

# Book a pool with the schedule assistant

Pool resources show as available in the schedule assistant if the system can distribute the planned work duration across multiple child resources. For example, if you want to schedule a task of one hour and no resource is available for the full hour, the system suggests two resources that are available for 30 minutes each.

Pool resources don't show as available in the schedule assistant if the pool is fully booked for the time period, even if the individual resource isn't booked for the period.

1. Select **Book** on a requirement form to open the schedule assistant. The schedule assistant shows the pool and individual pool members based on the configuration of the requirement.
2. Select the pool resource and a time slot and select **Book & Exit**.

## ⓘ Note

The schedule assistant won't return resource pools for onsite work requirements.

# Manage a pool on the schedule board

1. On the **Schedule Board**, open the **Filters** pane. Set the **Resource Types** filter to *Pool* and select **Apply**.
2. To set pool as the default view for the selected schedule board tab, select the ellipsis (...) and then **Save as default**.
3. Right-click the resource pool and select **View Group Members** to see the pool and pool members.

# Assign pool bookings

Assign specific resources to a requirement booked to a pool. Bookings assigned to pool member resources during non-working hours count against the capacity of the pool resource twice.

There are three ways to change a booking assignment:

1. **Manually drag and drop:** Select and drag a booking from the resource pool and drag it to one of the pool members.

2. **Substitution:** Right-click a booking in the resource pool, select **Book Substitute** and choose a pool member.
3. **Rebook:** Right-click a pool booking and select **Rebook**. Edit the filter criteria to no longer search for pools, but to search for Contact, User, and Account type resources.

The duration resets to the default booking duration. Edit the end time to match the original duration. Then, cancel the original pool booking.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) 

# Schedule a facility and associated resources

Article • 05/21/2024

Universal Resource Scheduling (URS) enables organizations to schedule interactions between customers and company resources. For scenarios where the customer is expected to travel to the company's location, the facility scheduling feature can coordinate physical spaces and related resources.

Typical examples include:

## Reserving a physical space

- Reserve a room for an event or party
- Reserve a room for an exercise class
- Reserve a bay at a mechanic shop
- Reserve a boat

## Reserving an appointment with a person at a facility

- Laptop repair at a Microsoft retail store
- Wealth management consultation at bank branch
- Doctor's office with related nurse and doctor

In this article, we explore a few example scenarios using facility scheduling.

## Prerequisites

- Universal Resource Scheduling (URS) v3.0
- Field Service v8.0 (for work orders, if applicable)

In general, to use facility scheduling, you create a facility resource, configure a requirement that calls for facility resources, and book the requirement.

## Scenario 1: Schedule a facility

In this scenario, schedulers want to search for nearby doctors' offices and schedule an appointment for a patient to arrive at the doctor's office.

We configure this scenario by creating a facility resource to represent the doctor's office, creating a requirement to represent the patient's request for an appointment at a

nearby facility, and then booking the appointment and viewing the facility and booking on the schedule board.

## 1. Create a facility resource

First, navigate to **Universal Resource Scheduling > Resources** and create a facility resource with the following attributes.

1. **Resource Type:** Facility.
2. **Start/end location:** Organizational Unit Address. The parent organization must have a latitude and longitude value that represents the location of the facility. This allows the schedule assistant to consider the facility's location when displaying available results.
3. **Display on Schedule Board = Yes**
4. **Enable for Availability Search = Yes**
5. (Optional) Enter working hours
6. (Optional) Add related characteristics, territory, resource roles, etc. to distinguish facility resources from other facilities and resources. For example, if a doctor's office has X-ray equipment, "X-Ray" can be added as a characteristic to the facility resource. This can influence which facility resources are filtered and displayed on the schedule board or which facility resources are returned during a schedule assistant search.

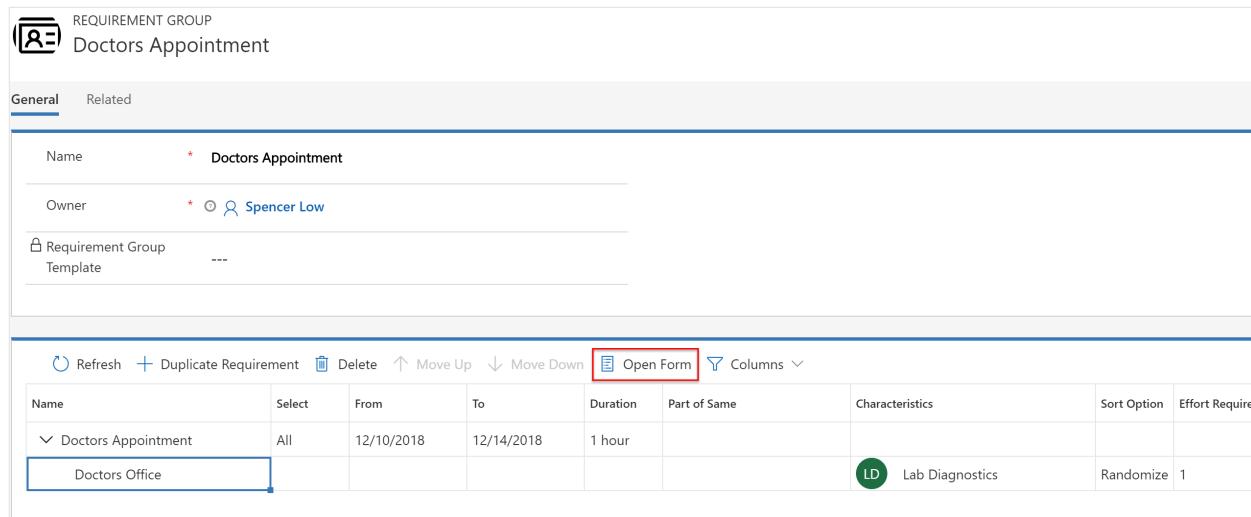
BOOKABLE RESOURCE Doctor's Office Seattle		ORGANIZATIONAL UNIT Seattle	
<b>General</b>	Project Service	<b>General</b>	Scheduling
Resource Type	Facility	Start Location	Organizational Unit Address
Name	Doctor's Office Seattle	End Location	Organizational Unit Address
Time Zone	(GMT-08:00) Pacific Time (US & Canada)	Scheduling	
		Display On Schedule Board	Enable for Availability Search
		Yes	Yes
		Latitude 47.70939	
		Longitude -122.31784	

## 2. Create a requirement for a facility

Next, create a requirement that calls for a facility resource.

Navigate to **Universal Resource Scheduling > Requirement Groups > +New**. Configure the following fields.

1. Enter a **Name**
2. Enter **From** and **To** dates
3. Set a **Duration**



REQUIREMENT GROUP  
Doctors Appointment

General Related

Name	* Doctors Appointment	
Owner	* <input checked="" type="radio"/> Spencer Low	
Requirement Group Template	---	

Refresh Duplicate Requirement Delete Move Up Move Down Open Form Columns

Name	Select	From	To	Duration	Part of Same	Characteristics	Sort Option	Effort Require
Doctors Appointment	All	12/10/2018	12/14/2018	1 hour				
Doctors Office						LD Lab Diagnostics	Randomize	1

4. Highlight the requirement and select **Open Form**, then set the **Resource Type** field to **Facility**, which accomplishes the following:
  - a. It filters schedule assistant results by only showing resources with the chosen types (User, Account, Contact, Equipment, Pool, Crew, or in this case, **Facility**).
  - b. This means the requirement requires a facility resource to be fulfilled and ensures the schedule assistant search results return facility type resources (as opposed to personnel or equipment).
  - c. It maps to the resource type field on the resource entity.
  - d. If this field is left empty, all resources are searched.



RESOURCE REQUIREMENT

Doctors Office

[General](#) [Project](#) [Field Service](#) [Scheduling](#) [Related](#)

Name	<a href="#">Doctors Office</a>
Owner	<a href="#">Spencer Low</a>
From Date	12/10/2018
To Date	12/14/2018
Duration	1 hour
Effort	1.0000
Status	<a href="#">Active</a>
Resource Type	Facility
Requirement Group	<a href="#">Doctors Appointme...</a>

## Skills

S-

Doctors Office  
Lab Diagnostics

## Roles



No data available

5. Set **Work Location** to Facility, which means:

- The interaction takes place at the scheduled facility and factors into travel time and distance calculations. Find more details in the configuration considerations section of this article.

6. Enter **latitude and longitude**.

- These values are typically entered manually or through workflows.
- These values represent the customer's location and are used to display facilities relative to the customer's location in schedule assistant results. This isn't the facility's location, as that is taken from the organizational unit of the facility type resource.

General   Project   Field Service   **Scheduling**   Related

Time Window Start	Time Window End
Time From Promised   --- <input type="button" value="..."/> <input type="button" value="Calendar"/> <input type="button" value="..."/> <input type="button" value="Clock"/>	Time To Promised   --- <input type="button" value="..."/> <input type="button" value="Calendar"/> <input type="button" value="..."/> <input type="button" value="Clock"/>
Priority   ---	Fulfillment Preference   ---
Work Location <b>Facility</b>	Territory   ---
Latitude   * <b>47.68716</b>	Longitude   * <b>-122.15224</b>
<input type="checkbox"/> Fulfilled Duration <b>0 minutes</b>	Is Primary <b>Yes</b>
<input type="checkbox"/> Remaining Duration <b>1 hour</b>	<input type="checkbox"/> Proposed Duration <b>0 minutes</b>

### 3. Book the requirement

After creating a facility resource and a requirement that calls for a facility, you can schedule the facility. Requirements that are part of a group can be scheduled via the **Book** button to trigger the schedule assistant, but not through drag and drop. Facility requirements not part of a group (single requirements) can be manually dragged and dropped to a facility on the schedule board or by using the schedule assistant.

The schedule assistant considers availability of resources and other set requirement constraints, such as characteristics, organizational units, categories, etc.

Select **Book** from the requirement or requirement group form, as seen in the following screenshot.

The screenshot shows the 'Schedule Assistant Filter' on the left and a results table on the right. The filter includes fields for Work Location (Facility), Available Duration (1 hour), Radius (50 mi), and Search Start/End dates/times. The results table lists 12 entries for 'Doctor's Office Seattle' with 'DS' icons, showing start and end times, travel time (~29 min), and distance (~19.26 miles) for each 30-minute interval from 9:00 AM to 4:00 PM on December 10, 2018.

Team	Team Photos	Start Time	End Time	Travel Time	Distance	Travel Start ...	Members	Requirement
Doctor's Office Seattle	DS	12/10 9:00 AM	12/10 10:00 AM	~29 min	~19.26 miles	~12/10 9:00 ...	1	---
Doctor's Office Seattle	DS	12/10 9:30 AM	12/10 10:30 AM	~29 min	~19.26 miles	~12/10 9:30 ...	1	---
Doctor's Office Seattle	DS	12/10 10:00 AM	12/10 11:00 AM	~29 min	~19.26 miles	~12/10 10:00...	1	---
Doctor's Office Seattle	DS	12/10 10:30 AM	12/10 11:30 AM	~29 min	~19.26 miles	~12/10 10:30...	1	---
Doctor's Office Seattle	DS	12/10 11:00 AM	12/10 12:00 PM	~29 min	~19.26 miles	~12/10 11:00...	1	---
Doctor's Office Seattle	DS	12/10 11:30 AM	12/10 12:30 PM	~29 min	~19.26 miles	~12/10 11:30...	1	---
Doctor's Office Seattle	DS	12/10 12:00 PM	12/10 1:00 PM	~29 min	~19.26 miles	~12/10 12:00...	1	---
Doctor's Office Seattle	DS	12/10 12:30 PM	12/10 1:30 PM	~29 min	~19.26 miles	~12/10 12:30...	1	---
Doctor's Office Seattle	DS	12/10 1:00 PM	12/10 2:00 PM	~29 min	~19.26 miles	~12/10 1:00 ...	1	---
Doctor's Office Seattle	DS	12/10 1:30 PM	12/10 2:30 PM	~29 min	~19.26 miles	~12/10 1:30 ...	1	---
Doctor's Office Seattle	DS	12/10 2:00 PM	12/10 3:00 PM	~29 min	~19.26 miles	~12/10 2:00 ...	1	---
Doctor's Office Seattle	DS	12/10 2:30 PM	12/10 3:30 PM	~29 min	~19.26 miles	~12/10 2:30 ...	1	---
Doctor's Office Seattle	DS	12/10 3:00 PM	12/10 4:00 PM	~29 min	~19.26 miles	~12/10 3:00 ...	1	---

Facility travel time and distance calculations in the preceding schedule assistant example represent the time and distance between the facility resource (as defined by the location of the related organizational unit) and the customer's location (as defined by the latitude and longitude values on the requirement). The schedule assistant's radius filter filters based on this travel calculation.

The screenshot shows the 'Schedule Assistant Filter' on the left and a results table on the right. The 'Work Location' dropdown is set to 'Location Agnostic'. The results table shows the same 12 entries as the first screenshot, but the 'Travel Time' and 'Distance' columns are now both filled with '---' for all rows, indicating that travel calculations are disabled when the work location is set to 'Location Agnostic'.

Team	Team Photos	Start Time	End Time	Travel Time	Distance	Travel Start Time	Members	Requirement
Doctor's Office Seattle	DS	12/10 9:00 AM	12/10 10:00 AM	---	---	~12/10 9:00 AM	1	---
Doctor's Office Seattle	DS	12/10 9:30 AM	12/10 10:30 AM	---	---	~12/10 9:30 AM	1	---
Doctor's Office Seattle	DS	12/10 10:00 AM	12/10 11:00 AM	---	---	~12/10 10:00 AM	1	---
Doctor's Office Seattle	DS	12/10 10:30 AM	12/10 11:30 AM	---	---	~12/10 10:30 AM	1	---
Doctor's Office Seattle	DS	12/10 11:00 AM	12/10 12:00 PM	---	---	~12/10 11:00 AM	1	---
Doctor's Office Seattle	DS	12/10 11:30 AM	12/10 12:30 PM	---	---	~12/10 11:30 AM	1	---
Doctor's Office Seattle	DS	12/10 12:00 PM	12/10 1:00 PM	---	---	~12/10 12:00 PM	1	---
Doctor's Office Seattle	DS	12/10 12:30 PM	12/10 1:30 PM	---	---	~12/10 12:30 PM	1	---
Doctor's Office Seattle	DS	12/10 1:00 PM	12/10 2:00 PM	---	---	~12/10 1:00 PM	1	---
Doctor's Office Seattle	DS	12/10 1:30 PM	12/10 2:30 PM	---	---	~12/10 1:30 PM	1	---
Doctor's Office Seattle	DS	12/10 2:00 PM	12/10 3:00 PM	---	---	~12/10 2:00 PM	1	---
Doctor's Office Seattle	DS	12/10 2:30 PM	12/10 3:30 PM	---	---	~12/10 2:30 PM	1	---
Doctor's Office Seattle	DS	12/10 3:00 PM	12/10 4:00 PM	---	---	~12/10 3:00 PM	1	---

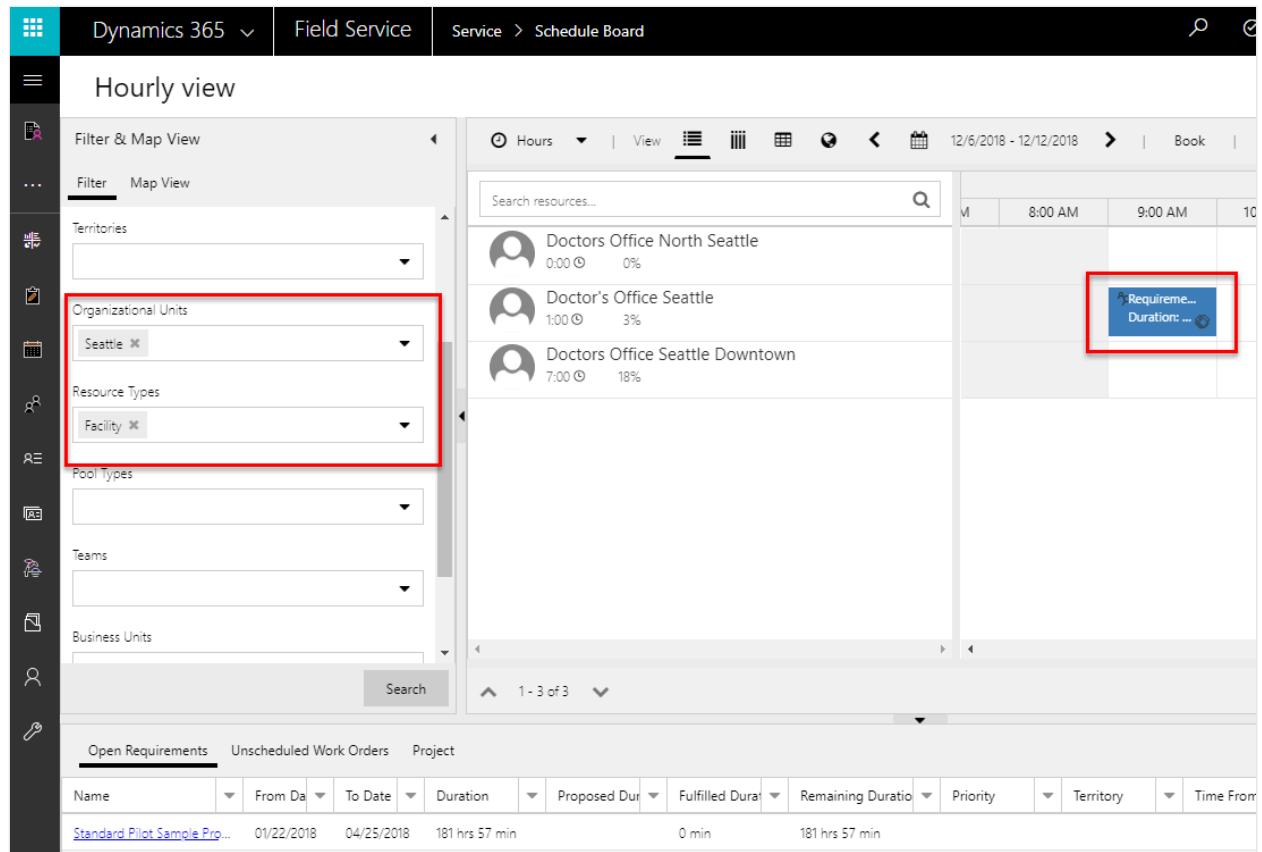
Filtering work location to location agnostic removes travel time and distance calculations from schedule assistant results.

