# How to Get Availability Details in Office 365

## Introduction

## Currently, Outlook Web App (OWA) allows you to check the availability by using Schedule Assistant. But you may want to have a list of events to track the availability of meeting rooms. In this application, we will demonstrate how to get availability details in Office 365.

## 1. You need input the email addresses and the duration from which you want to get the availability details.

## 2. The application will check the addresses and the date.

3. At last the application will show the result of the availability details.

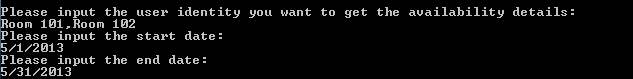
## Running the Sample

## Press F5 to run the sample.

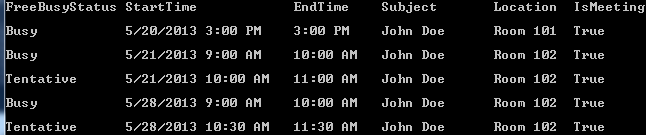
First, use your account to connect the Exchange Online.



Then you can input the identities and the duration for which you want to get the availability details.



At last, the results of availability will be shown as follows:



## Using the Code

Get the identities

|  |
| --- |
| -Code block start-  --C# code snippet start--  String[] identities = inputInfo.Split(',');  List<String> emailAddresses = new List<String>();  foreach (String identity in identities)  {  NameResolutionCollection nameResolutions =  service.ResolveName(identity, ResolveNameSearchLocation.DirectoryOnly, true);  if (nameResolutions.Count != 1)  {  Console.WriteLine("{0} is invalid user identity.", identity);  }  else  {  String emailAddress = nameResolutions[0].Mailbox.Address;  emailAddresses.Add(emailAddress);  }  }  if (emailAddresses.Count > 0)  {  GetAvailabilityDetails(service, startDate, endDate, emailAddresses.ToArray());  }  --C# code snippet end--  --VB code snippet start--  Dim identities() As String = inputInfo.Split(","c)  Dim emailAddresses As New List(Of String)()  For Each identity As String In identities  Dim nameResolutions As NameResolutionCollection =  service.ResolveName(identity, ResolveNameSearchLocation.DirectoryOnly, True)  If nameResolutions.Count <> 1 Then  Console.WriteLine("{0} is invalid user identity.", identity)  Else  Dim emailAddress As String = nameResolutions(0).Mailbox.Address  emailAddresses.Add(emailAddress)  End If  Next identity  If emailAddresses.Count > 0 Then  GetAvailabilityDetails(service, startDate, endDate, emailAddresses.ToArray())  End If  --VB code snippet end--  -Code block end- |

Set the duration and the attendees

|  |
| --- |
| -Code block start-  --C# code snippet start--  // If the date is invaild, we will set today as the start date.  DateTime startMeetingDate;  startMeetingDate=  DateTime.TryParse(startDate, out startMeetingDate)?startMeetingDate:DateTime.Now;  // If the date is invaild, we will set two days after the start date as the end date.  DateTime endMeetingDate;  endMeetingDate =  DateTime.TryParse(endDate, out endMeetingDate)&&endMeetingDate>=startMeetingDate ?  endMeetingDate : startMeetingDate.AddDays(2);  List<AttendeeInfo> attendees = new List<AttendeeInfo>();  foreach (String emailAddress in emailAddresses)  {  AttendeeInfo attendee = new AttendeeInfo(emailAddress);  attendees.Add(attendee);  }  TimeWindow timeWindow = new TimeWindow(startMeetingDate,endMeetingDate);  AvailabilityOptions availabilityOptions = new AvailabilityOptions();  availabilityOptions.MeetingDuration = 60;  --C# code snippet end--  --VB code snippet start--  ' If the date is invaild, we will set today as the start date.  Dim startMeetingDate As Date  startMeetingDate = If(Date.TryParse(startDate, startMeetingDate), startMeetingDate, Date.Now)  ' If the date is invaild, we will set two days after the start date as the end date.  Dim endMeetingDate As Date  endMeetingDate = If(Date.TryParse(endDate, endMeetingDate) AndAlso  endMeetingDate >= startMeetingDate, endMeetingDate,  startMeetingDate.AddDays(2))  Dim attendees As New List(Of AttendeeInfo)()  For Each emailAddress As String In emailAddresses  Dim attendee As New AttendeeInfo(emailAddress)  attendees.Add(attendee)  Next emailAddress  Dim timeWindow As New TimeWindow(startMeetingDate, endMeetingDate)  Dim availabilityOptions As New AvailabilityOptions()  availabilityOptions.MeetingDuration = 60  --VB code snippet end--  -Code block end- |

Get the availability results and display them

|  |
| --- |
| -Code block start-  --C# code snippet start--  GetUserAvailabilityResults userAvailabilityResults = service.GetUserAvailability(attendees,  timeWindow, AvailabilityData.FreeBusyAndSuggestions, availabilityOptions);  Console.WriteLine("{0,-15}{1,-21}{2,-11}{3,-14}{4,-10}{5,-9}", "FreeBusyStatus",  "StartTime", "EndTime", "Subject", "Location", "IsMeeting");  foreach (AttendeeAvailability userAvailabilityResult in  userAvailabilityResults.AttendeesAvailability)  {  if (userAvailabilityResult.ErrorCode.CompareTo(ServiceError.NoError) == 0)  {  foreach (CalendarEvent calendarEvent in userAvailabilityResult.CalendarEvents)  {  Console.WriteLine("{0,-15}{1,-21}{2,-11}{3,-14}{4,-10}{5,-9}",  calendarEvent.FreeBusyStatus,  calendarEvent.StartTime.ToShortDateString() + " " +  calendarEvent.StartTime.ToShortTimeString(),  calendarEvent.EndTime.ToShortTimeString(),  calendarEvent.Details.Subject,  calendarEvent.Details.Location,  calendarEvent.Details.IsMeeting);  }  }  }  --C# code snippet end--  --VB code snippet start--  Dim userAvailabilityResults As GetUserAvailabilityResults =  service.GetUserAvailability(attendees, timeWindow,  AvailabilityData.FreeBusyAndSuggestions, availabilityOptions)  Console.WriteLine("{0,-15}{1,-21}{2,-11}{3,-14}{4,-10}{5,-9}",  "FreeBusyStatus", "StartTime", "EndTime", "Subject", "Location", "IsMeeting")  For Each userAvailabilityResult As AttendeeAvailability In  userAvailabilityResults.AttendeesAvailability  If userAvailabilityResult.ErrorCode.CompareTo(ServiceError.NoError) = 0 Then  For Each calendarEvent As CalendarEvent In userAvailabilityResult.CalendarEvents  Console.WriteLine("{0,-15}{1,-21}{2,-11}{3,-14}{4,-10}{5,-9}",  calendarEvent.FreeBusyStatus,  calendarEvent.StartTime.ToShortDateString() & " " &  calendarEvent.StartTime.ToShortTimeString(),  calendarEvent.EndTime.ToShortTimeString(),  calendarEvent.Details.Subject,  calendarEvent.Details.Location,  calendarEvent.Details.IsMeeting)  Next calendarEvent  End If  Next userAvailabilityResult  --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)

## [Getting user free/busy information by using the EWS Managed API](http://msdn.microsoft.com/en-us/library/hh532567%28v=exchg.80%29.aspx)