You must enter your email address" />
  Subscribe to Newsletter:   <input type="radio" name="newsLetter" value="Yes" checked="checked" />Yes
                           <input type="radio" name="newsLetter" value="No" />No <br />
  <input type="submit" value="Submit my info" />
</form>
```



Adding Form Validation to FormExample.cfm (continued)

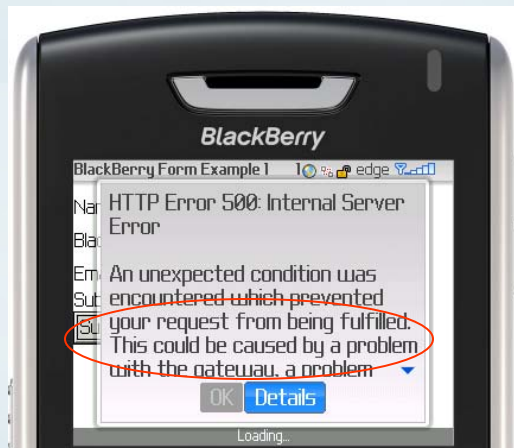
- Browser output

Form entries are incomplete or invalid.

- You must enter your name
- You must enter your email address

Go [back](#) and correct the problem.

- BlackBerry





Custom Server-side Form Validation

- Developers can create their own validation code
- This type of validation is required at the action URL and not the form URL
 - i.e. **processData.cfm** and not **FormExample.cfm**
- Requires the use of if conditional statements
 - **<CFIF**>



Custom Server-side Form Validation (continued)

```
<!-- Retrieve form values -->
<cfset fname = #form.firstName# >
<cfset device = #form.deviceType# >
<cfset email = #form.emailAddress# >
<cfset subscription = #form.newsLetter# >

<head>
<title>Thank you </title>
</head>

<body>
<cfif fname EQ "" >
    <font color="#FF0000" face="Arial, Helvetica, sans-serif">Error! You did not enter your name</font><br />
    Please go back and enter your name.
<cfelseif email EQ "" >
    <font color="#FF0000" face="Arial, Helvetica, sans-serif">Error! You did not enter a valid email address</font><br />
    Please go back and enter your email address.
<cfelse>
<cfoutput>
    Thank you <b>#fname</b> for submitting your information. <br/>
    Your device is: <b>#device</b> and email address is: <b>#email#.</b><br />
    Subscription: <b>#subscription</b>.
</cfoutput>
</cfif>

</body>
</html>
```

processData.cfm



Running Custom Server-side validation

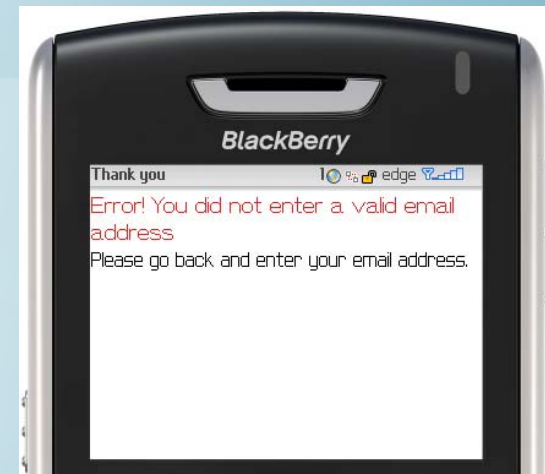
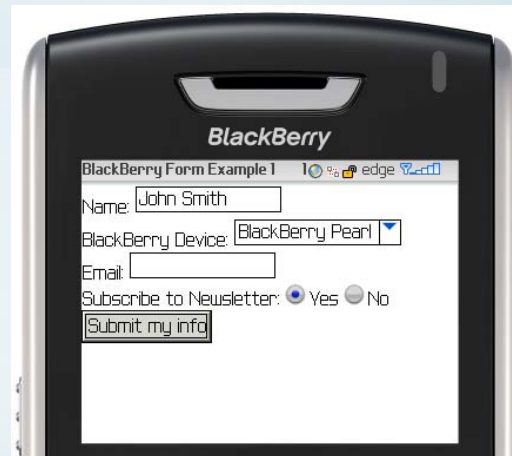
- Browser

Name:
BlackBerry Device:
Email:
Subscribe to Newsletter: Yes No



Error! You did not enter a valid email address
Please go back and enter your email address.

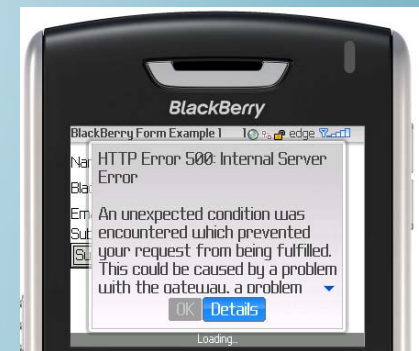
- BlackBerry





Server-side Form Validation: Summary

- Using the built-in validation may cause BlackBerry devices to “think” of exceptions as server side internal errors
 - *This is due to the way ColdFusion handles exceptions*
- It is recommended in this case to use your own validation code





Client-side Form Validation

- **Using server-side validation may increase the overhead for mobile clients**
 - **requires active connections to/from the device**
 - **User submits the form**
 - **Server validates the data**
 - **On errors, send message**
 - **User checks form data and resubmits**
 - **etc...**
- **Client-side validation is performed within the browsers using built-in JavaScript**



Client-side Form Validation (continued)

- Since mobile devices are limited in memory size, using too many JavaScript validations may consume too much memory
- Mobile devices, including BlackBerry, have limited browser versions
 - BlackBerry supports JavaScript 1.1, 1.2, 1.3 and subsets of 1.4 and 1.5
- BlackBerry devices must have JavaScript support enabled (**default is disabled**) in browser configuration





Client-side Form Validation Example

- To use CFM client-side form validation, one must use CFM controls
 - i.e. `<cfinput>`, `<cfform>`, etc.
- Using `FormExample.cfm`, let's add client-side validation for the fields **firstName** and **emailAddress**
- Example
 - `<cfinput type="..." validate="..." pattern="..." message="...">`



Client-side Form Validation Example (continued)

this will cause a message box to appear