## 4. Add the facility resource to the schedule board

Scheduling a facility is just like scheduling any other resource. You can drag and drop requirements to create bookings; you can drag existing bookings to change the time or

resource; or you can use the schedule assistant to help sift through the list of facilities based on availability and other constraints.

To view facility resources on the schedule board, filter by resource type and/or organizational units.



The screenshot shows the Dynamics 365 Field Service Schedule Board. The left sidebar has a 'Filter & Map View' section with dropdowns for Territories, Organizational Units (set to 'Seattle'), Resource Types (set to 'Facility'), and Pool Types. The main area shows a grid of resources with columns for Name, From Date, To Date, Duration, Proposed Duration, Fulfilled Duration, Remaining Duration, Priority, Territory, and Time From. Three resources are listed: 'Doctors Office North Seattle' (0.00 hours, 0%), 'Doctor's Office Seattle' (1:00 hours, 3%), and 'Doctors Office Seattle Downtown' (7:00 hours, 18%). A red box highlights the 'Organizational Units' and 'Resource Types' filters. Another red box highlights a blue button in the top right corner of the grid area, which says 'Requirement...' and 'Duration: ...'.

Facility resources will also be displayed on the schedule board map based on the location of the related organizational unit.

Name	From Date	To Date	Duration	Proposed Dur.	Fulfilled Dur.	Remaining Dur.	Priority	Territory	Time From Promis
Standard Pilot Sample Pro...	01/22/2018	04/25/2018	181 hrs 57 min	0 min		181 hrs 57 min			
Standard Pilot Sample Pro...	02/09/2018	04/11/2018	240 hrs	0 min		240 hrs			

## Scenario 2: Schedule a facility with 5 generic rooms

In this scenario, a doctor's office has 5 identical rooms and schedulers don't need to book each room specifically. They must, however, ensure that no more than 5 patients are booked across all rooms during any one time slot.

We'll configure this scenario by adding a **capacity** to a facility resource.

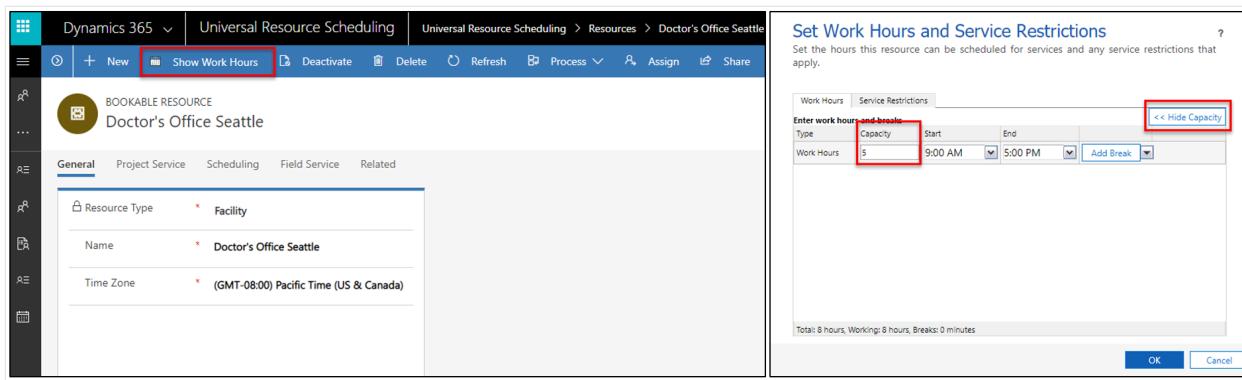
### 1. Create a facility resource

See scenario 1 in this article to create a facility resource, or select an existing facility resource record.

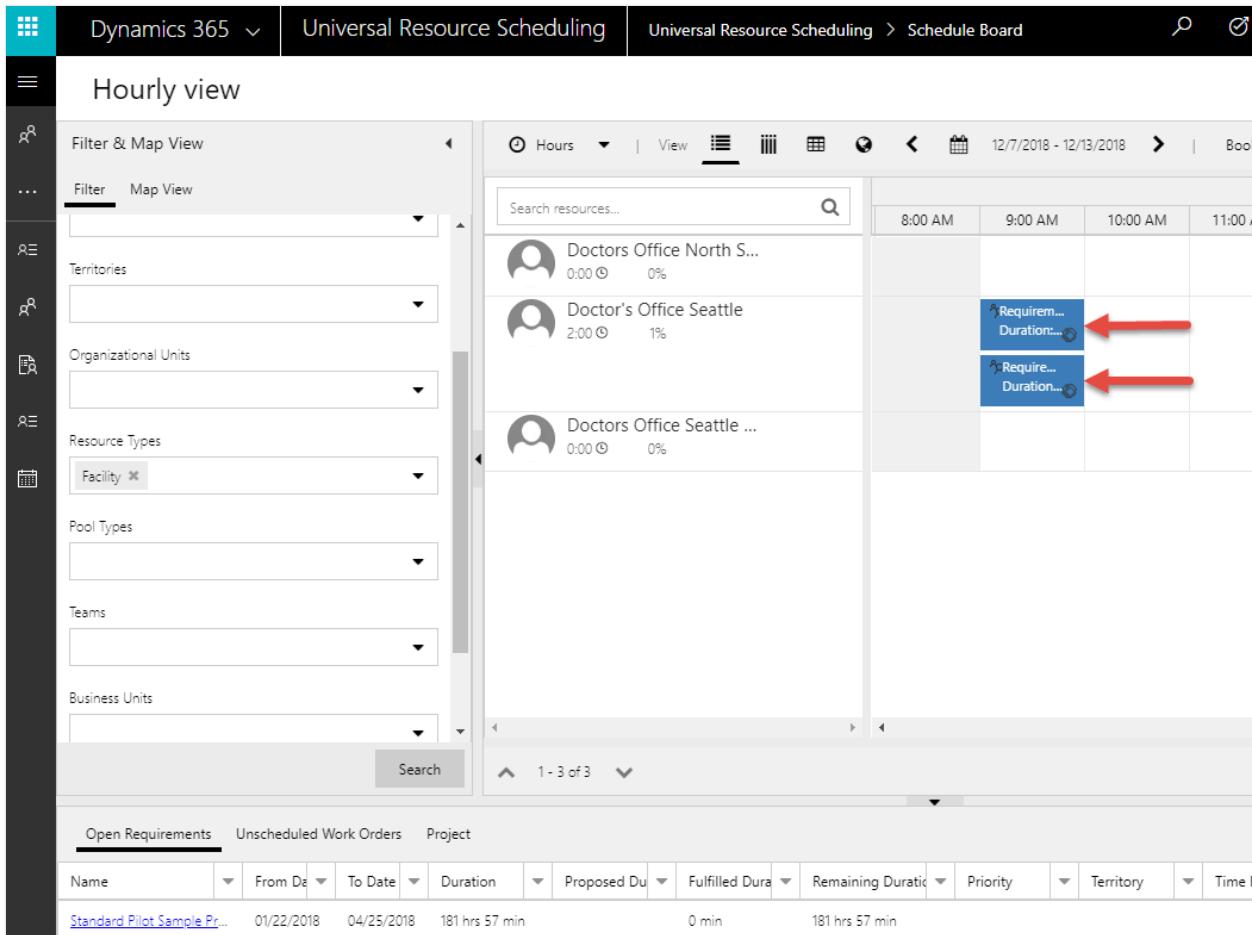
From the facility resource record, select **Show Work Hours** in the top ribbon.

### 2. Set the capacity of the facility resource

When choosing hours, select **Show Capacity** and enter 5. By default, the capacity is set to 1.



Setting the capacity to 5 means that when booking a facility with the schedule assistant, the facility resource shows as available and can be double-booked up to the capacity limit (in this case, 5 times).



In the preceding screenshot, two separate requirements for a facility were both scheduled to the same facility during the same time slot. Without increasing the capacity, once a timeslot at a facility is booked, it will no longer show in schedule assistant results.

### (!) Note

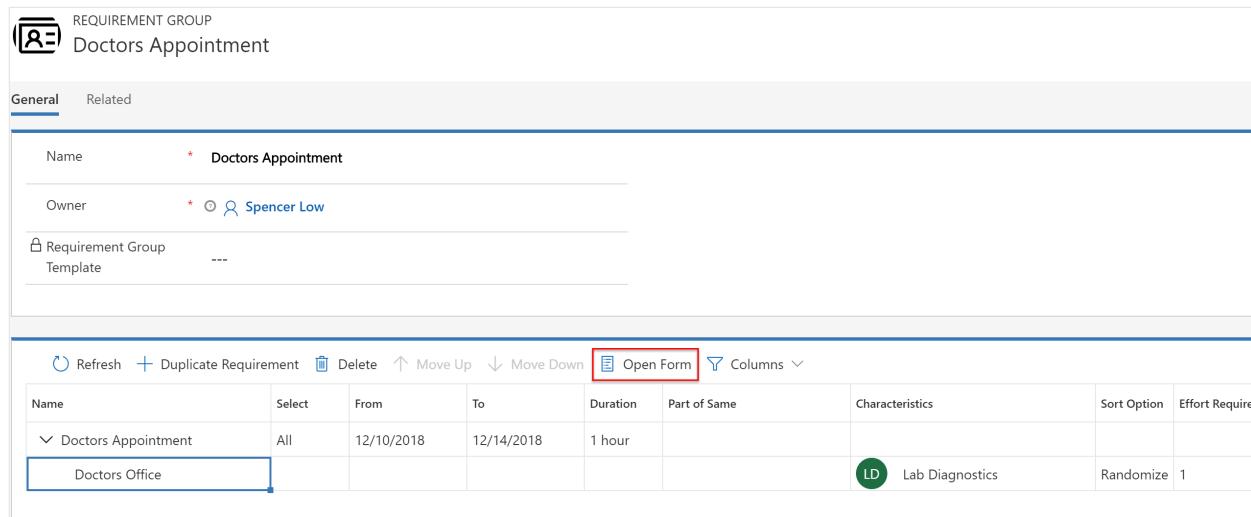
Capacity scheduling is not intended for booking the same requirement multiple times, but rather to book multiple requirements. Rebooking a previously booked requirement will cancel the existing booking and create a new one.

### 3. Create a requirement for a facility

Next, create a requirement that calls for a facility resource.

Navigate to **Universal Resource Scheduling > Requirement Groups > +New**. Configure the following fields.

1. Enter a **Name**
2. Enter **From** and **To** dates
3. Set a **Duration**



Name	Select	From	To	Duration	Part of Same	Characteristics	Sort Option	Effort Required
Doctors Appointment	All	12/10/2018	12/14/2018	1 hour		LD Lab Diagnostics	Randomize	1
Doctors Office								

4. Highlight the requirement and select **Open Form**, then set the **Resource Type** field to **Facility**, which accomplishes the following:
  - a. It filters schedule assistant results by only showing resources with the chosen types (User, Account, Contact, Equipment, Pool, Crew, or in this case, **Facility**).
  - b. This means the requirement requires a facility resource to be fulfilled and ensures the schedule assistant search results return facility type resources (as opposed to personnel or equipment).
  - c. It maps to the resource type field on the resource entity.
  - d. If this field is left empty, all resources are searched.



RESOURCE REQUIREMENT

Doctors Office

[General](#) [Project](#) [Field Service](#) [Scheduling](#) [Related](#)

Name	<a href="#">Doctors Office</a>
Owner	<a href="#">Spencer Low</a>
From Date	12/10/2018
To Date	12/14/2018
Duration	1 hour
Effort	1.0000
Status	<a href="#">Active</a>
Resource Type	Facility
Requirement Group	<a href="#">Doctors Appointme...</a>

## Skills

S-

Doctors Office  
Lab Diagnostics

## Roles



No data available

5. Set **Work Location** to Facility, which means:

- The interaction takes place at the scheduled facility and factors into travel time and distance calculations. Find more details in the configuration considerations section of this article.

6. Enter **latitude and longitude**.

- These values are typically entered manually or through workflows.
- These values represent the customer's location and are used to display facilities relative to the customer's location in schedule assistant results. This isn't the facility's location, as that is taken from the organizational unit of the facility type resource.

General   Project   Field Service   **Scheduling**   Related

Time Window Start	Time Window End				
Time From Promised	---	---	Time To Promised	---	---
Priority	---		Fulfillment Preference	---	
Work Location	Facility		Territory	---	
Latitude	* 47.68716		Longitude	* -122.15224	
Fulfilled Duration	0 minutes		Is Primary	Yes	
Remaining Duration	1 hour		Proposed Duration	0 minutes	

## Scenario 3: Schedule a facility and related resource

In this scenario, schedulers want to schedule a doctor's office and a related doctor at the same facility during the same timeslot.

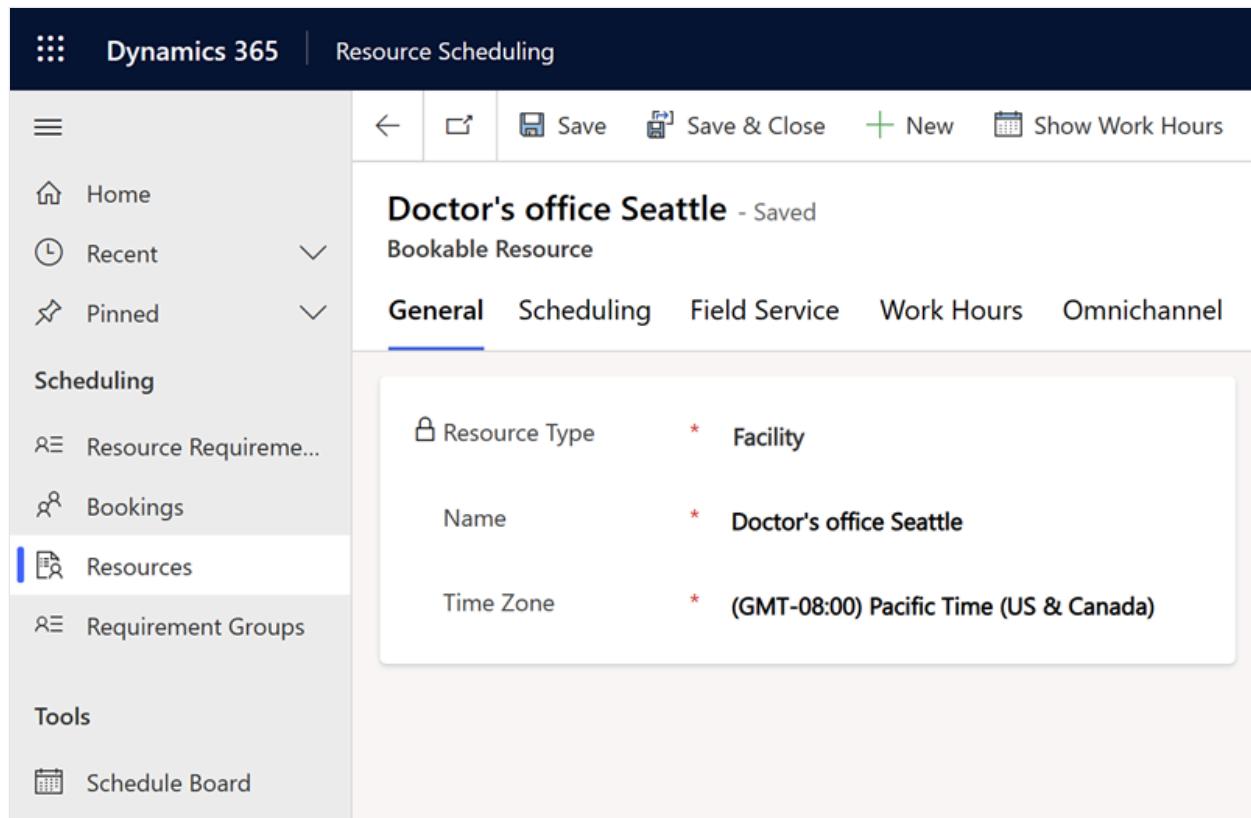
This scenario is configured by creating a facility resource, creating a doctor resource (a resource with resource type = user/contact/account), associating the doctor resource to the facility resource, and then creating a requirement group that calls for both a facility and a doctor.

