MEMBER PORTAL SERVICES

Adding a New Method

Member Portal Services and Member Portal Proxy

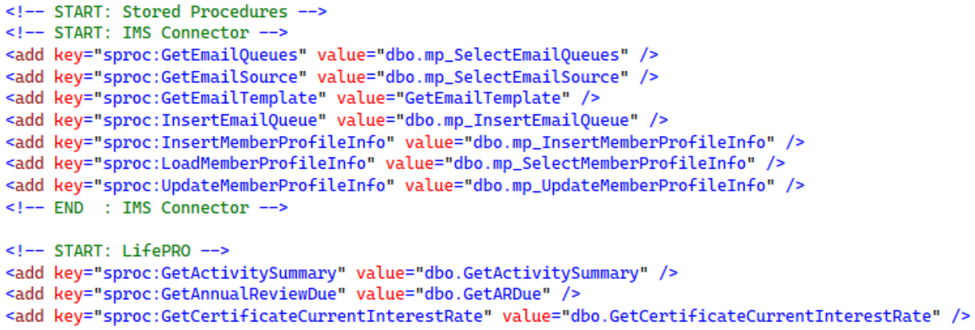
I. Member Portal Service

This section assumes the Developer is adding a new method that is using a SQL Stored Procedure to perform a database function, such as Delete, Insert, Select, or Update.

1. Adding a Stored Procedure

a. web.config

The web.config file contains a section for Stored Procedure names. The section is broken out by database:



```
<!-- START: Stored Procedures -->
<!-- START: IMS Connector -->
<add key="sproc:GetEmailQueues" value="dbo.mp_SelectEmailQueues" />
<add key="sproc:GetEmailSource" value="dbo.mp_SelectEmailSource" />
<add key="sproc:GetEmailTemplate" value="GetEmailTemplate" />
<add key="sproc:InsertEmailQueue" value="dbo.mp_InsertEmailQueue" />
<add key="sproc:InsertMemberProfileInfo" value="dbo.mp_InsertMemberProfileInfo" />
<add key="sproc:LoadMemberProfileInfo" value="dbo.mp_SelectMemberProfileInfo" />
<add key="sproc:UpdateMemberProfileInfo" value="dbo.mp_UpdateMemberProfileInfo" />
<!-- END  : IMS Connector -->

<!-- START: LifePRO -->
<add key="sproc:GetActivitySummary" value="dbo.GetActivitySummary" />
<add key="sproc:GetAnnualReviewDue" value="dbo.GetARDue" />
<add key="sproc:GetCertificateCurrentInterestRate" value="dbo.GetCertificateCurrentInterestRate" />
```

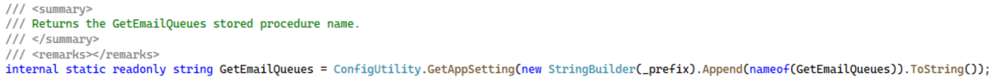
NOTE: Add your stored procedure name to the appropriate section.

b. AppProperties

Go to the AppProperties folder in the RNA.MemberPortal.DataLibrary project. Open the appropriate Stored Procedure class that corresponds to the where the Stored Procedure is located. For example, if the stored procedure is in the IMSConnector database you would open the ImsConnectorStoredProcedures.cs file.

Add the stored procedure to the Stored Procedure class.

Example:



```
/// <summary>
/// Returns the GetEmailQueues stored procedure name.
/// </summary>
/// <remarks></remarks>
internal static readonly string GetEmailQueues = ConfigUtility.GetAppSetting(new StringBuilder(_prefix).Append(nameof(GetEmailQueues)).ToString());
```

NOTE: The <StoredProcedureName> must match the name used for the Stored Procedure in the web.config file.

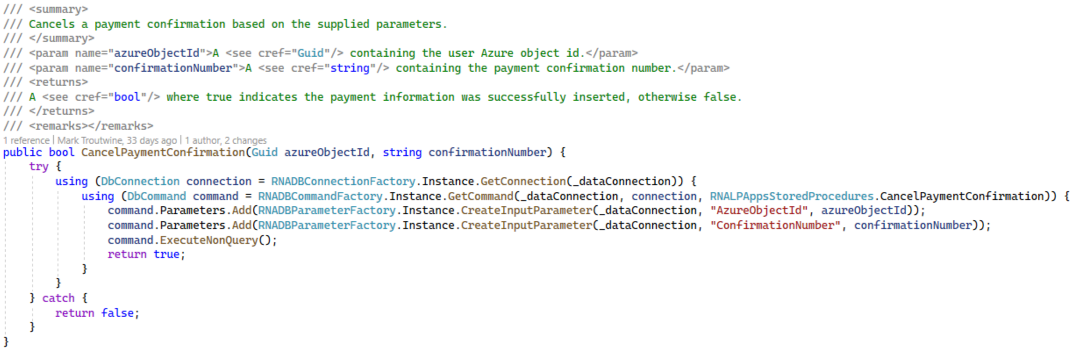
2. **Add Database Method to Database Class**

   Database classes are divided by workspace/folder. The workspace/folder is the same name as the database.
   The database class is named after the database as well with the term “Data” appended to the end.

