FlowTasks – HR Application

Contents

[Introduction 1](#_Toc333219003)

[Resume 1](#_Toc333219004)

[Database 2](#_Toc333219005)

[Code 2](#_Toc333219006)

# Introduction

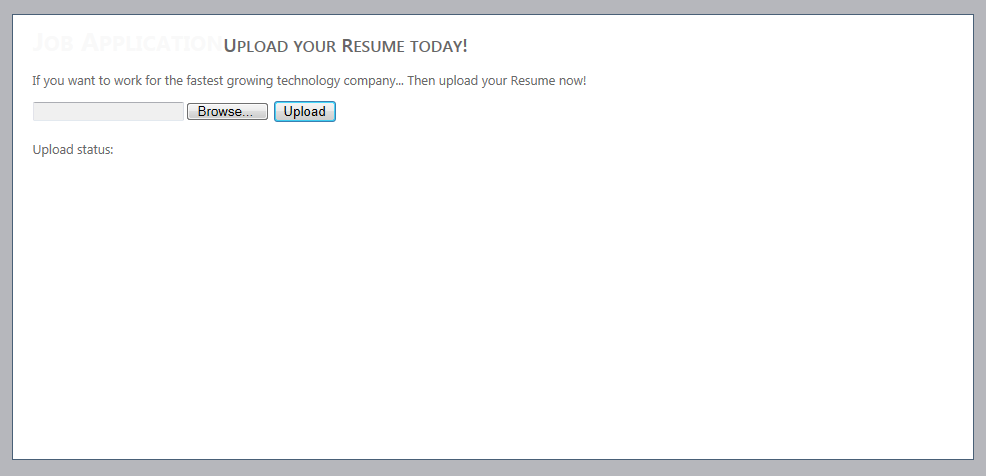
Now that you know how to create workflows and task views, otherwise please read “FlowTasks How To Create a Task View.docx” and “FlowTasks How To Create a Workflow.docx”, you can start to build a little bit more complex applications.

All the code for this sample can be found under Demo/HrApp.

# Resume

Let say the company you work for has a web page where someone can send you his/her Resume for a possible job opportunity.

The code for this application can be found in Demo\HrApp\Web.



This application uses the DocsOnFolder service. If you haven’t already installed, check out the document “FlowTasks Installation Guide.docx” for the instruction on how to install this service.

Make sure that the value for the setting key “ResumeLoadPath” points to the same location where the DocsOnFolder service refers to (the setting key called ImportLocation).

<appSettings>

<add key="ResumeLoadPath" value="C:\your path here"/>

</appSettings>

# Database

You need to create the users and the roles for the application. Run the scripts under “Demo/HrApp/Sql” called HrUsers.sql which will create all the correct data for you.

Basically there will be two new users:

Nkidman, who belongs to the “HR” role, and tcruise who belongs to the “Mgr” role.

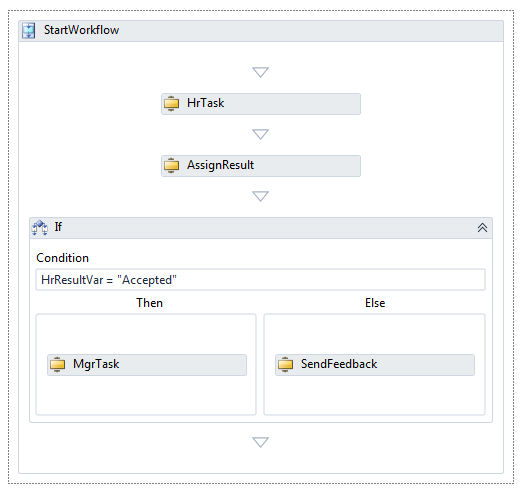
Both user passwords are “pwd”.

Run the script called HrTasks.sql to create the workflow configuration.

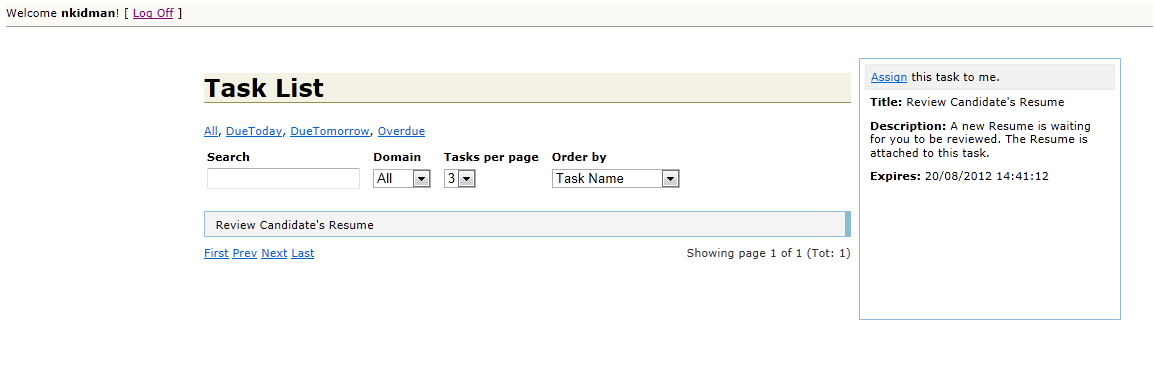
# Code

To build the code for the workflow, check the other documents “FlowTasks How To Create a Task View.docx” and “FlowTasks How To Create a Workflow.docx”. I will not go into the same details here, but will only explain the extra bits of code that are different in this application.