In order to schedule groups of resources to perform a task together at a facility, non-facility resources can be associated to facility/facility pool resources through the **Resource Associations entity (msdyn\_bookableresourceassociations)**.

Resources such as people, equipment, or pool resources may be associated to a facility or facility pool with date effectivity. This means resources perform work at the facility location during the expressed date range, and they aren't eligible for "onsite" work for which they would have to leave the facility and travel to a customer's location. This is important as it relates to using the option **Same Resource Tree**.

### 1. Create a facility resource

First, create a resource to represent the doctor's office. The resource type should be set to **Facility**. The following screenshot shows an example of a facility resource.



## 2. Create a doctor resource

Create a resource to represent a doctor. Set the resource type to **User, Account, or Contact**, based on your business needs.

### ⓘ Note

The user resource type is typically designated for employees who access Dynamics 365 data; contact and account resource types are typically for contractors who need to be scheduled but don't access data.

Start/end location should be set to **Organizational Unit Address** and the organizational unit should be set to the same organizational unit of the facility resource (in this case, the doctor's office). This is **recommended but not required**.

## 3. Associate the doctor resource to the facility resource

From the facility resource, navigate to **Related > Bookable Resource Association (Resource 2)**. This related entity is called **Resource Associations (msdyn\_bookableresourceassociations)**.

From here, associate the resource that represents the doctor.

In the following screenshot, Abraham McCormick represents a doctor and is associated to "Doctors Office North Seattle," which represents the doctor's office.

**A. McCormick MD in Seattle** - Saved  
Bookable Resource Association

**General** **Related** ▾

Name	* A. McCormick MD in Seattle		
Owner	*  MOD Administrator (Offline)		
Resource 1	*  Abraham McCormick MD		
Resource 2	*  Doctor's office Seattle		
From Date	* 2/15/2023  8:00 AM 		
To Date	* 1/29/2025  8:00 AM 		

## 4. Create a requirement group

Next, create a requirement group with one requirement that calls for a doctor's office (Resource Type = Facility), and another requirement that calls for a doctor (Resource Type = User/Contact/Account).

**Requirement Group**  
 Doctors appointment with doctor

**General** **Related**

Name	* Doctors appointment with doctor		
Owner	*  Spencer Low		
Requirement Group Template	---		

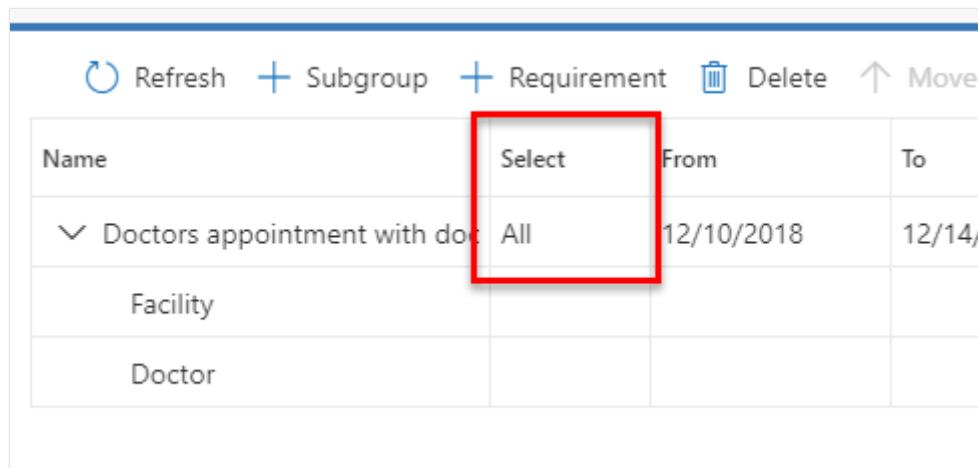
 Refresh  Subgroup  Requirement  Delete  Move Up  Move Down  Columns

Name	Select	From	To	Duration	Part of Same	Organizational Unit	Characteristics	Sort Option	Effort Required
Doctors appointment with doc	All	12/10/2018	12/14/2018	1 hour		<input type="checkbox"/> Organizational Unit	 LD Lab Diagnostics	Randomize	1
Facility						<input type="checkbox"/> Resource Tree	 PE Pediatrics	Randomize	1
Doctor						<input type="checkbox"/> Location			

Set the **Work Location** on each requirement to **Facility**, indicating the work takes place at the doctor's office.

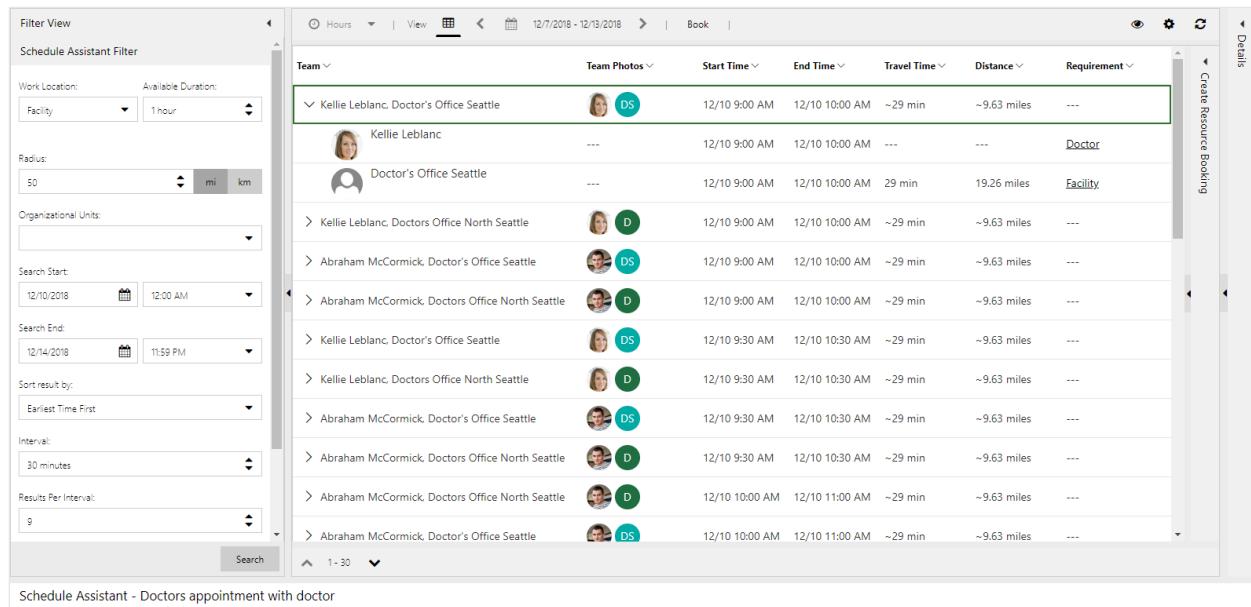
### ① Note

Using the **Select > All** option in the requirement group means that both requirements need to be fulfilled.



Name	Select	From	To
Doctors appointment with doc	All	12/10/2018	12/14/
Facility			
Doctor			

Set **Part of Same to Resource Tree**. This ensures resources from different locations aren't recommended for work taking place at a facility. As an example, a doctor associated with Facility B shouldn't be paired with Facility A. Find more details about this setting in the configuration considerations section of this article.



Filter View

Schedule Assistant Filter

Work Location: Facility Available Duration: 1 hour

Radius: 50 mi

Organizational Units:

Search Start: 12/10/2018 12:00 AM

Search End: 12/14/2018 11:59 PM

Sort result by: Earliest Time First

Interval: 30 minutes

Results Per Interval: 9

Team

Team	Team Photos	Start Time	End Time	Travel Time	Distance	Requirement
Kellie Leblanc, Doctor's Office Seattle		12/10 9:00 AM	12/10 10:00 AM	~29 min	~9.63 miles	---
Kellie Leblanc		---	12/10 9:00 AM	12/10 10:00 AM	---	---
Doctor's Office Seattle		---	12/10 9:00 AM	12/10 10:00 AM	29 min	19.26 miles
> Kellie Leblanc, Doctors Office North Seattle		12/10 9:00 AM	12/10 10:00 AM	~29 min	~9.63 miles	---
> Abraham McCormick, Doctor's Office Seattle		12/10 9:00 AM	12/10 10:00 AM	~29 min	~9.63 miles	---
> Abraham McCormick, Doctors Office North Seattle		12/10 9:00 AM	12/10 10:00 AM	~29 min	~9.63 miles	---
> Kellie Leblanc, Doctor's Office Seattle		12/10 9:30 AM	12/10 10:30 AM	~29 min	~9.63 miles	---
> Kellie Leblanc, Doctors Office North Seattle		12/10 9:30 AM	12/10 10:30 AM	~29 min	~9.63 miles	---
> Abraham McCormick, Doctor's Office Seattle		12/10 9:30 AM	12/10 10:30 AM	~29 min	~9.63 miles	---
> Abraham McCormick, Doctors Office North Seattle		12/10 10:00 AM	12/10 11:00 AM	~29 min	~9.63 miles	---
> Abraham McCormick, Doctor's Office Seattle		12/10 10:00 AM	12/10 11:00 AM	~29 min	~9.63 miles	---

Scheduling the requirement group creates a booking for the facility resource and the doctor resource.

The screenshot shows a software interface for managing resources and scheduling. On the left, there is a list of resources under the heading 'Search resources...'. The resources listed are 'Doctor's Office Seattle' (1:00, 1%) and 'Kellie Leblanc' (1:00, 3%). On the right, a calendar view shows time slots from 8:00 AM to 10:00 AM. Two specific time slots are highlighted in blue: 8:00 AM and 9:00 AM. Each highlighted slot contains a button labeled 'Require...' and 'Duration...' with a small circular icon.

#### ! Note

As in scenario 1, travel time and distance are calculated as the time and distance for the customer to travel to the facility. There is no travel calculation considered for the doctor resource, as it's assumed they will be at the facility at the required time.

## Scenario 4: Schedule a facility with 5 specific rooms

In this scenario, schedulers want to track all rooms within a doctor's office and schedule each individual room to patients.

This scenario is configured by creating a pool of facilities to represent the doctor's office and each individual room.

### 1. Create a facility pool resource

First, create a resource to represent the overall doctor's office where **Resource Type = Pool** and **Pool Type = Facility**.

In this example, we call it "Health Clinic," as seen in the following screenshot.

**BOOKABLE RESOURCE**  
Health Clinic

**General** Project Service Scheduling Field Service Related

Resource Type	*	Pool
Pool Type	*	Facility
Name	*	Health Clinic
Time Zone	*	(GMT-08:00) Pacific Time (US &

**BOOKABLE RESOURCE**  
Health Clinic

General Project Service **Scheduling** Field Service Related

Start Location	*	Organizational Unit Address
End Location	*	Organizational Unit Address
		Organizational Unit <input checked="" type="checkbox"/> Redmond
		Derive Capacity From Group Members <input checked="" type="checkbox"/> Yes

**Scheduling**

Display On Schedule Board	*	Yes
		Enable for Availability Search <input checked="" type="checkbox"/> Yes

## 2. Create facility resources to represent each room

Next, create multiple facility resources to represent each room.

Set **Resource Type** to **Facility** on each resource.

Set the **Start/End Location** to **Organizational Unit Address** and select an organizational unit to represent the location of the rooms.

**Universal Resource Scheduling** > Resources >

**Dynamics 365** Universal Resource Scheduling

New Show Work Hours Deactivate Delete Refresh Process ...

**BOOKABLE RESOURCE**  
Room 101

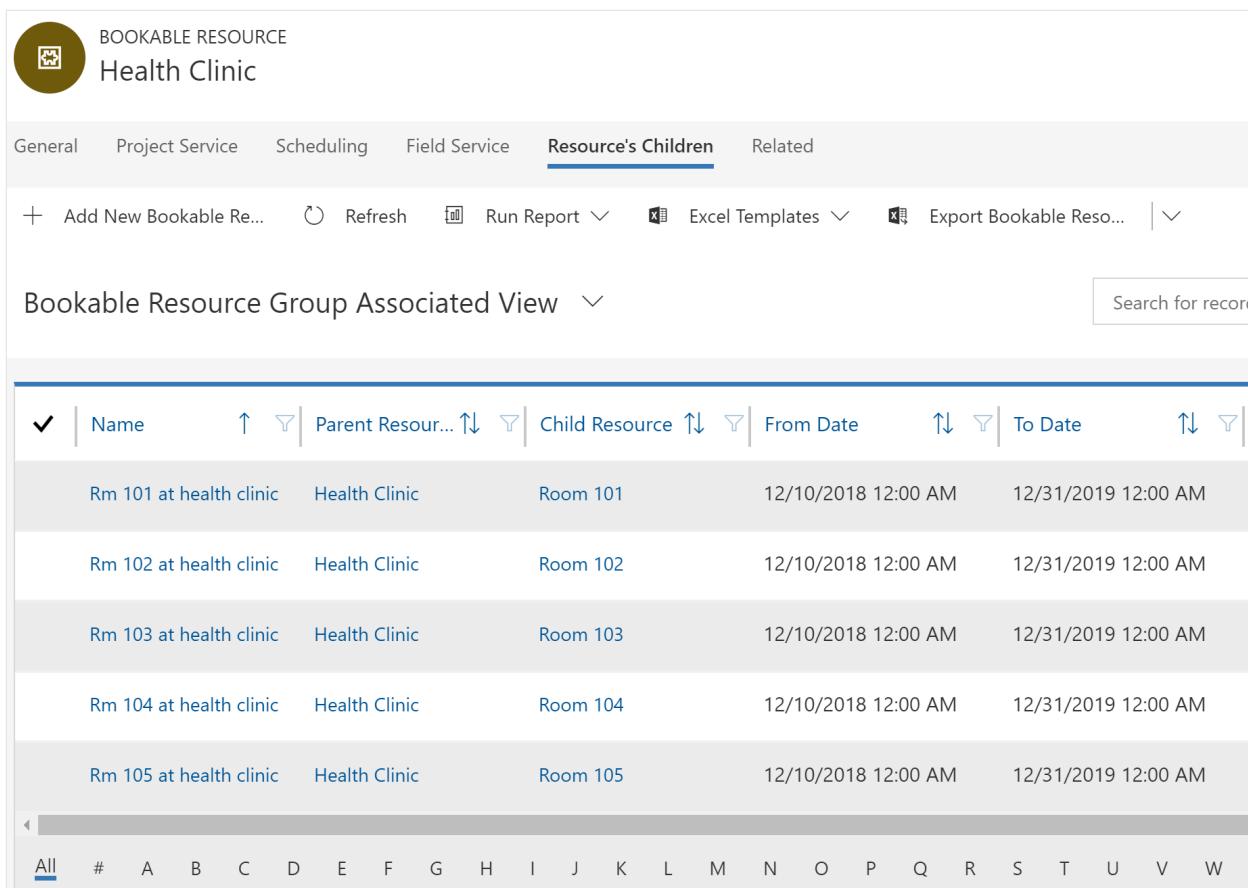
**General** Project Service Scheduling Field Service Related

Resource Type	*	Facility
Name	*	Room 101
Time Zone	*	(GMT-08:00) Pacific Time (US &

## 3. Add each room resource as a child resource to the doctor's office resource pool

Navigate to the doctor's office facility pool resource, and go to **Related > Resource Children**.

Add each room resource as a child record to the parent facility pool (health clinic), as seen in the following screenshot.



