\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Detail Design: CCO eCoaching Admin

Common

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prepared by: Lili Huang Date: 11/28/2016

Department: CCO

Change History Log

| Date | Change Description | Author |
| --- | --- | --- |
| 11/28/2016  03/07/2017  03/16/2017 | Initial Revision  TFS 5694 – Delete employee logs  Updated:  2.1.3. Links to other modules  7.1. IDashboardRepository.cs  7.2. IEmployeeLogRepository.cs  8.1. IDashboardService.cs  8.3 IEmployeeLogService.cs  TFS 5904 – Create infrastructure within eCL Admin Tool to allow for reporting;  TFS 5908 – Create Summary eCoaching Reports. | Lili Huang  Lili Huang  Lili Huang |
|  |  |  |

Table of Contents

[**1.** **Description** 5](#_Toc477424882)

[**2.** **Views** 5](#_Toc477424883)

[2.1. Home page (Views\Share\Index.cshtm) 5](#_Toc477424884)

[2.1.1. Layout page (Views\Share\\_Layout.cshtml) 5](#_Toc477424885)

[2.1.2. Screenshot 5](#_Toc477424886)

[2.1.3. Links to other modules 5](#_Toc477424887)

[**3.** **Controllers** 6](#_Toc477424888)

[3.1. BaseController.cs 6](#_Toc477424889)

[3.1.1. OnException 6](#_Toc477424890)

[3.1.2. GetUserFromSession 6](#_Toc477424891)

[3.1.3. IsAccessAllowed 6](#_Toc477424892)

[3.2. ErrorController.cs 6](#_Toc477424893)

[3.2.1. Index – Http Get 6](#_Toc477424894)

[3.3. HomeController.cs 6](#_Toc477424895)

[3.3.1. Index – Http Get 6](#_Toc477424896)

[3.4. UnauthorizedController.cs 6](#_Toc477424897)

[3.4.1. Index – Http Get 6](#_Toc477424898)

[**4.** **Extensions** 6](#_Toc477424899)

[4.1. EclExtendedMethods.cs 6](#_Toc477424900)

[4.1.1. IsEntitled 6](#_Toc477424901)

[4.1.2. ShowPreviousStatus 6](#_Toc477424902)

[4.2. EclHtmlHelper.cs 6](#_Toc477424903)

[4.3. EclExtendedMethods.cs 6](#_Toc477424904)

[4.3.1. ToDescription 7](#_Toc477424905)

[**5.** **Filters** 7](#_Toc477424906)

[5.1. EclAuthorizeAttribute.cs 7](#_Toc477424907)

[5.1.1. OnAuthorization 7](#_Toc477424908)

[**6.** **Utilities** 7](#_Toc477424909)

[6.1. EclAdminUtil.cs 7](#_Toc477424910)

[6.1.1. BuildLogStatusNamesDictionary 7](#_Toc477424911)

[6.1.2. GetLogTypeNameById 7](#_Toc477424912)

[6.2. EmailUtil.cs 7](#_Toc477424913)

[6.2.1. GetTemplateFileName 7](#_Toc477424914)

[6.2.2. GetSubject 7](#_Toc477424915)

[6.2.3. GetEmailTo 7](#_Toc477424916)

[**7.** **Repository** 7](#_Toc477424917)

[7.1. IDashboardRepository.cs 7](#_Toc477424918)

[7.1.1. GetLogCount 7](#_Toc477424919)

[7.1.2. GetLogList 7](#_Toc477424920)

[7.1.3. GetChartDataCoachingCompleted 7](#_Toc477424921)

[7.1.4. GetChartDataCoachingPending 7](#_Toc477424922)

[7.1.5. GetChartDataWarningActive 7](#_Toc477424923)

[7.1.6. GetLogListTotal 7](#_Toc477424924)

[7.2. IEmployeeLogRepository.cs 7](#_Toc477424925)

[7.2.1. GetModules 8](#_Toc477424926)

[7.2.2. GetLogsByEmpIdAndAction 8](#_Toc477424927)

[7.2.3. GetPendingLogsByReviewerEmpId 8](#_Toc477424928)

[7.2.4. GetPendingStatuses 8](#_Toc477424929)

[7.2.5. GetReasons 8](#_Toc477424930)

[7.2.6. ProcessActivation 8](#_Toc477424931)

[7.2.7. ProcessReassignment 8](#_Toc477424932)

[7.2.8. GetCoachingDetail 8](#_Toc477424933)

[7.2.9. GetWarningDetail 8](#_Toc477424934)

[7.2.10. Delete 8](#_Toc477424935)

[7.3. IEmployeeRepository.cs 8](#_Toc477424936)

[7.3.1. GetEmployees 8](#_Toc477424937)