```
<cfform action="processData.cfm" method="POST">
  Name: <cfinput name="firstName" type="text" required="yes" message="Please enter your name" /><br />
  BlackBerry Device: <cfselect name="deviceType">
    <option value="8800">BlackBerry 8800</option>
    <option value="8820">BlackBerry 8820</option>
    <option value="Pearl">BlackBerry Pearl</option>
    <option value="Curve">BlackBerry Curve</option>
  </cfselect><br />
  Email: <cfinput name="emailAddress" type="text" /><br />
  Subscribe to Newsletter:   <cfinput type="radio" name="newsLetter" value="Yes" checked="checked" />Yes
                             <cfinput type="radio" name="newsLetter" value="No" />No <br />
  <cfinput name="submit" type="submit" value="Submit my info" />
</cfform>
```

important: You must declare a name attribute for CFM controls, otherwise you will receive an error



Client-side Form Validation Example (continued)

- Browser

Name:

BlackBerry Device: BlackBerry Pearl

Email: john_smith@john_smith_c

Subscribe to Newsletter: Yes No

Submit my info

Windows Internet Explorer

Please enter your name

OK

- BlackBerry

BlackBerry

BlackBerry Form Example 1

Name:

BlackBerry Device: BlackBerry Pearl

Email: john_smith@john_smith_c

Subscribe to Newsletter: Yes No

Submit my info

Please enter your name

OK



Client-side Form Validation Regular Expressions

- Form validation can also use regular expressions
- The following example defines a valid format for email addresses

PATTERN="[A-Za-z0-9_]+@[A-Za-z0-9_]+\.[A-Za-z]+"

```
Email: <cfinput name="emailAddress" type="text" validate="regular_expression"  
pattern="[A-Za-z0-9_]+@[A-Za-z0-9_]+\.[A-Za-z]+"  
message="Please enter a valid email address"/><br />
```



Client-side Form Validation Regular Expressions (continued)

- Browser

A screenshot of a web browser window showing a form with the following fields: Name: John Smith, BlackBerry Device: BlackBerry 8800, Email: john, and Subscribe to Newsletter: Yes (selected). A 'Submit my info' button is visible. Below the form, a 'Windows Internet Explorer' dialog box displays a yellow warning icon and the message 'Please enter a valid email address' with an 'OK' button.

- BlackBerry

A screenshot of a BlackBerry device screen displaying the same form as the browser screenshot. The form fields are Name: John Smith, BlackBerry Device: BlackBerry Curve, Email: john, and Subscribe to Newsletter: Yes. A dialog box with a blue information icon and the message 'Please enter a valid email address' is overlaid on the form, with an 'OK' button.



Functions

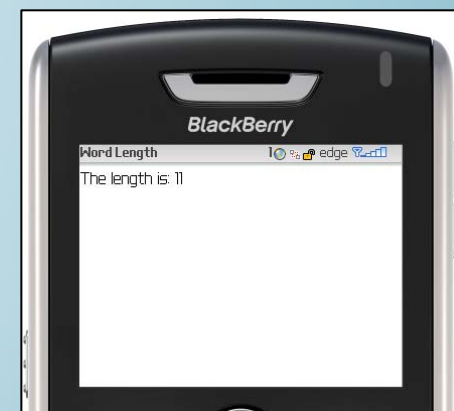
- Function is a piece of code that performs a specific task

- Format

*function_name(param₁,
param₂,...param_n)*

- Example