BOOKABLE RESOURCE  
Health Clinic

General Project Service Scheduling Field Service Resource's Children Related

+ Add New Bookable Re... Refresh Run Report Excel Templates Export Bookable Reso... |

Bookable Resource Group Associated View Search for record

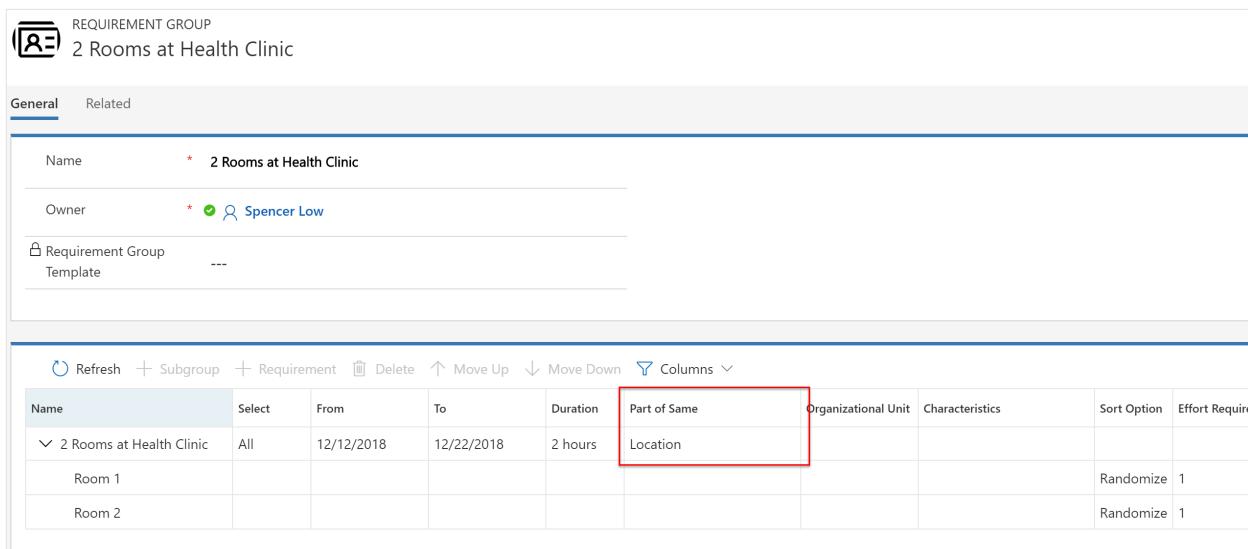
✓	Name	Parent Resour...	Child Resource	From Date	To Date
	Rm 101 at health clinic	Health Clinic	Room 101	12/10/2018 12:00 AM	12/31/2019 12:00 AM
	Rm 102 at health clinic	Health Clinic	Room 102	12/10/2018 12:00 AM	12/31/2019 12:00 AM
	Rm 103 at health clinic	Health Clinic	Room 103	12/10/2018 12:00 AM	12/31/2019 12:00 AM
	Rm 104 at health clinic	Health Clinic	Room 104	12/10/2018 12:00 AM	12/31/2019 12:00 AM
	Rm 105 at health clinic	Health Clinic	Room 105	12/10/2018 12:00 AM	12/31/2019 12:00 AM

All # A B C D E F G H I J K L M N O P Q R S T U V W

## 4. Create a requirement group

Create a requirement that calls for one or more facilities. In this example, we're looking for 2 specific rooms within the same doctor's office (health clinic).

Set **Part of Same to Same Location** to ensure each room is at the same physical address.



REQUIREMENT GROUP  
2 Rooms at Health Clinic

General Related

Name \* 2 Rooms at Health Clinic

Owner \*  Spencer Low

Requirement Group Template ---

Refresh Subgroup Requirement Delete Move Up Move Down Columns

Name	Select	From	To	Duration	Part of Same	Organizational Unit	Characteristics	Sort Option	Effort Require
2 Rooms at Health Clinic	All	12/12/2018	12/22/2018	2 hours	Location				
Room 1								Randomize	1
Room 2								Randomize	1

Each requirement should call for facility resource types, as seen in the following screenshot.

RESOURCE REQUIREMENT

Room 1 

General Project Field Service Scheduling Related

Name	Room 1
Owner	*  Spencer Low
From Date	12/12/2018 
To Date	12/22/2018 
Duration	2 hours
Effort	1.0000
Status	*  Active
Resource Type	Facility 
Requirement Group	 2 Rooms at Health Clinic

Skills   
No data available.

Roles   
No data available.

Again, the **Work Location** of each requirement should be set to **Facility** and the latitude and longitude fields of the requirements should correspond to the customer's (or patient's) location.

 **Note**

Latitude and longitude fields on all requirements within a group must be equal; updating the values on one requirement will update the others.

## 5. Book the requirement group

Select **Book** from the requirement group to trigger the schedule assistant.

Team	Team Photos	Start Time	End Time	Distance	Travel Start Ti...	Members	Excess M...	Crew/Pool	Requirement
Redmond: 24 min (14.91 miles)									
Room 105, Room 102	R1 R2	12/14 6:00 AM	12/14 8:00 AM	~17.40 miles	~12/14 6:00 AM	2	0	---	---
Room 105		---	12/14 6:00 AM	12/14 8:00 AM	17.40 miles	12/14 6:00 AM	---	---	Health Clinic Room 2
Room 102		---	12/14 6:00 AM	12/14 8:00 AM	17.40 miles	12/14 6:00 AM	---	---	Health Clinic Room 1
Room 104, Room 102	R1 R2	12/14 6:30 AM	12/14 8:30 AM	~17.40 miles	~12/14 6:30 AM	2	0	---	---
Room 101, Room 102	R1 R2	12/14 7:00 AM	12/14 9:00 AM	~17.40 miles	~12/14 7:00 AM	2	0	---	---
Room 102, Room 105	R1 R2	12/14 7:30 AM	12/14 9:30 AM	~17.40 miles	~12/14 7:30 AM	2	0	---	---
Room 103, Room 102	R1 R2	12/14 8:00 AM	12/14 10:00 ...	~17.40 miles	~12/14 8:00 AM	2	0	---	---
Room 103, Room 101	R1 R2	12/14 8:30 AM	12/14 10:30 ...	~17.40 miles	~12/14 8:30 AM	2	0	---	---
Room 105, Room 102	R1 R2	12/14 9:00 AM	12/14 11:00 ...	~17.40 miles	~12/14 9:00 AM	2	0	---	---
Room 104, Room 101	R1 R2	12/14 9:30 AM	12/14 11:30 ...	~17.40 miles	~12/14 9:30 AM	2	0	---	---
Room 103, Room 101	R1 R2	12/14 10:00 ...	12/14 12:00 ...	~17.40 miles	~12/14 10:00 ...	2	0	---	---
Room 105, Room 104	R1 R2	12/14 10:30 ...	12/14 12:30 ...	~17.40 miles	~12/14 10:30 ...	2	0	---	---
Room 104, Room 101	R1 R2	12/14 11:00 ...	12/14 1:00 PM	~17.40 miles	~12/14 11:00 ...	2	0	---	---
Room 102, Room 101	R1 R2	12/14 11:30 ...	12/14 1:30 PM	~17.40 miles	~12/14 11:30 ...	2	0	---	---

Schedule Assistant - 2 Rooms at Health Clinic

In the preceding screenshot's results, two specific rooms are recommended at the same location. The travel time and distance is calculated from the customer's location (latitude and longitude values on the requirement records) and the location of the facility resources (resource children organizational units).

## Scenario 5: Schedule a facility with 5 specific rooms and 5 related resources

In this scenario, schedulers want to schedule specific rooms within a doctor's office to a pool of available pediatric doctors who work at the health clinic.

This scenario is configured by creating a pool of facilities and a pool of doctors, and associating them together with **Resource Associations** (`msdyn_bookableresourceassociations`).

### 1. Create a facility pool

Using the same process we used in scenario 4, create a facility pool resource to represent the doctor's office.

**BOOKABLE RESOURCE**  
Health Clinic

General Project Service Scheduling Field Service Related

Resource Type	* Pool
Pool Type	* Facility
Name	* Health Clinic
Time Zone	* (GMT-08:00) Pacific Time (US &

**BOOKABLE RESOURCE**  
Health Clinic

General Project Service **Scheduling** Field Service Related

Start Location	* Organizational Unit Address	Organizational Unit <input checked="" type="checkbox"/> Redmond
End Location	* Organizational Unit Address	Derive Capacity From Group Members <input checked="" type="checkbox"/> Yes
<b>Scheduling</b>		
Display On Schedule Board	* Yes	Enable for Availability Search <input checked="" type="checkbox"/> Yes

## 2. Create facility resources for each room

Next, create facility resources to represent each room in the doctor's office facility pool.

Then add each room as a resource child to the doctor's office (health clinic) facility pool resource. The organizational unit of the office facility pool and the room facilities should be the same.

**BOOKABLE RESOURCE**  
Health Clinic

General Project Service Scheduling Field Service **Resource's Children** Related

+ Add New Bookable Re... Refresh Run Report Excel Templates Export Bookable Reso... |

Bookable Resource Group Associated View | Search for record

<input checked="" type="checkbox"/>	Name	Parent Resour...	Child Resource	From Date	To Date
	Rm 101 at health clinic	Health Clinic	Room 101	12/10/2018 12:00 AM	12/31/2019 12:00 AM
	Rm 102 at health clinic	Health Clinic	Room 102	12/10/2018 12:00 AM	12/31/2019 12:00 AM
	Rm 103 at health clinic	Health Clinic	Room 103	12/10/2018 12:00 AM	12/31/2019 12:00 AM
	Rm 104 at health clinic	Health Clinic	Room 104	12/10/2018 12:00 AM	12/31/2019 12:00 AM
	Rm 105 at health clinic	Health Clinic	Room 105	12/10/2018 12:00 AM	12/31/2019 12:00 AM

## 3. Create a pool of pediatric doctors

Create a new resource pool to represent the pediatric doctors.

Set Resource Type to Pool and set Pool Type to Contacts, Users, Accounts, as doctors are personnel.

As is true of all resource records, you can add characteristics to define and distinguish differences among resources. In this example, "pediatrics" could be a skill to add to a doctor resource.

Set Derive Capacity from Group Members to Yes. This means the capacity of the pool is based on how many doctors are associated to it.

 <p>BOOKABLE RESOURCE Pediatricians</p> <p>General Project Service Scheduling Field Service Related</p> <p>Resource Type * Pool</p> <p>Pool Type * Account, Contact, User</p> <p>Name * Pediatricians</p> <p>Time Zone * (GMT-08:00) Pacific Time (US &amp;</p>	 <p>BOOKABLE RESOURCE Pediatricians</p> <p>General Project Service <b>Scheduling</b> Field Service Related</p> <p>Start Location * Organizational Unit Address Organizational Unit <b>Redmond</b></p> <p>End Location * Organizational Unit Address Derive Capacity From Group Members <b>Yes</b></p> <p><b>Scheduling</b></p> <p>Display On Schedule Board * Yes Enable for Availability Search Yes</p>
--	---

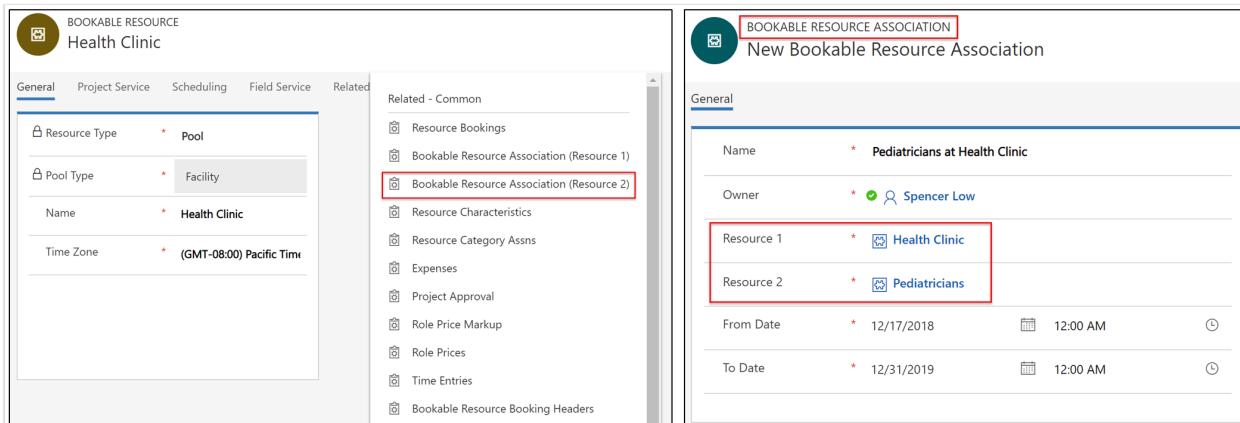
Create resources to represent doctors and add them as resource children to the pediatric doctors pool.

 <p>BOOKABLE RESOURCE Pediatricians</p> <p>General Project Service Scheduling Field Service <b>Resource's Children</b> Related</p> <p>+ Add New Bookable Re... Refresh Run Report Excel Templates Export Bookable Reso...  </p> <p>Bookable Resource Group Associated View  </p>																																				
<table border="1"> <thead> <tr> <th>✓</th> <th>Name</th> <th>Parent Resource</th> <th>Child Resource</th> <th>From Date</th> <th>To Date</th> </tr> </thead> <tbody> <tr> <td></td> <td>Tabatha pediatrician pool</td> <td>Pediatricians</td> <td>Tabatha Guy</td> <td>12/17/2018 12:00 AM</td> <td>12/31/2018 12:00 AM</td> </tr> <tr> <td></td> <td>Tamra pediatrician pool</td> <td>Pediatricians</td> <td>Tamra Riggs</td> <td>12/17/2018 12:00 AM</td> <td>12/31/2018 12:00 AM</td> </tr> <tr> <td></td> <td>Van pediatrician pool</td> <td>Pediatricians</td> <td>Van Amundson</td> <td>12/10/2018 12:00 AM</td> <td>12/31/2018 12:00 AM</td> </tr> <tr> <td></td> <td>Wayne pediatrician pool</td> <td>Pediatricians</td> <td>Wayne Goolsby</td> <td>12/10/2018 12:00 AM</td> <td>12/28/2018 12:00 AM</td> </tr> <tr> <td></td> <td>Wilson pediatrician pool</td> <td>Pediatricians</td> <td>Wilson Chew</td> <td>12/10/2018 12:00 AM</td> <td>12/31/2018 12:00 AM</td> </tr> </tbody> </table>	✓	Name	Parent Resource	Child Resource	From Date	To Date		Tabatha pediatrician pool	Pediatricians	Tabatha Guy	12/17/2018 12:00 AM	12/31/2018 12:00 AM		Tamra pediatrician pool	Pediatricians	Tamra Riggs	12/17/2018 12:00 AM	12/31/2018 12:00 AM		Van pediatrician pool	Pediatricians	Van Amundson	12/10/2018 12:00 AM	12/31/2018 12:00 AM		Wayne pediatrician pool	Pediatricians	Wayne Goolsby	12/10/2018 12:00 AM	12/28/2018 12:00 AM		Wilson pediatrician pool	Pediatricians	Wilson Chew	12/10/2018 12:00 AM	12/31/2018 12:00 AM
✓	Name	Parent Resource	Child Resource	From Date	To Date																															
	Tabatha pediatrician pool	Pediatricians	Tabatha Guy	12/17/2018 12:00 AM	12/31/2018 12:00 AM																															
	Tamra pediatrician pool	Pediatricians	Tamra Riggs	12/17/2018 12:00 AM	12/31/2018 12:00 AM																															
	Van pediatrician pool	Pediatricians	Van Amundson	12/10/2018 12:00 AM	12/31/2018 12:00 AM																															
	Wayne pediatrician pool	Pediatricians	Wayne Goolsby	12/10/2018 12:00 AM	12/28/2018 12:00 AM																															
	Wilson pediatrician pool	Pediatricians	Wilson Chew	12/10/2018 12:00 AM	12/31/2018 12:00 AM																															

## 5. Associate doctor pool to facility pool

Next, navigate to the original facility pool (health clinic), and go to **Related > Bookable Resource Association**.

Set the **Resource 2** field to the pediatric doctor pool resource, as seen in the following screenshot.



The screenshot shows two side-by-side screens. The left screen is the 'Bookable Resource' page for 'Health Clinic', showing fields like General, Project Service, Scheduling, Field Service, and Related. The 'Related' tab is selected, and the 'Bookable Resource Association (Resource 2)' field is highlighted with a red box. The right screen is the 'Bookable Resource Association' page for 'New Bookable Resource Association', showing fields for Name, Owner, Resource 1, Resource 2, From Date, and To Date. The 'Resource 2' field is also highlighted with a red box, showing 'Pediatricians' as the selected resource.