   Locate the database class for the method being added. Add the new method to the class.

   Example:

   1. A new method needs to be added for cancellation of a payment in the RNALPApps database.
   2. Go to the RNA.MemberPortal.DataLibrary project.
   3. Locate the RNALPApps folder.
   4. Open the RNALPAppsData class.
   5. Add your method:



```
/// <summary>
/// Cancels a payment confirmation based on the supplied parameters.
/// </summary>
/// <param name="azureObjectId">A <see cref="Guid"/> containing the user Azure object id.</param>
/// <param name="confirmationNumber">A <see cref="string"/> containing the payment confirmation number.</param>
/// <returns>
/// A <see cref="bool"/> where true indicates the payment information was successfully inserted, otherwise false.
/// </returns>
/// <remarks></remarks>
1 reference | Mark Troutwine, 33 days ago | 1 author, 2 changes
public bool CancelPaymentConfirmation(Guid azureObjectId, string confirmationNumber) {
    try {
        using (DbConnection connection = RNADBConnectionFactory.Instance.GetConnection(_dataConnection)) {
            using (DbCommand command = RNADBCommandFactory.Instance.GetCommand(_dataConnection, connection, RNALPAppsStoredProcedures.CancelPaymentConfirmation)) {
                command.Parameters.Add(RNADBParameterFactory.Instance.CreateInputParameter(_dataConnection, "AzureObjectId", azureObjectId));
                command.Parameters.Add(RNADBParameterFactory.Instance.CreateInputParameter(_dataConnection, "ConfirmationNumber", confirmationNumber));
                command.ExecuteNonQuery();
                return true;
            }
        }
    } catch {
        return false;
    }
}
```

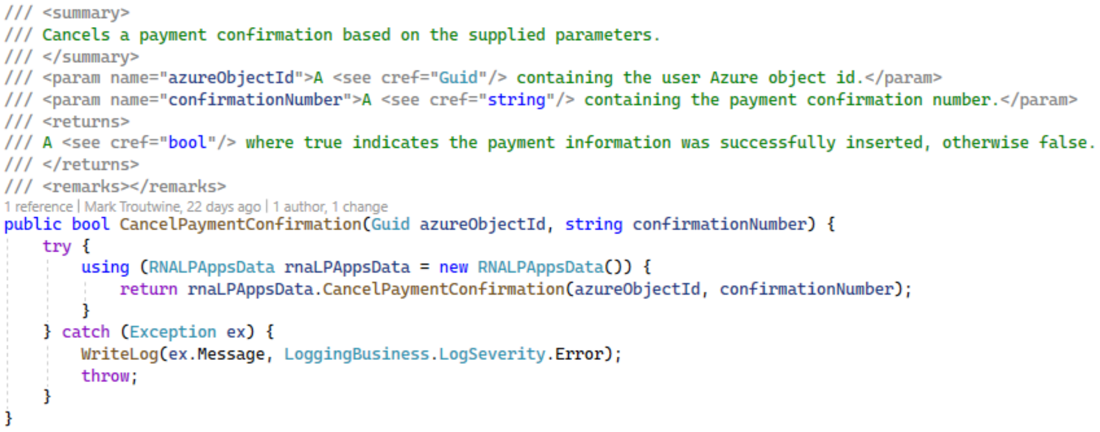
3. **Add Business Method to the Business Class**

   Business classes are divided by workspace/folder. The workspace/folder is the same name as the database.
   The business class is named after the database as well with the term “Business” appended to the end.

   Locate the business class for the method being added. Add the new method to the class.

   Example:

   1. A new method needs to be added for cancellation of a payment using the RNALPApps database class method.
   2. Go to the RNA.MemberPortal. BusinessLibrary project.
   3. Locate the RNALPApps folder.
   4. Open the RNALPAppsBusiness class.
   5. Add your method:



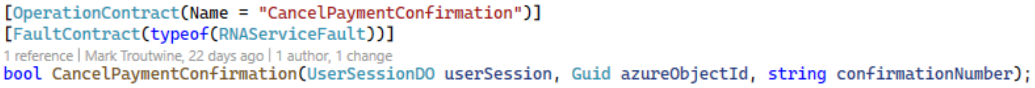
```
/// <summary>
/// Cancels a payment confirmation based on the supplied parameters.
/// </summary>
/// <param name="azureObjectId">A <see cref="Guid"/> containing the user Azure object id.</param>
/// <param name="confirmationNumber">A <see cref="string"/> containing the payment confirmation number.</param>
/// <returns>
/// A <see cref="bool"/> where true indicates the payment information was successfully inserted, otherwise false.
/// </returns>
/// <remarks></remarks>
1 reference | Mark Troutwine, 22 days ago | 1 author, 1 change
public bool CancelPaymentConfirmation(Guid azureObjectId, string confirmationNumber) {
    try {
        using (RNALPAppsData rnaLPAppsData = new RNALPAppsData()) {
            return rnaLPAppsData.CancelPaymentConfirmation(azureObjectId, confirmationNumber);
        }
    } catch (Exception ex) {
        WriteLog(ex.Message, LoggingBusiness.LogSeverity.Error);
        throw;
    }
}
```

4. **Add WCF Method to the WCF Contract**

Add the new method declaration to the class.

Example:

1. A new method declaration needs to be added for cancellation of a payment.
2. Go to the RNA.Services.WCF.Contracts project.
3. Locate the RNA.Services.WCF.Contracts folder.
4. Open the IMemberPortalService class.
5. Add your method:

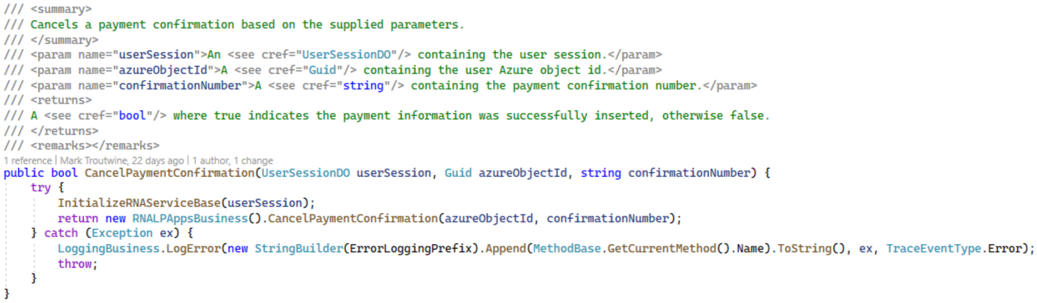


```
[OperationContract(Name = "CancelPaymentConfirmation")]
[FaultContract(typeof(RNAServiceFault))]
1 reference | Mark Troutwine, 22 days ago | 1 author, 1 change
bool CancelPaymentConfirmation(UserSessionDO userSession, Guid azureObjectId, string confirmationNumber);
```

5. Add a Method to the Member Portal Service

Example:

1. A new method needs to be added for cancellation of a payment to the Member Portal service.
2. Go to the RNA.MemberPortalServices project.
3. Open the MemberPortalService.svc class.
4. Add your method:



```
/// <summary>
/// Cancels a payment confirmation based on the supplied parameters.
/// </summary>
/// <param name="userSession">An <see cref="UserSessionDO"/> containing the user session.</param>
/// <param name="azureObjectId">A <see cref="Guid"/> containing the user Azure object id.</param>
/// <param name="confirmationNumber">A <see cref="string"/> containing the payment confirmation number.</param>
/// <returns>
/// A <see cref="bool"/> where true indicates the payment information was successfully inserted, otherwise false.
/// </returns>
/// <remarks></remarks>
1 reference | Mark Troutwine, 22 days ago | 1 author, 1 change
public bool CancelPaymentConfirmation(UserSessionDO userSession, Guid azureObjectId, string confirmationNumber) {
    try {
        InitializeRNAServiceBase(userSession);
        return new RNALPAppsBusiness().CancelPaymentConfirmation(azureObjectId, confirmationNumber);
    } catch (Exception ex) {
        LoggingBusiness.LogError(new StringBuilder(ErrorLoggingPrefix).Append(MethodBase.GetCurrentMethod().Name).ToString(), ex, TraceEventType.Error);
        throw;
    }
}
```

II. Member Portal Proxy

This section assumes the Developer has already added the method to the Member Portal Service.