[7.3.2. GetPendingReviewers 8](#_Toc477424938)

[7.3.3. GetAssignToList 8](#_Toc477424939)

[7.4. IUserRepository.cs 9](#_Toc477424940)

[7.4.1. GetUserByLanId 9](#_Toc477424941)

[7.4.2. GetEntitlementByUserLanId 9](#_Toc477424942)

[**8.** **Services** 9](#_Toc477424943)

[8.1. IDashboardService.cs 9](#_Toc477424944)

[8.1.1. GetLogCount 9](#_Toc477424945)

[8.1.2. GetLogList 9](#_Toc477424946)

[8.1.3. GetChartDataCoachingCompleted 9](#_Toc477424947)

[8.1.4. GetChartDataCoachingPending 9](#_Toc477424948)

[8.1.5. GetChartDataWarningActive 9](#_Toc477424949)

[8.1.6. GetLogListTotal 9](#_Toc477424950)

[8.2. IEmailService.cs 9](#_Toc477424951)

[8.2.1. Send 9](#_Toc477424952)

[8.3. IEmployeeLogService.cs 9](#_Toc477424953)

[8.3.1. GetModules 9](#_Toc477424954)

[8.3.3. GetLogsByEmpIdAndAction 9](#_Toc477424955)

[8.3.4. GetPendingLogsByReviewerEmpId 9](#_Toc477424956)

[8.3.5. GetPendingStatuses 10](#_Toc477424957)

[8.3.6. GetReasons 10](#_Toc477424958)

[8.3.7. ProcessActivation 10](#_Toc477424959)

[8.3.8. ProcessReassignment 10](#_Toc477424960)

[8.3.9. GetLogDetail 10](#_Toc477424961)

[8.3.10. GetLogsByLogName 10](#_Toc477424962)

[8.3.11. Delete 10](#_Toc477424963)

[8.4. IEmployeeService.cs 10](#_Toc477424964)

[8.4.1. GetEmployees 10](#_Toc477424965)

[8.4.2. GetPendingReviewers 10](#_Toc477424966)

[8.4.3. GetAssignToList 10](#_Toc477424967)

[8.5. IUserService.cs 10](#_Toc477424968)

[8.5.1. GetUserByLanId 10](#_Toc477424969)

[8.5.2. UserIsEntitled 10](#_Toc477424970)

1. **Description**

This document covers the common code used by all of the ECL Admin UI modules.

The common code consists of the following:

Views\

* Index.cshtml
* \_Layout.cshtml
* SessionExpire.cshtml

Controller\

* BaseController.cs
* ErrorController.cs
* HomeController.cs
* UnauthorizedController.cs

Extensions\

* EclExtendedMethods.cs
* EclHtmlHelper.cs
* EnumExtensions.cs

Filters\

* EclAuthorizeAttribute.cs
* EclExceptionAttribute.cs
* SessionCheckAttribute.cs

Utilities\

* EclAdminUtil.cs
* EmailUtil.cs
* FileUtil.cs

Repository\

* IDashboardRepository.cs
* IEmployeeLogRepository.cs
* IEmployeeRepository.cs
* IUserRepository.cs

Services\

* IDashboardService.cs
* IEmailService.cs
* IEmployeeLogService.cs
* IEmployeeService.cs
* IUserService.cs

1. **Views**
   1. Home page (Views\Share\Index.cshtm)
      1. Layout page (Views\Share\\_Layout.cshtml)
      2. Screenshot



* + 1. Links to other modules

|  |  |
| --- | --- |
| Link Link Name | Destination |
| Dashboard | Views\Dashboard\Index.cshtml |
| Manage Employee Logs | Dropdown menu display with the following links based on the user Role(s):  Inactivate Employee Logs (Admin and manager roles)  Reassign Employee Logs (Admin and manager roles)  Reactivate Employee Logs (Admin role) |
| Inactivate Employee Logs | Inactivate Employee Logs page  Views\EmployeeLogs\SearchForInactivate.cshtml |
| Reassign Employee Logs | Reassign Employee Logs page  Views\EmployeeLogs\SearchForReassign.cshtml |
| Reactivate Employee Logs | Reactivate Employee Logs page  Views\EmployeeLogs\SearchForReactivate.cshtml |
| Delete Employee Logs | Delete Employee Logs page  Views\EmployeeLogs\SearchForDelete.cshtml |
| Report | Report page  Views\Report\ReportTemplate.cshtml |

1. **Controllers**
   1. BaseController.cs
      1. OnException

Use log4net to log any exceptions thrown in all controllers.

Display Error page.

* + 1. GetUserFromSession

Return the user object stored in session.