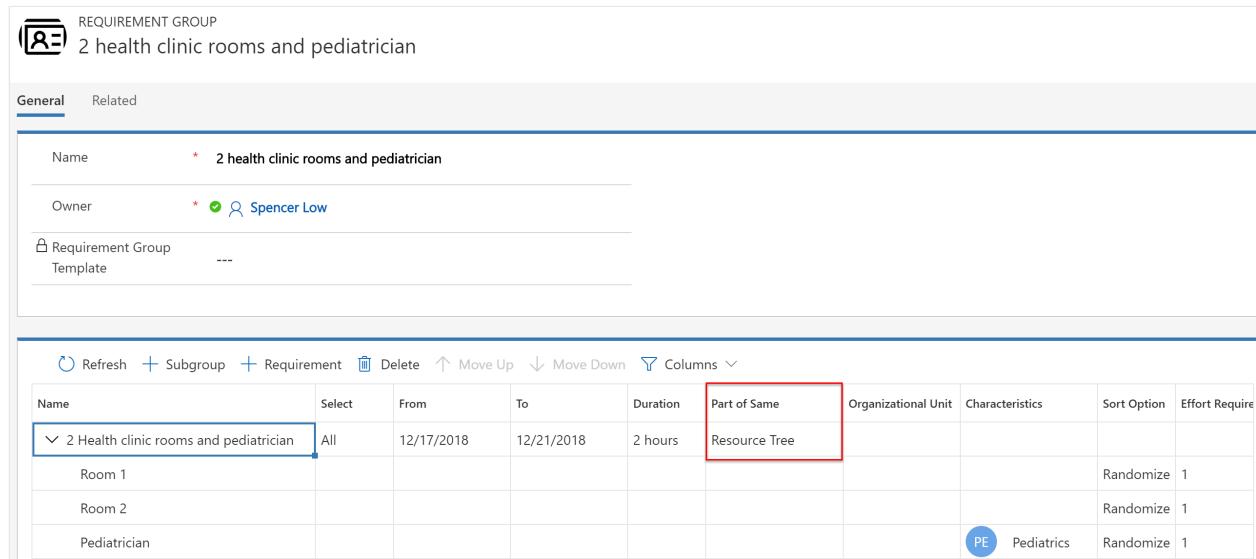
## 6. Create a requirement group for rooms and doctors

Navigate to **Universal Resource Scheduling > Requirement Groups > +New**.

In this example, we created a requirement group that calls for two rooms (facilities) and a pediatric doctor.

Setting **Part of Same to Resource Tree** ensures rooms and pediatricians are related to same facility resource through resource children or resource association.

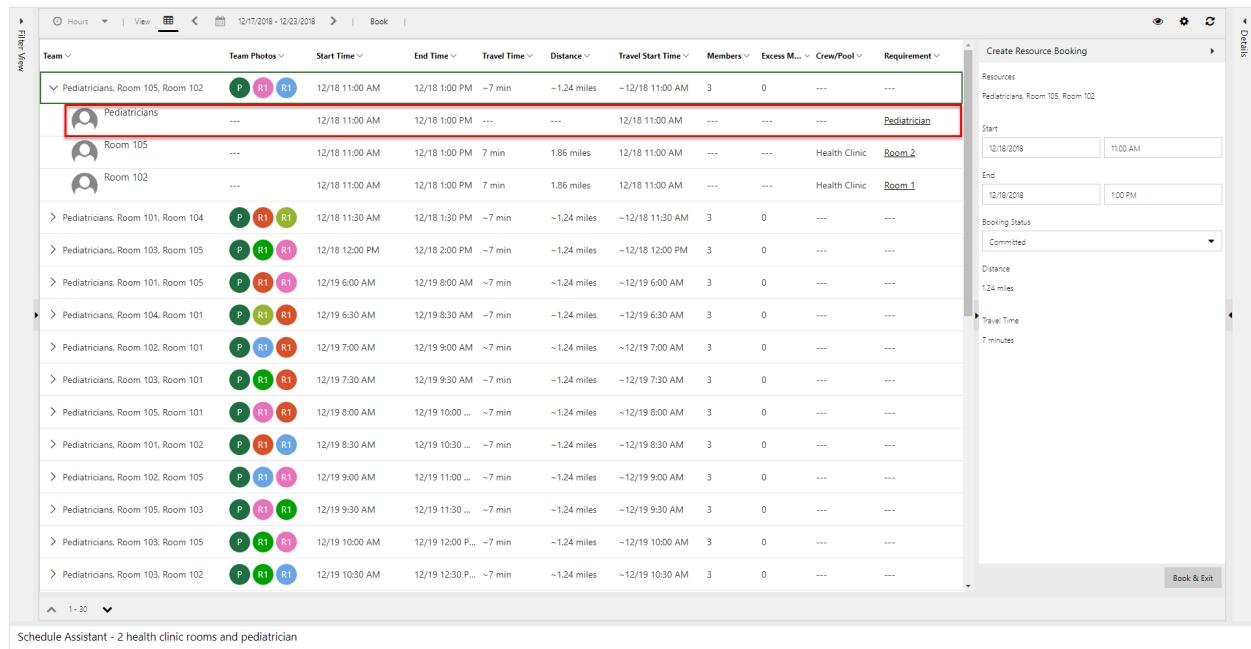
For each room requirement, set the resource type to facility.



The screenshot shows the 'Requirement Group' page for '2 health clinic rooms and pediatrician'. The 'General' tab is selected, showing fields for Name, Owner, and Requirement Group Template. The 'Related' tab is also present. Below the tabs is a table with columns for Name, Select, From, To, Duration, Part of Same, Organizational Unit, Characteristics, Sort Option, and Effort Required. The 'Part of Same' column for the first requirement row is highlighted with a red box, showing 'Resource Tree'. The 'Select' column for the first requirement row is also highlighted with a blue box, showing 'All'. The table also lists 'Room 1', 'Room 2', and 'Pediatrician' as requirements.

The resource types you choose for the pediatrician doctor requirement affects schedule assistant results.

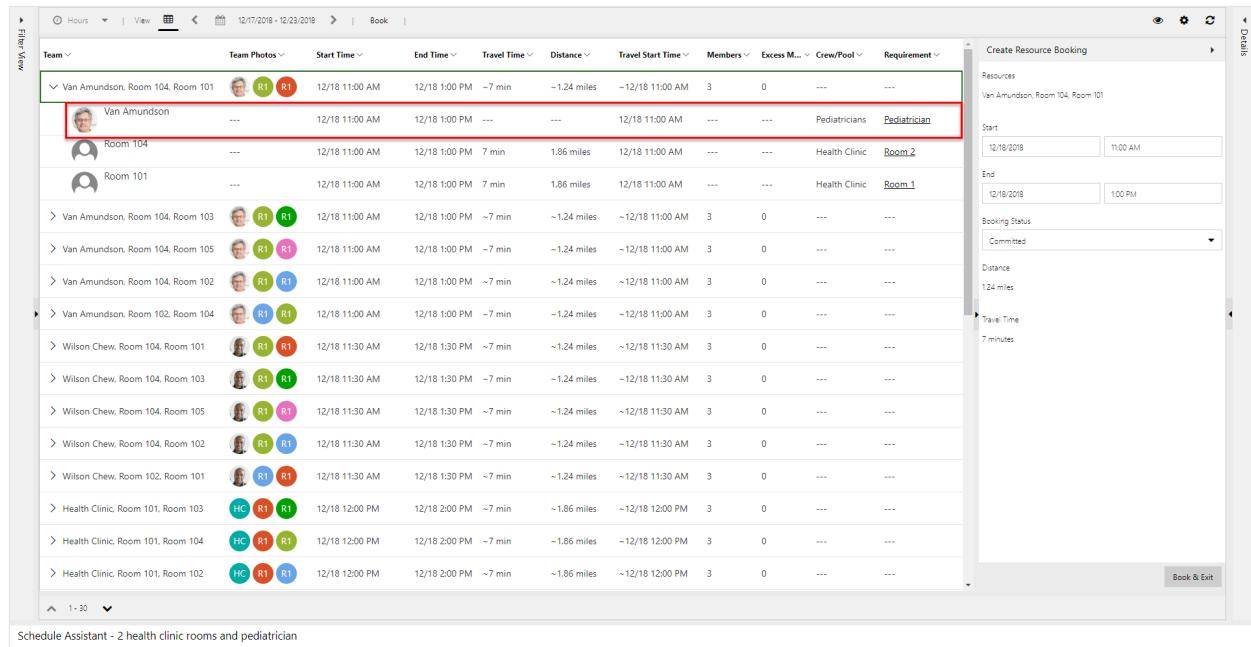
**On the requirement, if Resource Type = Pool and Pool Type = Users, Accounts, Contacts, the pediatrician pool resource shows in the results.**



The screenshot shows the Schedule Assistant interface with a 'Create Resource Booking' dialog open. The dialog is set for 'Pediatricians, Room 105, Room 102' on '12/18/2018' from '11:00 AM' to '1:00 PM', with 'Pediatrician' assigned. The main table lists various pediatrician shifts across different rooms and dates, with a specific row for 'Pediatricians, Room 105, Room 102' highlighted. The table columns include Team, Team Photos, Start Time, End Time, Travel Time, Distance, Travel Start Time, Members, Excess M., Crew/Pool, and Requirement.

This allows you to book the pediatrician pool and assign a specific pediatric doctor at a later time. Whether pediatricians show as results depend on capacity (as derived from the number of doctors in the pool). Using the pool allows schedulers to book appointments based on capacity without having to assign a specific doctor at the time of scheduling.

**On the requirement, if Resource Type = Users, Accounts, Contacts, (Not Pools), specific doctor resources show in results, as seen in the following screenshot.**



The screenshot shows the Schedule Assistant interface with a 'Create Resource Booking' dialog open. The dialog is set for 'Van Amundson, Room 104, Room 101' on '12/18/2018' from '11:00 AM' to '1:00 PM', with 'Pediatrician' assigned. The main table lists various shifts for specific doctors like Van Amundson, Wilson Chew, and Health Clinic staff across different rooms and dates. The table columns include Team, Team Photos, Start Time, End Time, Travel Time, Distance, Travel Start Time, Members, Excess M., Crew/Pool, and Requirement.

**! Note**

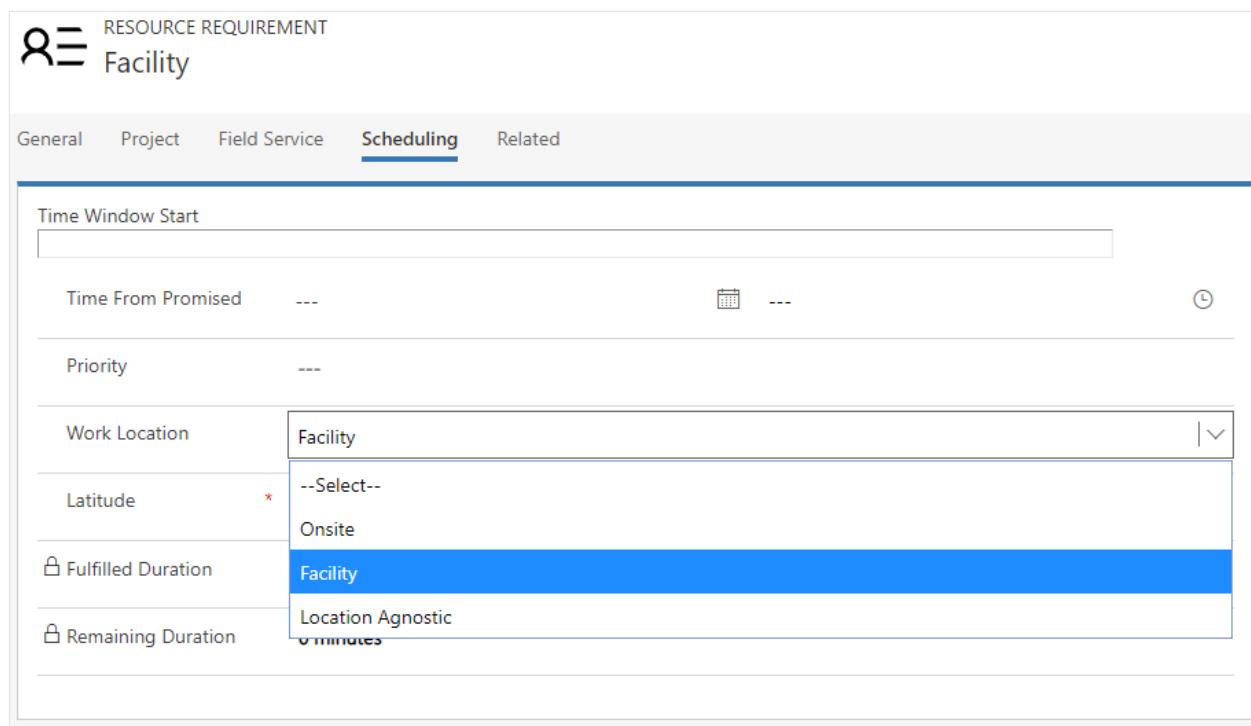
Use fulfillment preferences to display schedule assistant results in neat hourly timeslots.

## Configuration considerations

### Choosing the right work location on requirements

Let's take a look at the following work location types:

- Facility
- Onsite
- Location agnostic



- **Facility** work location implies the interaction takes place at the facility and travel time is calculated as the distance between the customer location and the facility location. The requirement's latitude and longitude fields are used as the customer location. It also means at least one facility or facility pool must return in schedule assistant search results in order for a resource to be returned.
- **On Site** work location implies the interaction takes place at the customer location and travel time is calculated as the distance between the customer location and the resource (typically field technician) location, which is variable based on the resource's schedule that day. The requirement's latitude and longitude fields are used as the customer location. As a result, facility resources and facility pools are excluded from the results.

- **Location Agnostic** work location implies the interaction takes place remotely and the location of the customer nor the resource is considered for scheduling. Travel time isn't applicable and isn't calculated. Facility resources can still be returned as part of the schedule assistant search, but travel time won't be displayed or considered in ranking.

## "Part of Same" options on requirement groups

- **Same Location:** Same location means that only teams of resources working at the same location will be returned. This uses the logic expressed in this document to determine the location, using the Resource Associations (msdyn\_bookableresourceassociations) and the Bookable Resource Group (bookableresourcegroup) entities. Using this option, regardless of which specific facility or facility pool other non-facility resources may be associated to, all that matters is that the resources are at the same physical location (organizational unit).
- **Same Resource Tree:** This option adds an extra layer of stringency to the search. It means that the teams assembled must actually be associated to the same facility or facility pool in order to be returned as a team. For example, let's assume there's one physical location, Location A.

There are 2 facilities at location A: facility 1 and facility 2. If resource 1 is associated to facility 1, and "Same Resource Tree" is selected, the one team that can be assembled is facility 1 + resource 1. Facility 2 and resource 1 can't be returned. This combo could however be returned if "Same Location" is the only option selected.

It works the same with facility pools. Let's assume there's one physical location, Location A. At location A are 2 facilities, facility 1 and facility 2, and a facility pool (facility pool 1). If resource 1 is associated to facility pool 1, and "Same Resource Tree" is selected, the one team that can be assembled is facility pool 1 (or one of its child facilities) + resource 1.

### ⓘ Note

If neither of these two options are selected on the requirement relationship (msdyn\_requirementrelationship), and work location is set to facility, the schedule assistant search will execute as if "Same Resource Tree" was selected.

- **Same Organizational Unit:** An even more stringent option is same organizational unit. This option ensures that the parent organizational unit of the resources are

the same. It doesn't check the bookable resource group or the bookable resource association entity. It only checks the parent organizational unit.

