# How to Search for Calendar Items in a Shared Mailbox in Office 365 Exchange Online

## Introduction

Currently, Outlook Web App doesn't allow you to search for calendar items in a shared mailbox. But some of you require this feature for some reasons. In this sample, we will demonstrate how to search for calendar items in a shared mailbox:

1. Get the shared mailbox that users input

2. Get the search filter, such as the start date, end date, subject.

3. Set the ImpersonatedUserId property if the login account has the impersonation permission.

4. Search for the items in the shared mailbox.

## Running the Sample

Press F5 to run the sample.



If you want to search a shared mailbox, you should have impersonation permission.

First, you should input the shared mailbox which you want to search.



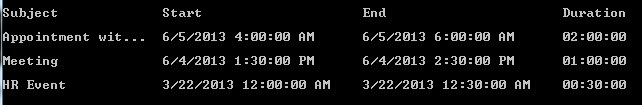
Then you can define the duration you want to search for.



At last, you can define the subject you want to search for, or you can just press Enter to get all the items.



The result will be shown as follows:



## Using the Code

If we have impersonation permission, we can search for items in a shared mailbox by setting ImpersonatedUserId property.

|  |
| --- |
| -Code block start-  --C# code snippet start--  // If the shared mailbox identity is valid, we will set it as the ImpersonatedUserId.  NameResolutionCollection nameResolutions =  service.ResolveName(identity, ResolveNameSearchLocation.DirectoryOnly, false);  if (nameResolutions.Count != 1)  {  Console.WriteLine("{0} is invalid Shared Mailbox identity.", identity);  Console.WriteLine();  }  else  {  String emailAddress = nameResolutions[0].Mailbox.Address;  Console.WriteLine();  Console.WriteLine("Please input the start date(15 days before today is the defined date.):");  String startDate = Console.ReadLine();  Console.WriteLine("Please input the end dateI(30 days after start date is the defined date.):");  String endDate = Console.ReadLine();  Console.WriteLine("Please input the subject that you want to search(Press Enter directly to get all the itmes):");  String searchSubject = Console.ReadLine();  Console.WriteLine();  service.ImpersonatedUserId =  new ImpersonatedUserId(ConnectingIdType.SmtpAddress, emailAddress);  GetSharedMailboxCalendarItems(service, emailAddress, searchSubject,  startDate, endDate);  }  --C# code snippet end--  --VB code snippet start--  ' If the shared mailbox identity is valid, we will set it as the ImpersonatedUserId.  Dim nameResolutions As NameResolutionCollection =  service.ResolveName(identity, ResolveNameSearchLocation.DirectoryOnly, False)  If nameResolutions.Count <> 1 Then  Console.WriteLine("{0} is invalid Shared Mailbox identity.", identity)  Console.WriteLine()  Else  Dim emailAddress As String = nameResolutions(0).Mailbox.Address  Console.WriteLine()  Console.WriteLine("Please input the start date(15 days before today is the defined date.):")  Dim startDate As String = Console.ReadLine()  Console.WriteLine("Please input the end dateI(30 days after start date is the defined date.):")  Dim endDate As String = Console.ReadLine()  Console.WriteLine("Please input the subject that you want to search(Press Enter directly to get all the itmes):")  Dim searchSubject As String = Console.ReadLine()  Console.WriteLine()  service.ImpersonatedUserId =  New ImpersonatedUserId(ConnectingIdType.SmtpAddress, emailAddress)  GetSharedMailboxCalendarItems(service, emailAddress, searchSubject,  startDate, endDate)  --VB code snippet end--  -Code block end- |

We search for items in the specified shared mailbox.