```
<cfset wordLength = len("Hello World")>  
The length is: <cfoutput>#wordLength#</cfoutput>
```





Common Built-in Functions

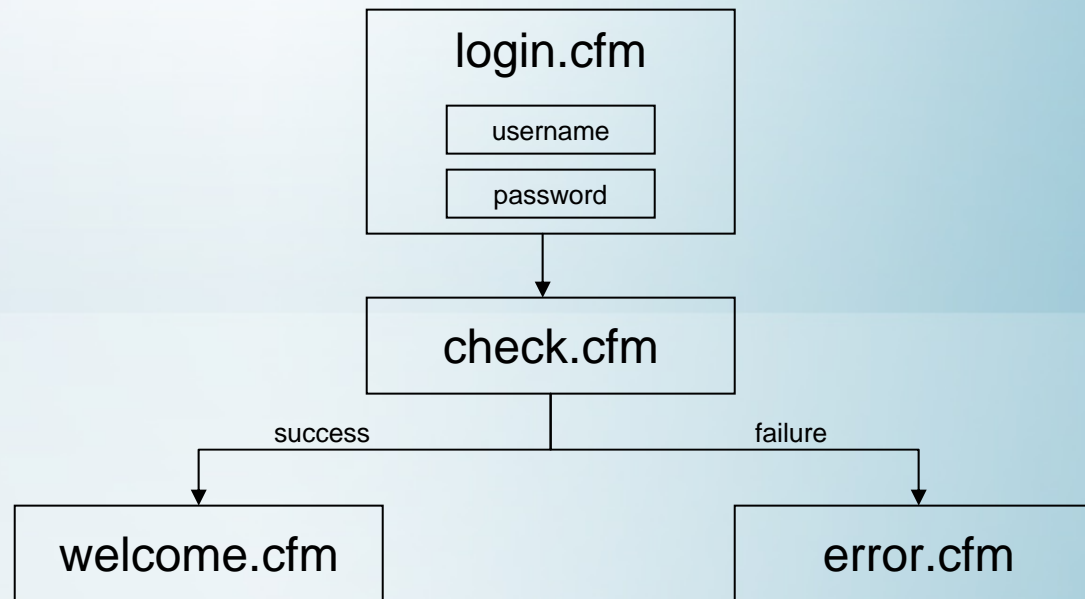
- **Now()** returns the system date and time
- **UCase(...)** returns the uppercase of a string
- **LCase(...)** returns the lowercase of a string
- **Trim(...)** removes spaces
- **FileExists(path)** checks the existence of a file
- **Compare(str1,str2)** compares two strings

More functions can be found in <http://livedocs.adobe.com/coldfusion/6.1/htmldocs/function.htm>



Log In Example

- Let's create a form that contains two input text boxes

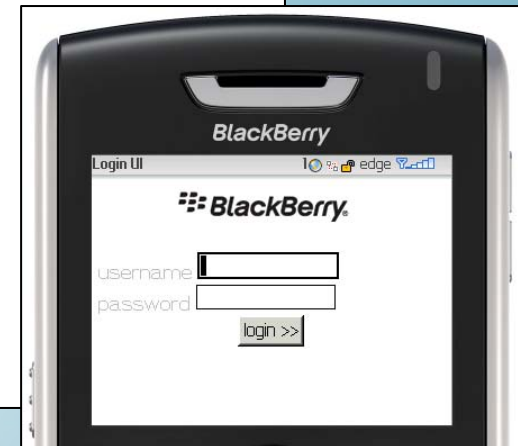




Log In Example login.cfm

login.cfm

```
<html>
<head>
<title>Login UI</title>
</head>
<body>
<p align="center"><img src='blackberry.gif' /></p>
<cfform name="LoginUI" method="post" action="check.cfm">
<table width="100%" border="0">
  <tr>
    <td><font face="Arial, Helvetica, sans-serif" >username</font></td>
    <td>
      <cfinput name="username" type="text" required="yes" size="10"
        message="Please enter a username"/>
    </td>
  </tr>
  <tr>
    <td><font face="Arial, Helvetica, sans-serif">password</font></td>
    <td>
      <cfinput name="password" type="password" required="yes" size="11"
        message="Please enter a password"/>
    </td>
  </tr>
  <tr>
    <td colspan="2"><div align="center"></div>
    <div align="center">
      <cfinput type="submit" name="login" value="login >>"/>
    </div></td>
  </tr>
</table>
</cfform>
<p>&nbsp;</p>
</body>
</html>
```





Log In Example

check.cfm

check.cfm

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>checking...</title>
</head>

<body>
<cfset username = #form.username# >
<cfset password = #form.password# >

<cfif username EQ "jsmith" and password EQ "jpass" >
  <cflocation url = "welcome.cfm">
<cfelse>
  <cflocation url = "error.cfm">
</cfif>

</body>
</html>
```



Log In Example

welcome.cfm and error.cfm

welcome.cfm

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Welcome</title>
</head>

<body>
<font face="Arial, Helvetica, sans-serif" color="#0000CC">You have successfully logged in</font>
</body>
</html>
```

error.cfm

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Welcome</title>
</head>

<body>
<font face="Arial, Helvetica, sans-serif" color="#FF0000">Error is username or password. Please try again.</font>
</body>
</html>
```




Log In Example Output

Browser
BlackBerry