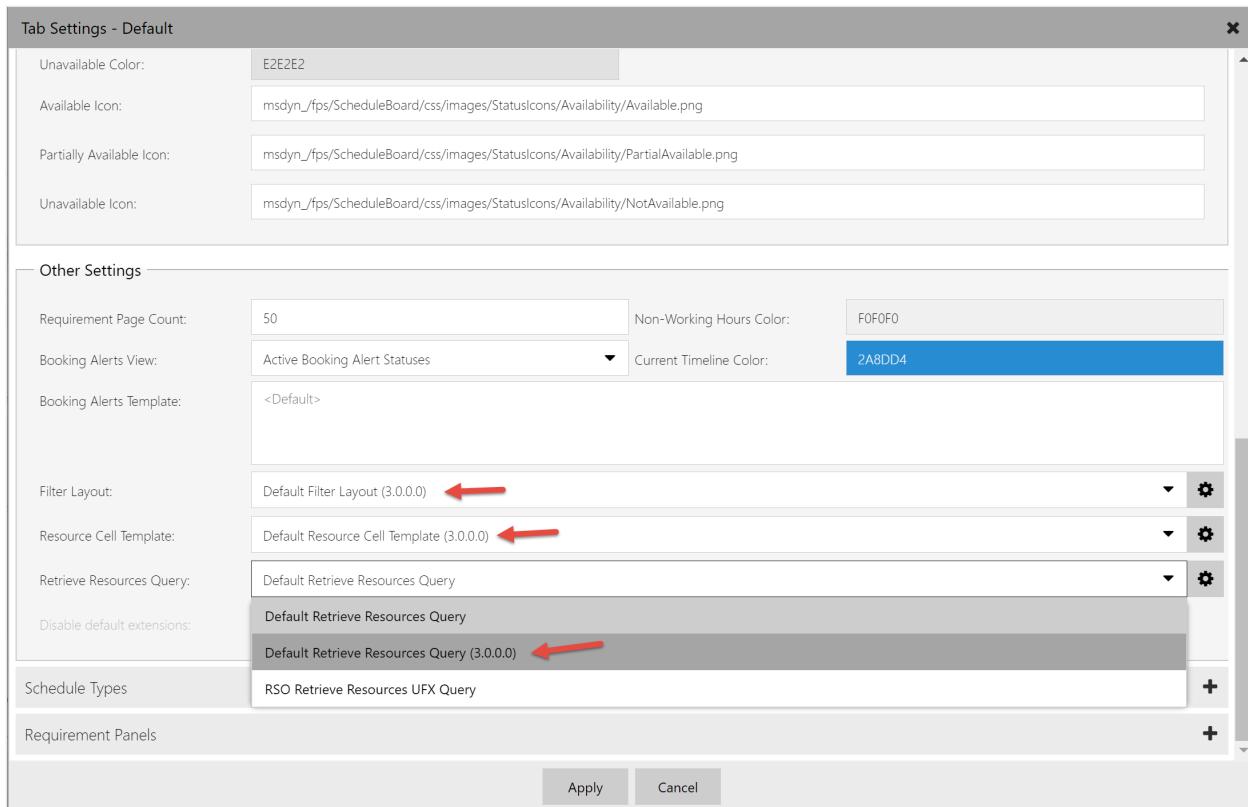
If your implementation uses requirements that are location agnostic, this option may be used without either of the other two options; however, it completely ignores the two aforementioned entities (associations and groups). This could work in a simple implementation where resources are always staffed at the same location, and you don't need the advanced location search functionality of the work location "facility."

 **Note**

The Part of Same field schema name is **msdyn\_requirementrelationship**

## When to use "facility with capacity," "multiple facilities," and "facility pool"

- **Facility with capacity:** this option is configured by adding a capacity to a single facility. It's most useful when schedulers care most about not overbooking, and either don't need to schedule specific facilities or can handle coordination in person when customers arrive at the facility.
- **Multiple facilities:** this option is configured by creating multiple facility resources and relating them to each other through an organizational unit. This option makes the most sense when each facility needs to be scheduled individually.
- **Facility pool:** this option is configured by creating a facility pool and adding facilities as pool members. This option makes the most sense when schedulers want to (1) utilize capacity scheduling by having the facility pool capacity increase and decrease as facilities are added or removed and (2) use local scheduling where bookings are first assigned to the facility pool and then later assigned to pool members. Example: a hotel (facility pool) is first scheduled for a weekend and travelers are assigned specific rooms (facility pool members) at a later date when they arrive.
- When upgrading from Field Service v7.x to v8.x, facility type resources may not appear in schedule assistant results due to default filter settings. To fix this, go to **Field Service > Schedule Board**. Double-click on a schedule board tab, and then **Open Default Settings** in the upper right. Scroll down to **Other Settings**. Set **Retrive Resources Query** to 3.0.0.0 as seen in the following screenshot. **Filter Layout** and **Resource Cell Template** should also be set to 3.0.0.0.



## Additional notes

- For requirements that aren't part of a requirement group, only facility or facility pool resources can return in the schedule assistant if **Work Location** is set to **Facility**.
- A resource can't be related to two facilities (child or association) at the same time.
- There's currently no specific way to visualize every resource related to a facility on the schedule board. The closest way to achieve this is to filter by organizational units.
- Manually scheduling a single requirement to a facility won't create records for all resources related to the facility.

## Facility pool location

The location for a facility pool is taken from the parent organizational unit. If a facility resource is a member of a facility pool, the location of the facility is taken from the pool resource. For example, if you create a facility with a location/organizational unit of **location A**, and you add this facility to a pool, which is located at **location B**, the facility is considered as located at **location B** for as long as it remains in the pool.

## Booking location

When a team is selected and booked, the latitude and longitude of the booked facility/facility pool's location will be stored on the booking record. The work location will also be set based on the work location used when booking in the schedule assistant. In the following conditions, the work location, latitude, and longitude will still be set on the booking:

- If bookings are created without using the schedule assistant
  - If the requirement's work location is set to facility
  - There are latitude and longitude values on the requirement
- 

## Feedback

Was this page helpful?

 Yes

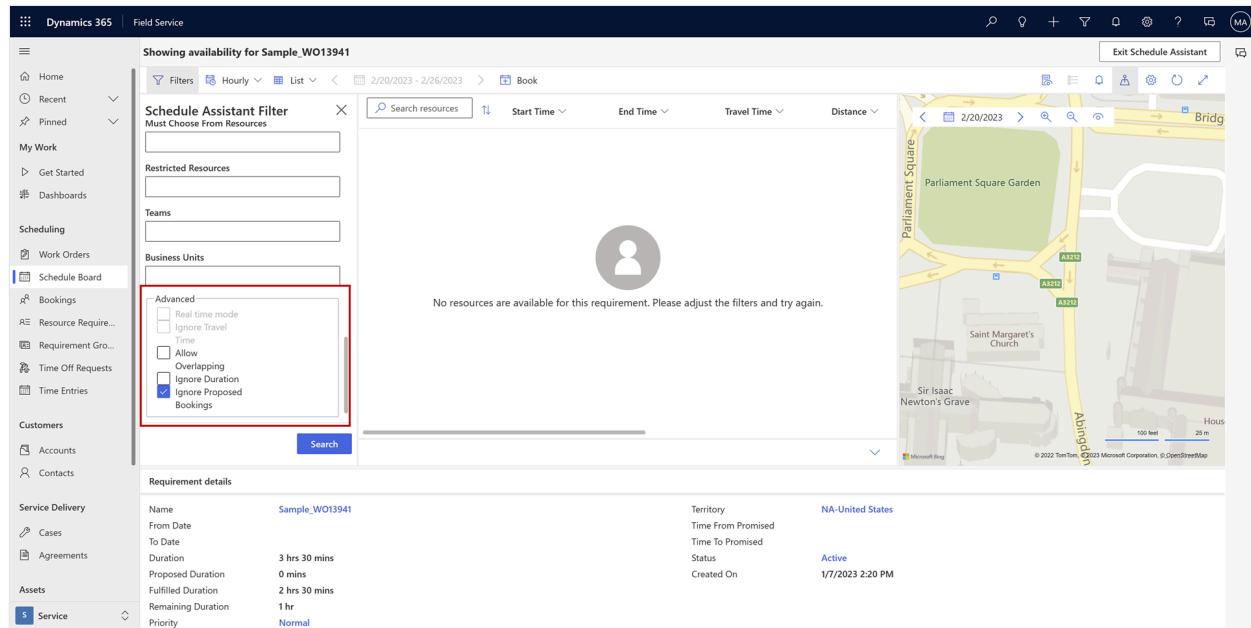
 No

[Provide product feedback !\[\]\(3f1d9e9f6bcc0837db71d34c7a09f75d\_img.jpg\)](#)

# Advanced filters for the schedule assistant

Article • 11/26/2024

The schedule assistant includes advanced filters that provide more granularity for scheduling needs.



## Real time mode

Helps find the closest possible resource to arrive onsite more quickly.

The schedule assistant will use the location of the technician's mobile device when calculating the estimated travel time to arrive at the work order location. A truck icon represents the technician's location on the schedule board map.

The mobile device sends location information from the last sync to a defined threshold in the past. Define the time threshold in the [Geo Location Expires After X Minutes setting for scheduling parameters](#).

See the article on [geofencing](#) to learn how to set up location auditing and sharing.

## Ignore travel time

Ignores the estimated travel time and only checks if a resource has enough time available in their schedule.

## Ignore duration

Ignores the requirement duration when determining if a resource has enough time available in their schedule.

## Ignore proposed bookings

Ignores bookings with a **Proposed** booking status and shows the time slot as available.

Some organizations use the proposed status for bookings until their customer confirms the schedule. Dispatchers could use those time slots to expedite committed work.

## Allow overlapping

Shows bookings with the **Allow Overlap** field set to **Yes** on the bookable resource booking form.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Search resource availability API

Article • 12/12/2024

Field service organizations need to schedule work, often through a service agent directly by the customer. Bookings are typically created based on the resources available to the company and the requirements of the work.

When using at least Dynamics 365 Field Service v8.8.43.51 and Universal Resource Scheduling v3.12.46.21 to schedule work, the `msdyn_SearchResourceAvailability` API can be used to retrieve all the eligible resources for the job, to allow for efficient scheduling of the work. At the time of writing, v3 is the latest version of `msdyn_SearchResourceAvailability` and supports web API calls.

## ① Note

Using the latest version of the API is highly recommended as older versions may use deprecated authentication methods.

## Input parameters

 Expand table

Name	Type	Description	Required	Default
Version	String	<p>The version number of the API identifies the version of the API that should be invoked. It follows the format of major.minor.patch. The request doesn't have to contain the complete version number.</p> <ul style="list-style-type: none"><li>• If only a major version is specified, it invokes the highest minor and patch version available for that major version.</li><li>• If both major and minor versions are specified, it invokes the highest patch version available.</li><li>• If all three parts of the version are mentioned, it invokes the exact version of the API specified.</li></ul>	Yes	-N/A-
IsWebApi	Boolean	Set to <i>True</i> in order to use the schedule assistant via the web API.	Yes	-N/A-
Requirement	Entity	<p>This attribute specifies the resource requirement for which resource availability is being retrieved. It's expected to be a <code>msdyn_resourcerequirement</code> type entity. The requirement can be a preexisting record from the database, or one created on the fly with the necessary constraints. The entity should contain all the specifics that are relevant for your search. The <code>@odata.type</code> for this entity should be <code>Microsoft.Dynamics.CRM.msdyn_requirement</code>. The following some important attributes to populate:</p>	Yes	-N/A-

Name	Type	Description	Required	Default
		1. <b>msdyn_fromdate</b> ( <i>DateTime</i> ): Requirement's from date in ISO format 2. <b>msdyn_todate</b> ( <i>DateTime</i> ): Requirement's to date in ISO format 3. <b>msdyn_remainingduration</b> ( <i>Integer</i> ): The remaining duration of the requirement in minutes 4. <b>msdyn_duration</b> ( <i>Integer</i> ): The total duration of the requirement in minutes		
Settings	Entity	The settings attribute helps to filter the retrieved resources further. Settings are specified as attributes in an entity bag. The type of entity doesn't matter. You can specify any entity logical name.	Yes	-N/A-
ResourceSpecification	Entity	The <code>resourceSpecification</code> attribute is defined as attributes in an entity bag. The <code>@odata.type</code> for this entity should be <code>Microsoft.Dynamics.CRM.expando</code> .	No	None

## Settings entity

The settings entity isn't an entity that exists in the Dataverse; however, it's a collection of all the following attributes that helps the schedule assistant API filter results. Thus, the `@odata.type` for this entity should be `Microsoft.Dynamics.CRM.expando`.

[Expand table](#)

Name	Type	Description	Required	Default
ConsiderSlotsWithLessThanRequiredCapacity	Boolean	Set this to <i>True</i> if a time slot with less than the required capacity (effort) should be considered when computing potential available time slots on the resource's calendar.	No	False
ConsiderSlotsWithLessThanRequiredDuration	Boolean	Set this to <i>True</i> if a time slot with less than the required duration should be considered when computing potential available time slots on the resource's calendar.	No	False
ConsiderSlotsWithOverlappingBooking	Boolean	Set this to <i>True</i> if a time slot with overlapping bookings should be considered when computing potential available time slots on the resource's calendar.	No	False
ConsiderSlotsWithProposedBookings	Boolean	Set this to <i>True</i> if a time slot with proposed bookings should be	No	False

Name	Type	Description	Required	Default
		considered when computing potential available time slots on the resource's calendar.		
ConsiderAppointments	Boolean	<p>Set this to <i>True</i> for search resource availability API to respect existing Dataverse appointments as bookings on the resource, provided the <a href="#">organization and resource level settings have been set</a>.</p> <p>Appointments with statuses <i>Busy</i> or <i>Completed</i> will be considered as unavailable for scheduling operations.</p>	No	False
ConsiderTravelTime	Boolean	Set this to <i>True</i> if travel time should be considered when computing potential time slots on the resource's calendar.	No	True
MovePastStartDateToDate	Boolean	Set this to <i>True</i> to move a start date in the past to the current date.	No	False
UseRealTimeResourceLocation	Boolean	Set this to <i>True</i> if the real-time location of resources should be used when computing potential time slots on the resource's calendar.	No	False
SortOrder	Entity	<p>The sort order can be specified using an entity collection. Each entity in the collection represents one sort criteria. The <code>@odata.type</code> for this entity should be <code>Microsoft.Dynamics.CRM.expando</code>.</p> <p>The following are the attributes you need to populate:</p> <ol style="list-style-type: none"> <li>1. <b>Name</b> (<i>String</i>): The sort criteria</li> <li>2. <b>SortOrder</b> (<i>Integer</i>): The sort direction (0 for ascending and 1 for descending)</li> </ol>	No	None
MaxResourceTravelRadius	Entity	<p>This attribute specifies the maximum that can be defined in an entity. The <code>@odata.type</code> for this entity should be <code>Microsoft.Dynamics.CRM.expando</code>.</p>	No	0 km. If that's the case, no resources are returned for onsite requirements.

Name	Type	Description	Required	Default
		<p>The following are the attributes you need to populate:</p> <ol style="list-style-type: none"> <li>1. <b>Value</b> (<i>Decimal</i>): The radius</li> <li>2. <b>Unit</b> (<i>Integer</i>): The distance unit. See <code>msdyn_distance</code> unit option set for possible values.</li> </ol>		
MaxNumberOfResourcesToEvaluate	Integer	This attribute defines a limit on the number of resources that are considered for the request.	No	If this attribute is not included in the API call, the system uses the Resource Availability Retrieval Limit from schedulable entity definition as defined in <a href="#">Edit settings for enabled entities</a> . If included in the call, it will overwrite the defined Resource Availability Retrieval Limit.
ConsiderOutlookSchedules	Boolean	<p>Set this to <i>True</i> if schedules from Outlook should be considered.</p> <p>Only available in versions 3.1.0 and later</p>	No	False

## Resource specification entity

[Expand table](#)

Name	Type	Description	Required	Default
ResourceTypes	EntityCollection	This attribute specifies the resource type required for the requirement. It can be specified using an entity collection. Each entity in the collection represents one bookable resource type. The <code>@odata.type</code>	No	All resource types except crews

Name	Type	Description	Required	Default
		<p>for this entity should be <code>Microsoft.Dynamics.CRM.msdyn_resourceType</code>. This is the attribute required:</p> <ol style="list-style-type: none"> <li>1. <b>Value (Integer):</b> The option set value that represents the resource type: <ul style="list-style-type: none"> <li>• 1- Generic</li> <li>• 2- Contact</li> <li>• 3- User</li> <li>• 4- Equipment</li> <li>• 5- Account</li> <li>• 6- Crew</li> <li>• 7- Facility</li> <li>• 8- Pools</li> </ul> </li> </ol>		
PreferredResources	EntityCollection	This attribute specifies the resources preferred for the requirement. Adding resources to this entity collection ensures that they are at the top of the list of available resources. Even resources that aren't a part of the entity collection will be on the list, but only after the preferred resources.	No	None
RestrictedResources	EntityCollection	This attribute specifies the resources that shouldn't be considered for the requirement. All time slots of this resource will be filtered out of the list of results from this API.	No	None
MustChooseFromResources	EntityCollection	This attribute specifies the only resources that can be on the list of available resources. It filters out all the other results from the output list.		
Constraints	Entity	This attribute specifies the additional constraints that should be applied to the retrieval of available resources.	No	None
RetrieveResourcesQueryId	Guid	The ID for the Retrieve Resources query.	No	The default Retrieve Resource Query ID.
BookedResourceId	Guid	This attribute specifies the resource currently booked for the requirement.	No	None

**① Note**

The **Preferred/ Restricted / MustChooseFrom** resources attributes can be specified using an entity collection of bookable resource entities. Each entity in the collection represents one **Preferred / Restricted / MustChooseFrom** resource. This is the attribute required for them:

1. **Value (Guid):** The bookable resource ID of the **Preferred / Restricted / MustChooseFrom** resource. The `@odata.type` for this entity should be `Microsoft.Dynamics.CRM.msdyn_bookableresource`.

## Constraints

Additional constraints can be specified through attributes in this entity. The type of entity doesn't matter. You can specify any entity logical name.