|  |
| --- |
| -Code block start-  --C# code snippet start--  static void GetSharedMailboxCalendarItems(ExchangeService service, String emailAddress,  String searchSubject, String startDate, String endDate)  {  // If the date is invaild, we will set 15 days before today as the start date.  DateTime startSearchDate;  startSearchDate =  DateTime.TryParse(startDate, out startSearchDate) ? startSearchDate : DateTime.Now.AddDays(-15);  // If the date is invaild, we will set 30 days after the start date as the end date.  DateTime endSearchDate;  endSearchDate =  DateTime.TryParse(endDate, out endSearchDate) && endSearchDate >= startSearchDate ?  endSearchDate : startSearchDate.AddDays(30);  FolderId folderId = new FolderId(WellKnownFolderName.Calendar);  SearchFilter.SearchFilterCollection searchFilterCollection =  new SearchFilter.SearchFilterCollection();  searchFilterCollection.LogicalOperator = LogicalOperator.And;  // If you want search the specified subject, you can define the filter; or you will get all  // the items that contain the Subject schema.  if (String.IsNullOrWhiteSpace(searchSubject))  {  SearchFilter searchFilter = new SearchFilter.Exists(AppointmentSchema.Subject);  searchFilterCollection.Add(searchFilter);  }  else  {  SearchFilter searchFilter =  new SearchFilter.ContainsSubstring(AppointmentSchema.Subject, searchSubject);  searchFilterCollection.Add(searchFilter);  }  SearchFilter startDateFilter =  new SearchFilter.IsGreaterThanOrEqualTo(AppointmentSchema.DateTimeCreated, startSearchDate);  SearchFilter endDateFilter =  new SearchFilter.IsLessThanOrEqualTo(AppointmentSchema.DateTimeCreated, endSearchDate);  searchFilterCollection.Add(startDateFilter);  searchFilterCollection.Add(endDateFilter);  ItemView itemView = new ItemView(100);  itemView.PropertySet = new PropertySet(BasePropertySet.FirstClassProperties);  FindItemsResults<Item> findItems=null;  Console.WriteLine("{0,-20}{1,-25}{2,-25}{3,-10}","Subject","Start","End","Duration");  do  {  findItems=service.FindItems(folderId,searchFilterCollection,itemView);  foreach (Item item in findItems)  {  Console.Write(  "{0,-20}",item.Subject.Length>18?item.Subject.Substring(0,15)+"...":item.Subject);    if (item is Appointment)  {  Appointment appointment = item as Appointment;  Console.Write("{0,-25}", appointment.Start);  Console.Write("{0,-25}", appointment.End);  Console.Write("{0,-10}", appointment.Duration);  }  Console.WriteLine();  }  }while(findItems.MoreAvailable);  Console.WriteLine();  }  --C# code snippet end--  --VB code snippet start--  Private Shared Sub GetSharedMailboxCalendarItems(ByVal service As ExchangeService,  ByVal emailAddress As String,  ByVal searchSubject As String,  ByVal startDate As String,  ByVal endDate As String)  ' If the date is invaild, we will set 15 days before today as the start date.  Dim startSearchDate As Date  startSearchDate = If(Date.TryParse(startDate, startSearchDate), startSearchDate,  Date.Now.AddDays(-15))  ' If the date is invaild, we will set 30 days after the start date as the end date.  Dim endSearchDate As Date  endSearchDate = If(Date.TryParse(endDate, endSearchDate) AndAlso  endSearchDate >= startSearchDate, endSearchDate,  startSearchDate.AddDays(30))  Dim folderId As New FolderId(WellKnownFolderName.Calendar)  Dim searchFilterCollection As New SearchFilter.SearchFilterCollection()  searchFilterCollection.LogicalOperator = LogicalOperator.And  ' If you want search the specified subject, you can define the filter; or you will get all  ' the items that contain the Subject schema.  If String.IsNullOrWhiteSpace(searchSubject) Then  Dim searchFilter As SearchFilter = New SearchFilter.Exists(AppointmentSchema.Subject)  searchFilterCollection.Add(searchFilter)  Else  Dim searchFilter As SearchFilter =  New SearchFilter.ContainsSubstring(AppointmentSchema.Subject, searchSubject)  searchFilterCollection.Add(searchFilter)  End If  Dim startDateFilter As SearchFilter =  New SearchFilter.IsGreaterThanOrEqualTo(AppointmentSchema.DateTimeCreated, startSearchDate)  Dim endDateFilter As SearchFilter =  New SearchFilter.IsLessThanOrEqualTo(AppointmentSchema.DateTimeCreated, endSearchDate)  searchFilterCollection.Add(startDateFilter)  searchFilterCollection.Add(endDateFilter)  Dim itemView As New ItemView(100)  itemView.PropertySet = New PropertySet(BasePropertySet.FirstClassProperties)  Dim findItems As FindItemsResults(Of Item) = Nothing  Console.WriteLine("{0,-20}{1,-25}{2,-25}{3,-10}", "Subject", "Start", "End", "Duration")  Do  findItems = service.FindItems(folderId, searchFilterCollection, itemView)  For Each item As Item In findItems  Console.Write("{0,-20}", If(item.Subject.Length > 18, item.Subject.Substring(0, 15) & "...", item.Subject))  If TypeOf item Is Appointment Then  Dim appointment As Appointment = TryCast(item, Appointment)  Console.Write("{0,-25}", appointment.Start)  Console.Write("{0,-25}", appointment.End)  Console.Write("{0,-10}", appointment.Duration)  End If  Console.WriteLine()  Next item  Loop While findItems.MoreAvailable  Console.WriteLine()  End Sub  --VB code snippet end--  -Code block end- |

## More Information

[EWS Managed API 2.0](http://msdn.microsoft.com/en-us/library/dd633709(v=exchg.80).aspx)

[Working with impersonation by using the EWS Managed API](http://msdn.microsoft.com/en-us/library/dd633680(v=exchg.80).aspx)