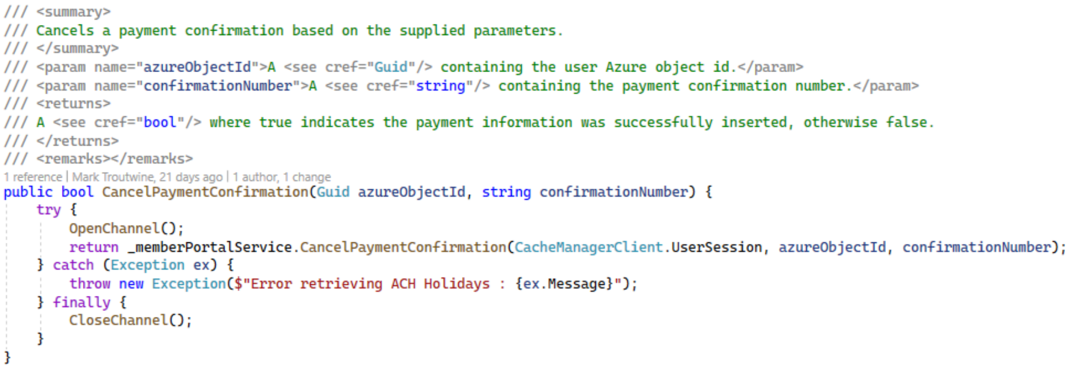
NOTE: The Member Portal Service and Member Portal Proxy share the following projects. Updates to either of these projects is reflected in both the Member Portal Service and Member Portal Proxy:

- RNA.MemberPortal.ObjectLibrary
- RNA.Services.WCF.Contracts

1. Add a Method to the Member Portal Proxy Data Provider

Example:

1. Go to the RNA.MemberPortalProxy project.
2. Expand the DataProvider folder.
3. Open the MemberPortalDataProvider.cs class.
4. Add your method:

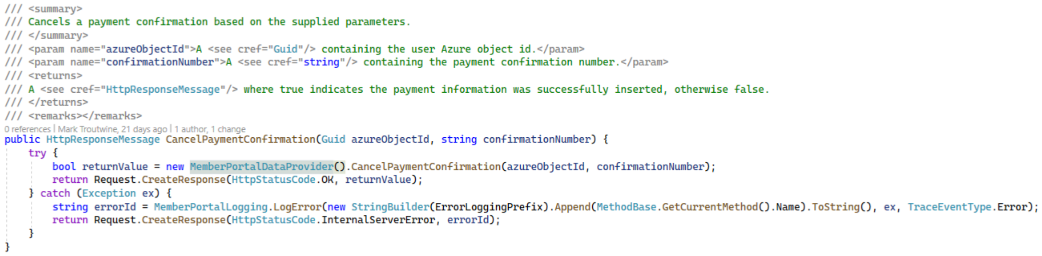


```
/// <summary>
/// Cancels a payment confirmation based on the supplied parameters.
/// </summary>
/// <param name="azureObjectId">A <see cref="Guid"/> containing the user Azure object id.</param>
/// <param name="confirmationNumber">A <see cref="string"/> containing the payment confirmation number.</param>
/// <returns>
/// A <see cref="bool"/> where true indicates the payment information was successfully inserted, otherwise false.
/// </returns>
/// <remarks></remarks>
1 reference | Mark Troutwine, 21 days ago | 1 author, 1 change
public bool CancelPaymentConfirmation(Guid azureObjectId, string confirmationNumber) {
    try {
        OpenChannel();
        return _memberPortalService.CancelPaymentConfirmation(CacheManagerClient.UserSession, azureObjectId, confirmationNumber);
    } catch (Exception ex) {
        throw new Exception($"Error retrieving ACH Holidays : {ex.Message}");
    } finally {
        CloseChannel();
    }
}
```

2. Add a Method to the Member Portal Proxy Controller

Example:

1. Determine what controller where the method should be located. In this example we continue adding the cancellation of payment method started in the Member Portal Service.
2. Go to the RNA.MemberPortalProxy project.
3. Expand the DataProvider folder.
4. Open the MemberPortalDataProvider.cs class.
5. Add your method:



```
/// <summary>
/// Cancels a payment confirmation based on the supplied parameters.
/// </summary>
/// <param name="azureObjectId">A <see cref="Guid"/> containing the user Azure object id.</param>
/// <param name="confirmationNumber">A <see cref="string"/> containing the payment confirmation number.</param>
/// <returns>
/// A <see cref="HttpResponseMessage"/> where true indicates the payment information was successfully inserted, otherwise false.
/// </returns>
/// <remarks></remarks>
0 references | Mark Troutwine, 21 days ago | 1 author, 1 change
public HttpResponseMessage CancelPaymentConfirmation(Guid azureObjectId, string confirmationNumber) {
    try {
        bool returnValue = new MemberPortalDataProvider().CancelPaymentConfirmation(azureObjectId, confirmationNumber);
        return Request.CreateResponse(HttpStatusCode.OK, returnValue);
    } catch (Exception ex) {
        string errorId = MemberPortalLogging.LogError(new StringBuilder(ErrorLoggingPrefix).Append(MethodBase.GetCurrentMethod().Name).ToString(), ex, TraceEventType.Error);
        return Request.CreateResponse(HttpStatusCode.InternalServerError, errorId);
    }
}
```