Review the **Retrieve Resources Query** on the schedule board settings to identify which constraints might apply. By default, it includes the following:

[ ] [Expand table](#)

Name	Type	Description
Characteristics	EntityCollection	A collection of characteristic IDs that a qualified resource must have.
Roles	EntityCollection	A collection of role IDs that a qualified resource must have.
Territories	EntityCollection	A collection of territory IDs. A qualified resource must be assigned to one of the territories.
UnspecifiedTerritory	Boolean	In combination with the territories constraint, specifies that a qualified must be assigned to one of the territories or no territory at all.
OrganizationalUnits	EntityCollection	A collection of organizational unit IDs. A qualified resource must be a member of one of the specified organizational units.
Teams	EntityCollection	A collection of team IDs. A qualified resource must belong to one of the teams (implies that the resource type is a system user).
BusinessUnits	EntityCollection	A of collection of business unit IDs. A qualified resource must belong to one of the business units (implies that the resource is a system user).

## Output parameters

At the highest level, the output has the following four parameters. The results are represented in entity collections and entities. Responses might not include all the attributes described here as null value or not NA values are omitted from the response. Always check for the presence of an attribute before trying to access it.

[ ] [Expand table](#)

Name	Type	Description
TimeSlots	EntityCollection	A collection of time slot results. For more information, see (time slot entity) [#time-slots-entity] section.
Resources	EntityCollection	A collection of resource results. Resources are represented as a collection of entities with the following attributes: <ol style="list-style-type: none"> <li><b>BookableResource</b> (<i>Entity</i>): The bookable resource entity that is available for the requirement.</li> <li><b>TotalAvailableTime</b> (<i>Double</i>): The total available time for the resource to perform the requirement.</li> </ol>
Related	Entity	Related resources represent resources and time slots of resources that aren't directly qualified for the requested requirement but are related. For example, if a crew member qualifies for a requirement, then the other members of that crew would be related results. <ol style="list-style-type: none"> <li><b>Timeslots</b> (<i>EntityCollection</i>): Time slots of related resources. The definition of time slots is the same as described in the <a href="#">time slots section</a>.</li> <li><b>Resources</b> (<i>EntityCollection</i>): The related resources. The definition of resources is the same as described in the resources attribute definition.</li> </ol>
Exceptions	Entity	This attribute contains information about any exception that occurred and information about if and where the resource search was truncated. <ol style="list-style-type: none"> <li><b>Message</b> (<i>String</i>): Exception message</li> <li><b>ResourcesTruncatedAt</b> (<i>Integer</i>): If the number of resources exceeded the retrieval limit; the number where the resources where truncated.</li> </ol>

## Time slots entity

[Expand table](#)

Name	Type	Description
ID	Guid	Unique identifier for the time slot
Type	Integer	The type of time slot can be one of the following: <ul style="list-style-type: none"> <li>0: Available</li> <li>1: Scheduled</li> <li>2: Off</li> <li>3: Break</li> </ul>
StartTime	DateTime	The start time of the time slot. If there's travel for the requirement, then this is the start time of travel. If not, this is the start time of the requirement.
ArrivalTime	DateTime	The arrival time of the time slot. If there's travel for the requirement, then this is the start time of requirement, after travel has been completed. If not, it's the same as the start time of the time slot.

Name	Type	Description
EndTime	DateTime	The end time of the time slot.
Effort	Integer	The effort or capacity of the resource to carry out the requirements.
ResourceRequirement	EntityReference	The resource requirement for which time slots are being retrieved.
Potential	Boolean	A boolean value indicating if the time slot has potential to fulfill the requested requirement.
IsDuplicate	Boolean	A boolean value indicating if the time slot is a duplicate.
AllowOverlapping	Boolean	A boolean value indicating if overlapping is allowed.
Resource	Entity	The resource to which the time slot belongs. For more information, see <a href="#">time slot resource</a> .
Location	Entity	<p>The location has three attributes:</p> <ol style="list-style-type: none"> <li><b>Location (Entity):</b> It has two attributes - <ul style="list-style-type: none"> <li>Latitude</li> <li>Longitude</li> </ul> </li> <li><b>WorkLocation (Integer):</b> It has three attributes - <ul style="list-style-type: none"> <li>Onsite. Onsite requirements exclude pool and facility resource types from the results.</li> <li>Facility</li> <li>Location Agnostic</li> </ul> </li> <li><b>LocationSourceSlot (Integer):</b> The source of location information has three attributes - <ul style="list-style-type: none"> <li>Common</li> <li>Custom GPS entity</li> <li>Mobile audit</li> </ul> </li> </ol>
Travel	Entity	<p>This entity contains details of travel time and distance information for a time slot. The following are the attributes:</p> <ol style="list-style-type: none"> <li><b>Distance (Double):</b> The travel distance</li> <li><b>TravelTime (Double):</b> The travel time in minutes.</li> <li><b>DistanceFromStartLocation (Double):</b> The distance from the resource's start location.</li> <li><b>DistanceFromEndLocation (Double):</b> The distance from the resource's end location.</li> <li><b>DistanceMethodSourceSlot (Integer):</b> The source / calculation type of the distance values <ul style="list-style-type: none"> <li>Map Service</li> <li>As the crow flies</li> </ul> </li> </ol>
Next	Entity	This entity contains details about the travel time and distance to the next time slot booking.

Name	Type	Description
		<ol style="list-style-type: none"> <li>1. <b>NextScheduleLocation</b> (<i>Entity</i>): The location of the next booking. The entity has two attributes:           <ul style="list-style-type: none"> <li>• Latitude</li> <li>• Longitude</li> </ul> </li> <li>2. <b>NextScheduleTravelTime</b> (<i>Integer</i>): The travel time to the next booking in minutes.</li> </ol>
Availability	Entity	<p>The detailed availability information for a time slot. This is used with time groups.</p> <ol style="list-style-type: none"> <li>1. <b>AvailableIntervals</b> (<i>EntityCollection</i>): A collection of available intervals. Each entity in this collection contains details about a time group interval.           <ul style="list-style-type: none"> <li>• <b>StartTime</b> (<i>DateTime</i>): The start time.</li> <li>• <b>ArrivalTime</b> (<i>DateTime</i>): The arrival time.</li> <li>• <b>EndTime</b> (<i>DateTime</i>): The end time.</li> <li>• <b>TimeGroupId</b> (<i>DateTime</i>): The time group ID.</li> <li>• <b>TimeGroupDetailStartTime</b> (<i>DateTime</i>): The time group start time.</li> <li>• <b>TimeGroupDetailEndTime</b> (<i>DateTime</i>): The time group end time.</li> </ul> </li> <li>2. <b>TotalAvailableDuration</b> (<i>Double</i>): The total available duration in minutes.</li> <li>3. <b>TotalAvailableTime</b> (<i>Double</i>): The total available time a resource has in a day (in minutes).</li> </ol>
TimeGroup	Entity	<p>The details about a time group.</p> <ol style="list-style-type: none"> <li>1. <b>TimeGroupId</b> (<i>Guid</i>): The time group ID.</li> <li>2. <b>TimeGroupDetail</b> (<i>EntityReference</i>): An entity reference to the time group detail.</li> <li>3. <b>TimeGroupDetailStartTime</b> (<i>DateTime</i>): The time group detail start time.</li> <li>4. <b>TimeGroupDetailEndTime</b> (<i>DateTime</i>): The time group detail end time.</li> </ol>

### 💡 Tip

When you create bookings with the API, use the *Potential* field described in the table. Not using that field might lead to overlapping or unsuitable bookings.

## Time slot resource

 [Expand table](#)

Name	Type	Description
Resource	EntityReference	An entity reference to the bookable resource.
ResourceGroup	EntityReference	An entity reference to the bookable resource group.
BusinessUnit	EntityReference	An entity reference to the business unit.
OrganizationalUnit	EntityReference	An entity reference to the organizational unit.
ResourceType	Integer	The resource type. See the <b>ResourceType</b> attribute on the <b>BookableResource</b> entity for possible values.
PoolId	Guid	The ID of the pool that the resource is a member of during the time slot.
CrewId	Guid	The ID of the crew that the resource is a member of during the time slot.
Characteristics	EntityCollection	<p>The bookable resource characteristics. Each entity in the collection contains entities with characteristics and rating information.</p> <ol style="list-style-type: none"> <li><b>Characteristic</b> (<i>EntityReference</i>): An entity reference to the characteristic.</li> <li><b>RatingId</b> (<i>Guid</i>) The rating ID for the characteristic.</li> <li><b>RatingName</b> (<i>String</i>): The rating name.</li> <li><b>RatingValue</b> (<i>Integer</i>): The rating value.</li> </ol>
HasStartLocation	Boolean	A boolean value indicating if the resource has a start location.
HasEndLocation	Boolean	A boolean value indicating if the resource has an end location.
Email	String	The resource's email address.
Phone	String	The resource's phone number.
ImagePath	String	The path to the resource's image.
CalendarId	Guid	The resource's calendar ID.

## Examples

In this example, v3 of schedule assistant API, which allows for web API calls, is being used for a requirement of duration 60 minutes. Using the settings attribute, the results are being filtered down. Two resource types are being considered for the final results: 1 and 2 (in other words, generic and contact).

JSON

```
{
  "Version": "3",
  "IsWebApi": true,
  "Requirement": {
    "msdyn_fromdate": "2021-07-14T00:00:00Z",
    "msdyn_todate": "2021-07-15T23:59:00Z",
    "msdyn_remainingduration": 60,
    "msdyn_duration": 60,
    "msdyn_TimeGroup@odata.bind": "/msdyn_timegroups(c3dc79ea-d12f-ee11-9cc9-
```

```

000d3a745a58)",
    "@odata.type": "Microsoft.Dynamics.CRM.msdyn_resourcerequirement"
},
"Settings": {
    "ConsiderSlotsWithProposedBookings": false,
    "MovePastStartDateToCurrentDate": true,
    "@odata.type": "Microsoft.Dynamics.CRM.expando"
},
"ResourceSpecification": {
    "@odata.type": "Microsoft.Dynamics.CRM.expando",
    "ResourceTypes@odata.type": "Collection(Microsoft.Dynamics.CRM.expando)",
    "ResourceTypes": [
        {
            "@odata.type": "Microsoft.Dynamics.CRM.expando",
            "value": "1"
        },
        {
            "@odata.type": "Microsoft.Dynamics.CRM.expando",
            "value": "2"
        }
    ],
    "Constraints": {
        "@odata.type": "Microsoft.Dynamics.CRM.expando",
        "Characteristics@odata.type": "Collection(Microsoft.Dynamics.CRM.expando)",
        "Characteristics": [
            {
                "@odata.type": "Microsoft.Dynamics.CRM.expando",
                "characteristic": {
                    "@odata.type": "Microsoft.Dynamics.CRM.expando",
                    "value": "67387f9f-12e2-ec11-bb43-000d3aed25f7"
                }
            }
        ],
        "Territories@odata.type": "Collection(Microsoft.Dynamics.CRM.expando)",
        "Territories": [
            {
                "@odata.type": "Microsoft.Dynamics.CRM.expando",
                "value": "cc19f004-4483-ee11-8178-000d3a5c32c3"
            }
        ]
    }
}
}

```

The following example demonstrates proper usage of entity collections. In this case, it specifies MustChooseFromResources.

JSON

```
{
    "Version": "3",
    "IsWebApi": true,
    "Requirement": {
        "msdyn_fromdate": "2021-07-14T00:00:00Z",
        "msdyn_todate": "2021-07-15T23:59:00Z",
        "msdyn_remainingduration": 60,
        "msdyn_duration": 60,
        "msdyn_TimeGroup@odata.bind": "/msdyn_timegroups(c3dc79ea-d12f-ee11-9cc9-000d3a745a58)",
        "msdyn_StartTimeGroup@odata.bind": "/msdyn_timegroups(1a233a2a-1a23-4a23-8a23-1a233a2a1a23)"
    }
}
```

```

    "@odata.type": "Microsoft.Dynamics.CRM.msdyn_resourcerequirement"
  },
  "Settings": {
    "ConsiderSlotsWithProposedBookings": false,
    "MovePastStartDateToCurrentDate": true,
    "MaxNumberOfResourcesToEvaluate":500,
    "@odata.type": "Microsoft.Dynamics.CRM.expando"
  },
  "ResourceSpecification": {
    "@odata.type": "Microsoft.Dynamics.CRM.expando",
    "ResourceTypes@odata.type": "Collection(Microsoft.Dynamics.CRM.expando)",
    "ResourceTypes": [
      {
        "@odata.type": "Microsoft.Dynamics.CRM.expando",
        "value": "1"
      },
      {
        "@odata.type": "Microsoft.Dynamics.CRM.expando",
        "value": "2"
      }
    ],
    "MustChooseFromResources@odata.type": "Collection(Microsoft.Dynamics.CRM.expando)",
    "MustChooseFromResources": [
      {
        "@odata.type": "Microsoft.Dynamics.CRM.expando",
        "value": "2145a982-f718-ed11-b83e-0022482d79c8",
      }
    ],
    "Constraints": {
      "@odata.type": "Microsoft.Dynamics.CRM.expando",
      "Characteristics@odata.type": "Collection(Microsoft.Dynamics.CRM.expando)",
      "Characteristics": [
        {
          "@odata.type": "Microsoft.Dynamics.CRM.expando",
          "characteristic": {
            "@odata.type": "Microsoft.Dynamics.CRM.expando",
            "value": "67387f9f-12e2-ec11-bb43-000d3aed25f7"
          }
        }
      ],
      "Territories@odata.type": "Collection(Microsoft.Dynamics.CRM.expando)",
      "Territories": [
        {
          "@odata.type": "Microsoft.Dynamics.CRM.expando",
          "value": "cc19f004-4483-ee11-8178-000d3a5c32c3"
        }
      ]
    }
  }
}

```

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

# Search for resource availability and create bookings for requirement groups

Article • 09/19/2024

Use the *Search Resource Availability for Requirement Group API* and the *Create Requirement Group Bookings API* to book the resources that meet the needs you identify in your requirement groups.

- The [Search Resource Availability for Requirement Group API](#) returns available time slots for resources when you search by using requirement groups.
- The [Create Requirement Group Bookings API](#) uses the available time slots for resources to generate booking records for your requirement groups.

You pass the details of a requirement group in your API calls and retrieve a list of available resources and their open time slots. This list is helpful for self-scheduling scenarios where a user wants to view the availability of resources, or for portal scheduling scenarios where a customer wants to view resource availability from a website or app.

At the time of writing, v1 is the latest version of `msdyn_SearchResourceAvailabilityForRequirementGroup` and supports web API calls.

## Search Resource Availability for Requirement Group API

Use the following input and output parameters for the *Search Resource Availability for Requirement Group* (`msdyn_SearchResourceAvailabilityForRequirementGroup`) API.

### Parameters

 [Expand table](#)

Name	Type	Required	Description
Version	String	Yes	The version number of the API. The version number identifies the version of the API that should be invoked. The version number is a semantic version number of the format major.minor.patch. The request doesn't have to contain the complete version number.
RequirementGroup		Yes	An entity reference to the requirement group entity.
RequirementSpecification	Integer	No	If left null, respects the targeting requirement group duration by default.
Settings	<code>Entity&lt;InputSettings&gt;</code>	No	Sets the settings for the request.

### Input

 [Expand table](#)

Name	Type	Required	Description
ConsiderSlotsWithLessThanRequiredDuration	Boolean	No	Specifies if a time slot with less than the required remaining duration should be considered when computing potential time slots. It's false by default.
ConsiderSlotsWithOverlappingBooking	Boolean	No	Specifies if time slots with overlapping bookings should be considered when computing potential time slots. It's false by default.
ConsiderSlotsWithProposedBooking	Boolean	No	Specifies if time slots with proposed bookings should be considered when computing potential time slots. It's false by default.
MaxResourceTravelRadius	Distance	No	Specifies

Name	Type	Required	Description
			the maximum travel radius for resources when computing available time slots.
SortOrder	Integer	No	Specifies the requirement group order for the response.
PageSize	Integer	No	Numbers of item returned in a page. It's 20 by default.
PagingCookie	String	No	Paging cookie retrieved from previous searching result.
OrganizationUnits	List<Guid>	No	A collection of organization unit IDs. A qualified resource must be a member of one of the specified organization units.
MustChooseFromResources	List<Guid>	No	Evaluate and select results from resources in this list.
RequiredResources	List<Guid>	No	Evaluate all resources, filter the results based on this list. In general, use MustChooseFromResources instead for improved performance.
IgnoreTimeSlots	Boolean	No	Specifies if the returned time slots should be ignored. When true, list of time slots returned is empty. It's false by default.
ConsiderAppointments	Boolean	No	Set to True for search resource availability API to respect existing Dataverse appointments as bookings on the resource, provided the organization and resource level settings are set. Appointments with busy or completed statuses are considered as unavailable for scheduling operations.