* + 1. IsAccessAllowed

Check if the user has access to a given function by calling UserService.UserIsEntitled (user, entitlement).

* 1. ErrorController.cs
     1. Index – Http Get

Display Error page.

* 1. HomeController.cs
     1. Index – Http Get

Get the user LAN ID for the current HTTP request.

Get the user object by calling UserService.GetUserByLanId(userLanId).

If it returns null

Redirect the user to the unauthorized page.

Else

Store the user object returned in Session.

Display the main menu page.

* 1. UnauthorizedController.cs
     1. Index – Http Get

Abandon session and display unauthorized page.

1. **Extensions**
   1. EclExtendedMethods.cs
      1. IsEntitled

Check if the user is authorized to perform a given action by calling UserService.UserIsEntitled(user, entitlementName).

* + 1. ShowPreviousStatus

Check if it displays logs’ previous status for a given action.

* 1. EclHtmlHelper.cs
  2. EclExtendedMethods.cs
     1. ToDescription

Return enum description for a given value.

1. **Filters**
   1. EclAuthorizeAttribute.cs
      1. OnAuthorization

Called upon requesting authorization. Check if the user is allowed to perform a given action.

1. **Utilities**
   1. EclAdminUtil.cs
      1. BuildLogStatusNamesDictionary

Return a dictionary with keys as log status id and values as a list of log name for a given list of logs.

* + 1. GetLogTypeNameById

Return the log type name for a given log type id.

* 1. EmailUtil.cs
     1. GetTemplateFileName

Return email template file name for a given email type.

* + 1. GetSubject

Return email subject text for a given email type.

* + 1. GetEmailTo

Return email address for a given module id and log status id.

1. **Repository**
   1. IDashboardRepository.cs
      1. GetLogCount

Return logs count for a given user, log status, log type, start time, and end time by calling stored procedure sp\_SelectFrom\_SRMGR\_Count.

* + 1. GetLogList

Return a list of logs for a given user, log status, log type, start time, and end time by calling stored procedure sp\_SelectFrom\_SRMGR\_Details.

* + 1. GetChartDataCoachingCompleted

Return a list of ChartCoachingCompleted objects for a given user, start time, and end time by calling stored procedure sp\_SelectFrom\_SRMGR\_Completed\_CoachingByWeek.

* + 1. GetChartDataCoachingPending

Return a list of ChartCoachingPending objects for a given user, start time, and end time by calling stored procedure sp\_SelectFrom\_SRMGR\_Pending\_CoachingByWeek.

* + 1. GetChartDataWarningActive

Return a list of ChartWarningActive objects for a given user, start time, and end time by calling stored procedure sp\_SelectFrom\_SRMGR\_Active\_WarningByWeek.

* + 1. GetLogListTotal

Return total number of logs for a given user, log status, log type, start time, end time, and search text by calling stored procedure sp\_SelectFrom\_SRMGR\_Detail\_Count.

* 1. IEmployeeLogRepository.cs
     1. GetModules

Return a list of modules for a given user and log type by calling stored procedure sp\_AT\_Select\_Modules\_By\_LanID.

* + 1. GetLogsByEmpIdAndAction

Return a list of employee log objects for a given module, log type, employee, and action by calling stored procedure sp\_AT\_Select\_Logs\_Inactivation\_Reactivation.

* + 1. GetPendingLogsByReviewerEmpId

Return a list of employee log objects for a given module, status, and reviewer employee id by calling stored procedure sp\_AT\_Select\_Logs\_Reassign.

* + 1. GetPendingStatuses

Return a list of pending status for a given module by calling stored procedure sp\_AT\_Select\_Status\_By\_Module.

* + 1. GetReasons

Return a list of Reason objects for a given log type and action to perform by calling stored procedure sp\_AT\_Select\_Action\_Reasons.

* + 1. ProcessActivation

Inactivate or activate a given a list of logs for a given user, action to perform, log type, reason to inactivate or activate, reason text, and comment by calling stored procedure sp\_AT\_Coaching\_Inactivation\_Reactivation for log type Coaching or sp\_AT\_Warning\_Inactivation\_Reactivation for log type Warning.

* + 1. ProcessReassignment

Re-assign a given a list of logs for a given user to another reviewer by calling stored procedure sp\_AT\_Coaching\_Reassignment passing in userLanId, a list of log ids to be reassigned, reviewer employee id to be reassigned to, re-assign reason id, re-assign reason text, and re-assign comment text.

* + 1. GetCoachingDetail

Return CoachingLogDetail object for a given log id by calling stored procedure sp\_SelectFrom\_SRMGR\_Review.

* + 1. GetWarningDetail

Return WarningLogDetail object for a given log id by calling stored procedure sp\_SelectFrom\_SRMGR\_Review.