First of all have a look at the workflow definition. There is a first task for the HR users, if the candidate Resume looks OK they will then send the candidate to the Mgr users otherwise will send a feedback to the applicant.



When a HR users logs in he/she will see the TaskList. There will be a task for every Resume sent.



Then once they click on the task the task view appears.



If you check the code you see that for example the textbox for the name is defined like this

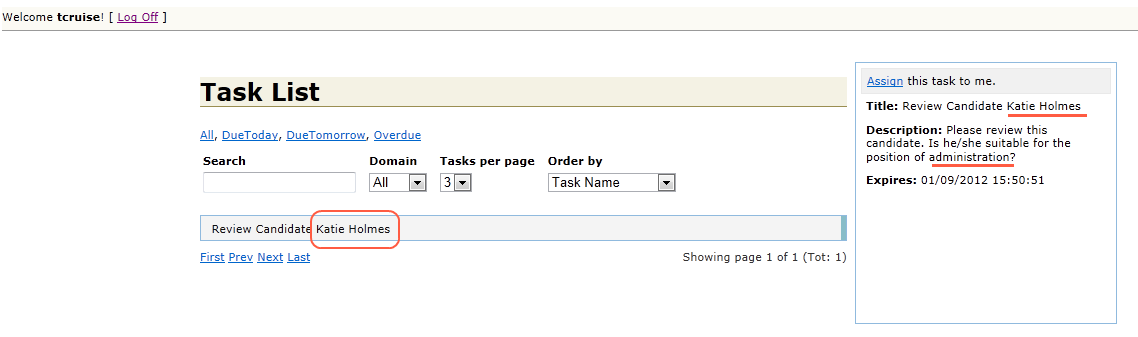
<input id="TaskParameterCandidateName" name="TaskParameterCandidateName" type="text" />

The id has a prefix “TaskParameter” this is a convention that tells the view to “send” this element to the workflow (more about this later).

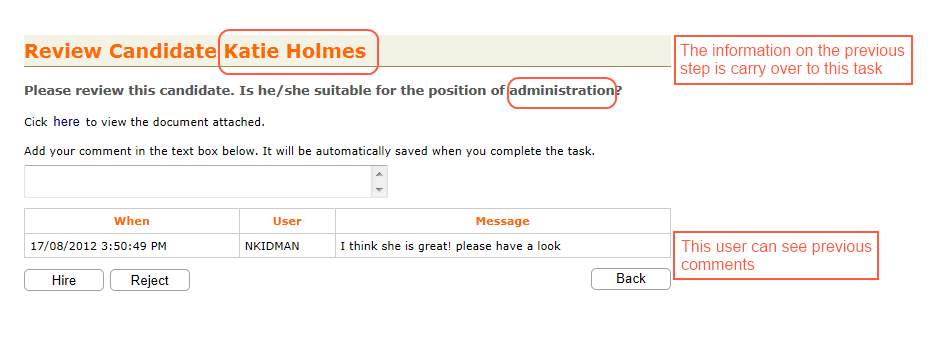
Another thing to notice is that the button is called “Hire”. The default for the view when you want to approve a task is “Accept”; if you want to overwrite this default use the bellow code:

@Html.Hidden("TaskAcceptFlag", "Hire")

If HR press “Hire”, when an Mgr user logs in he/she will found a task waiting in the tasklist.



As you can see both the title and the description are customized by the input from the previous task.



To do this you just need to define the task’s title in the following way:

Title = "Review Candidate {p.CandidateName}"

And make sure that you copy the parameters you need from the previous task “HrTask” to the current one like this:

TaskStatus taskStatus = Request.Get(context);

var hrTask = taskStatus.WorkflowStatusRef.Tasks["HrTask"];

if (hrTask.Parameters.ContainsKey("CandidateName"))

{

taskStatus.Parameters.Add("CandidateName",

hrTask.Parameters["CandidateName"]);

}

if (hrTask.Parameters.ContainsKey("CandidateJobType"))

{

taskStatus.Parameters.Add("CandidateJobType",

hrTask.Parameters["CandidateJobType"]);

}

You can define new parameters in your view like explain above with the “TaskParameter” prefix

# Javascript and CSS

You can add you own styles and javascript like this:

Html.Assets().Styles.Add("/Content/hr.css");

Html.Assets().Scripts.Add("/Scripts/jquery.validate.min.js");

Html.Assets().Scripts.Add("/Scripts/HrView.js");

Then remember to copy these files under the TaskList web application root path.