* + 1. Delete

Delete a log for a given log id and log type.

Return true if success; false otherwise.

* 1. IEmployeeRepository.cs
     1. GetEmployees

Return a list of Employee objects for a given user, log type, module, and action to perform by calling stored procedure sp\_AT\_Select\_Employee\_Inactivation\_Reactivation.

* + 1. GetPendingReviewers

Return a list of pending reviewers (Employee objects) for a given user, module, and log status by calling stored procedure sp\_AT\_Select\_ReassignFrom\_Users.

* + 1. GetAssignToList

Return a list of Employee objects to assign to for a given user, module, log status, and current reviewer by calling stored procedure sp\_AT\_Select\_ReassignTo\_Users.

* 1. IUserRepository.cs
     1. GetUserByLanId

Return a User object for a given user lan id.

* + 1. GetEntitlementByUserLanId

Return a list of Entitlement objects for a given user by calling stored procedure sp\_AT\_Check\_Entitlements.

1. **Services**
   1. IDashboardService.cs
      1. GetLogCount

Return total number of logs for a given user, log status, log type, start time, and end time by calling DashboardRepository.GetLogCount.

* + 1. GetLogList

Return a list of logs for a given user, status, log type, start time, and end time by calling DashboardRepository.GetLogList.

* + 1. GetChartDataCoachingCompleted

Return a list of ChartCoachingCompleted objects for a given user, start time, and end time by calling DashboardRepository.GetChartDataCoachingCompleted.

* + 1. GetChartDataCoachingPending

Return a list of ChartCoachingPending objects for a given user, start time, and end time by calling DashboardRepository.GetChartDataCoachingPending.

* + 1. GetChartDataWarningActive

Return a list of ChartWarningActive objects for a given user, start time, and end time by calling DashboardRepository.GetChartDataWarningActive.

* + 1. GetLogListTotal

Return total number of logs for a given user, log status, log type, start time, end time, and search text by calling DashboardRepository.GetLogListTotal.

* 1. IEmailService.cs
     1. Send

Send email.

* 1. IEmployeeLogService.cs
     1. GetModules

Return a list of modules for a given user and log type by calling EmployeeLogRepository.GetModules.

* + 1. GetTypes

Return a list of log types for a given user and action the user to perform.

* + 1. GetLogsByEmpIdAndAction

Return a list of employee log objects for a given module, log type, employee, and action by calling EmployeeLogRepository.GetLogsByEmpIdAndAction.

* + 1. GetPendingLogsByReviewerEmpId

Return a list of employee log objects for a given module, status, and reviewer employee id by calling EmployeeLogRepository.GetPendingLogsByReviewerEmpId.

* + 1. GetPendingStatuses

Return a list of pending status for a given module by calling EmployeeLogRepository.GetPendingStatuses.

* + 1. GetReasons

Return a list of Reason objects for a given log type and action to perform by calling EmployeeLogRepository.GetReasons.

* + 1. ProcessActivation

Inactivate or activate a given a list of logs for a given user, action to perform, log type, reason to inactivate or activate, reason text, and comment by calling EmployeeLogRepository.ProcessActivation.

Return true if successful, false otherwise.

* + 1. ProcessReassignment

Re-assign a given a list of logs for a given user to another reviewer by calling EmployeeLogRepository.ProcessReassignment passing in userLanId, a list of log ids to be reassigned, reviewer employee id to be reassigned to, re-assign reason id, re-assign reason text, and re-assign comment text.

Return true if successful, false otherwise.

* + 1. GetLogDetail

Return log detail object for a given log id and log type by calling EmployeeLogRepository.GetCoachingDetail or by calling EmployeeLogRepository.GetWarningDetail.

* + 1. GetLogsByLogName

Return a list of logs for a given log name by calling EmployeeRepository.GetLogsByLogname.

* + 1. Delete

Delete a log for a given log id and log type by calling EmployeeRepository.Delete.

* 1. IEmployeeService.cs
     1. GetEmployees

Return a list of Employee objects for a given user, log type, module, and action to perform by calling EmployeeRepository.GetEmployees.

* + 1. GetPendingReviewers

Return a list of pending reviewers (Employee objects) for a given user, module, and log status by calling EmployeeRepository.GetPendingReviewers.

* + 1. GetAssignToList

Return a list of Employee objects to assign to for a given user, module, log status, and current reviewer by calling EmployeeRepository.GetAssignToList.

* 1. IUserService.cs
     1. GetUserByLanId

Return a User object for a given user lan id by calling UserRepository.GetUserByLanId.

* + 1. UserIsEntitled

Return a User object for a given user lan id by calling UserRepository.GetUserByLanId.