## Output

[Expand table](#)

Returns	Name(Type)	Description
TimeSlots (List<OutputTimeSlot>)	StartTime (DateTime)	The start time.
	EndTime (DateTime)	The end time.
	ArrivalTime (DateTime)	The arrival time.
	Travel(OutputTimeSlotTravel)	The time slot travel information.
	OutputTimeSlotTravel	Is only present if the resource requirement contains values for latitude and longitude.
	<ul style="list-style-type: none"> <li>Distance (Double)</li> <li>TravelTime (Double)</li> <li>DistanceFromStartLocation (Double)</li> <li>TravelTimeToEndLocation (Double)</li> </ul>	
	Effort (Double)	The effort/capacity.
	IsDuplicate (Boolean)	A Boolean value indicating if the time slot is a duplicate.
	Resource(OutputResource)	The Resource entity as explained in this article.
	OutputResource	<ul style="list-style-type: none"> <li>Resource (BookableResource)</li> <li>TotalAvailableTime (Double)</li> </ul>
	Location(OutputTimeSlotLocation)	The entity contains details about the location of a time slot. For more information, see TimeSlotLocation in this article.
	OutputTimeSlotLocation:	
	<ul style="list-style-type: none"> <li>WorkLocation (Enum): <ul style="list-style-type: none"> <li>Onsite (0)</li> <li>Facility (1)</li> <li>Location agnostic (2)</li> </ul> </li> </ul>	

>Returns	Name(Type)	Description
	<ul style="list-style-type: none"> <li>• LocationSourceSlot (Enum): <ul style="list-style-type: none"> <li>◦ Common (1)</li> <li>◦ Custom GPS entity (2)</li> <li>◦ Mobile audit (3)</li> </ul> </li> </ul>	
	TimeGroup(TimeSlotTimeGroup)  OutputTimeSlotTimeGroup: <ul style="list-style-type: none"> <li>• TimeGroupId (Guid)</li> <li>• TimeGroupDetail (EntityReference)</li> <li>• TimeGroupDetailStartTime (DateTime)</li> <li>• TimeGroupDetailEndTime (DateTime)</li> </ul>	The entity contains details about a time group. For more information, Refer to TimeSlotTimeGroup in this article.
	AvailableIntervals (List<<Guide>OutputTimeSlot>)	A collection of available intervals.
Resources (List<OutputResource>)	Resource (EntityReference)	An entity reference to the bookable resource.
	BusinessUnit (EntityReference)	An entity reference to the bookable resource group.
	OrganizationalUnit (EntityReference)	An entity reference to the organizational unit.
	ResourceType (Int)	The resource type. Refer to the resourcetype attribute on the BookableResource entity for possible values.
	PoolId (Guid)	The ID of the pool that the resource is a member of during the time slot.
	CrewId (Guid)	The ID of the crew that the resource is a member of during the time slot.
	Email (String)	The resource's email address.
	Phone (String)	The resource's phone number.
	ImagePath (String)	The path to the resource's image.
Requirements (List<OutputRequirements>)	Requirement (EntityReference)	An entity reference to the Resource Requirement record.
	ConstraintBag (String)	Requirement constraint in ufx bag(internal)
	Resources (List<<EntityReference>EntityReference>)	Entity reference list of resource that is available to the requirements.
ProposalResourceAssignmentSets (List<OutputProposalResourceAssignmentSet>)	IntervalStart (DateTime)	Start time for each proposal resource assignment set.
	ProposalResourceAssignments (List<OutputProposalResourceAssignments>)  OutputProposalResourceAssignments: <ul style="list-style-type: none"> <li>• RequirementId (Guid)</li> <li>• ResourceId (Guid)</li> </ul>	List of Resources assigned to Requirement.
PagingInfos (OutputPagingInfo)	MoreResults (Boolean)	If there are more results or not.

>Returns	Name(Type)	Description
	PagingCookie (String)	Paging cookie that can be used in the future search.

## Example payload

JSON

```
{
  "RequestName": "msdyn_SearchResourceAvailabilityForRequirementGroup",
  "Parameters": [
    {
      "Key": "Version",
      "Value": "1"
    },
    {
      "Key": "RequirementGroup",
      "Value": {
        "Id": "6927721a-0137-42be-8092-26995625a9d9",
        "LogicalName": "msdyn_requirementgroup",
        "Name": null,
        "KeyAttributes": [],
        "RowVersion": null
      }
    }
  ],
  "RequestId": null
}
```

## Example response

JSON

```
{
  "ResponseName": "msdyn_SearchResourceAvailabilityForRequirementGroup",
  "Results": [
    {
      "Key": "TimeSlots",
      "Value": {....}
    },
    {
      "Key": "Requirements",
      "Value": {....}
    },
    {
      "Key": "ProposalResourceAssignmentSets",
      "Value": {....}
    },
    {
      "Key": "PagingInfos",
      "Value": {....}
    }
  ]
}
```

## Create Requirement Group Bookings API

Use the following input and output parameters for the Create Requirement Group Bookings (msdyn\_CreateRequirementGroupBookings) API.

### Parameters

 [Expand table](#)

Name	Type	Required	Description
Version	String	Yes	The version number of the API. The version number identifies the version of the API that should be invoked. The version number is a semantic version number of the format major.minor.patch. The request doesn't have to contain the complete version number.
RequirementGroup	EntityReference	Yes	An entity reference to the requirement group entity, usually is a GUID, as shown in the following sample.
Start	DateTime	Yes	Start time of the Timeslot.
Duration	Integer	Yes	The Duration of the Booking to be created.
ResourceAssignments	EntityCollection	Yes	It's an entity collection of the Resource Assignments that are to be made for the Bookings to be created. Look at the Resource Assignment entity table for more details

## Resource Assignments

[Expand table](#)

Name	Type	Required	Description
RequirementId	Guid	Yes	The resource requirement ID of the Requirement for which the booking record is to be created.
ResourceId	Guid	Yes	The bookable resource ID of the Resource for which you want to create the booking.
BookingStatusId	Guid	Yes	The booking status ID of the booking to be created.
Effort	Integer	No	The capacity of the Bookable Resource that this booking consumes.
TravelTime	Integer	No	The travel time in minutes.

## Output

Returns `HandlerExecuted` that is of Boolean type.

---

## Feedback

Was this page helpful?

[Yes](#)

[No](#)

[Provide product feedback](#)

# Enable an entity for scheduling

Article • 11/13/2024

Enabling an entity for scheduling allows dispatchers to schedule the requirements related to that entity to the matching resources.

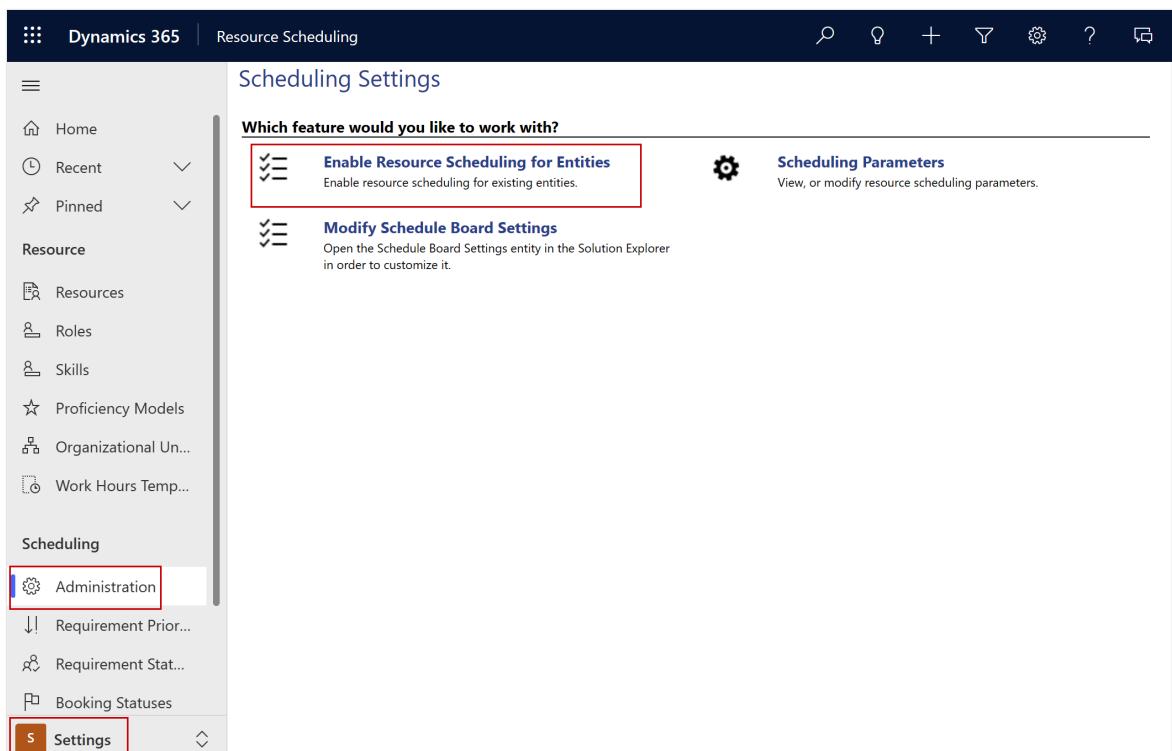
## Prerequisites

- Field Service - Administrator or System Administrator security role.

## Enable an entity for scheduling

Enable an entity for scheduling.

1. In Resource Scheduling, change to the **Settings** area. Under **Scheduling**, select **Administration**. Select **Enable Resource Scheduling for Entities**.



2. In the **Enable New Entity** section, under **Add Entity**, choose an entity from the list.

### Tip

If the entity you want to enable doesn't show in list, update the **Can be customized** setting to **True** for [the managed property](#).

3. For **Booking Relationship** and **Requirement Relationship**, choose **Create New Relationship**. Or, if you have an existing relationship to the *bookable resource booking* or *resource requirement* entities, select it.
4. Select **Publish Customization**.

## Disable scheduling for an enabled entity

Turn off scheduling for a previously enabled entity.

1. In Resource Scheduling, change to the **Settings** area. Under **Scheduling**, select **Administration**. Select **Enable Resource Scheduling for Entities**.
2. In the **Enabled Entities** section, double-click the entity you want to disable.

Setup Wizard – Enable Scheduling

Enable New Entity

Add Entity:

Booking Relationship:

Requirement Relationship:

Publish Customization

Enabled Entities

Default Metadata Settings (none)

Appointment (appointment)

Work Order (msdyn\_workorder)

3. A new tab with the booking setup metadata for the selected entity opens. Select **Deactivate** in the command bar and confirm the deactivation.

## Edit settings for enabled entities

Edit booking settings for entities you enabled for scheduling.

1. In Resource Scheduling, change to the **Settings** area. Under **Scheduling**, select **Administration**. Select **Enable Resource Scheduling for Entities**.
2. In the **Enabled Entities** section, double-click the entity you want to edit. A new tab with the booking setup metadata for the selected entity opens.

Expand table

Setting	Effect	Value
Default booking duration	Defines the default duration of a new booking.	Predefined set of duration options.
Default booking committed status	Sets the default value to use when a booking moves into <i>Committed</i> state.	Available <a href="#">booking statuses</a> .
Default requirement active status	Sets the default value to use when a requirement moves into <i>Active</i> state.	Available requirement statuses.
Default requirement canceled status	Sets the default value to use when a requirement moves into <i>Canceled</i> state.	Available requirement statuses.
Resource availability retrieval limit	Defines the maximum number of resources returned as a result in the scheduling assistant.	Number.
Enable quick book	Use the <a href="#">quick book experience</a> on an entity record.	Yes: Use quick book experience. No: Use schedule assistant experience.
Cancel bookings when moving	Cancel bookings when moving the booking to a different time slot by using the <i>Move bookings</i> feature on the schedule board.	Yes: Cancel booking and create a new one in the updated time slot. No: Keep booking and update start/end time.
Default booking canceled status	Sets the default value to use when a booking moves into <i>Canceled</i> state.	Available <a href="#">booking statuses</a> .
Disable requirement auto creation for bookings	Defines if the system creates a requirement for a booking.	Yes: Disable automated requirement creation. No: Enable

Setting	Effect	Value
		automated booking creation.
Default requirement completed status	Sets the default value to use when a requirement moves into <i>Completed</i> state.	Available requirement statuses.
Available duration minimum (%)	Defines the availability of a resource in relation to the booking duration. For example, if the value is 50 and the booking takes four hours. The system only considers resources for the job that have at least two hours available.	0 to 100.

3. In the **Attribute Mapping** section, choose values on the target entity that match with the requirement fields.

4. **Save** your changes.

An entity is successfully enabled for scheduling when:

- *Resource Requirements* shows as a related entity. Create them manually or with a custom workflow.
- The **Book** option appears on the entity record.

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) 

# Enable automatic address geocoding

Article • 11/07/2024

Geocoding is the process of transforming text-based address descriptions into geographical coordinates. Dynamics 365 Field Service coordinate to estimate travel times when scheduling a work order to a resource.

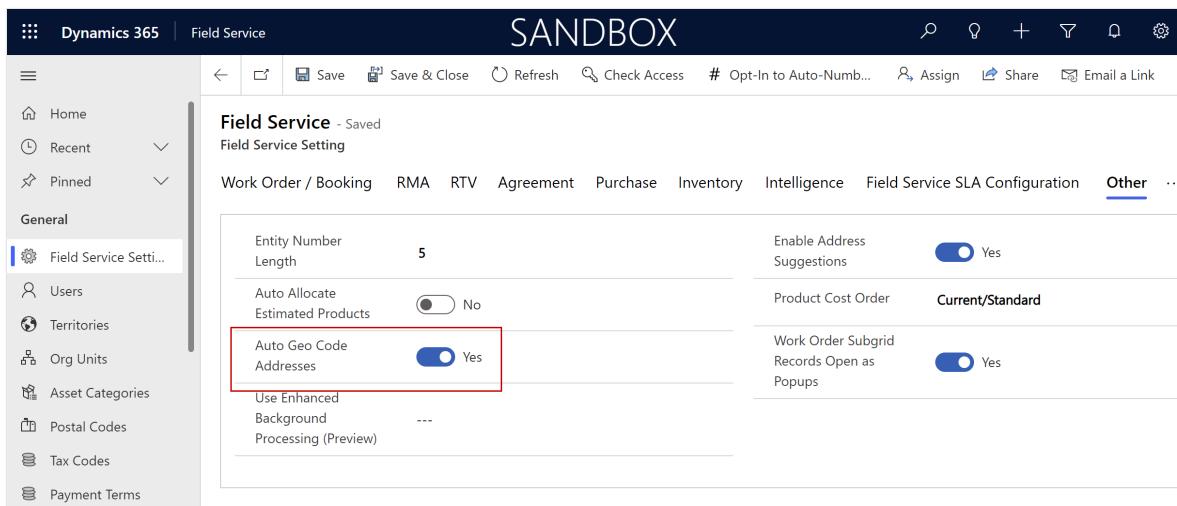
The **Auto Geo Code Addresses** setting enables the system to automatically get coordinates (latitude and longitude) based on the address of a service account or a resource.

## Prerequisites

To use geocoding or location services, you need to enable on maps. For more information, see [Connect to maps](#).

## Enable automatic geocoding

1. Open the **Field Service** app.
2. Change to the **Settings** area and then choose **Field Service Settings**.
3. In **Other** tab, set **Auto Geo Code Addresses** to **Yes**.



4. Select **Save**.

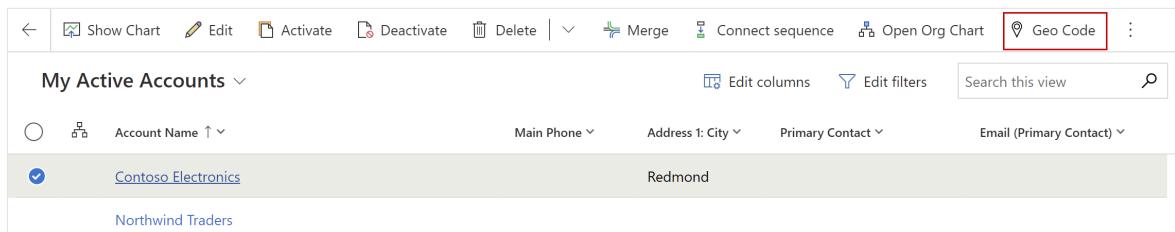
The geocoding process in Field Service uses only one street field (**Street1**). Additional information like apartment number in **Street2** and **Street3** is ignored. Don't add the extended information to **Street1**. The string length could be too long to process.

## 💡 Tip

When the system finds no geocode match, it returns the value (0,0). Any custom logic that uses geocoding should ignore these results to avoid incorrect locations and travel routes.

## Get coordinates for an address on a record

1. Open a work order or a service account.
2. Ensure the address is accurate.
3. On the command bar, select **Geo Code**.



4. On the map dialog box, make sure you have the correct address, and then select **Change**.

## Feedback

Was this page helpful?

 Yes

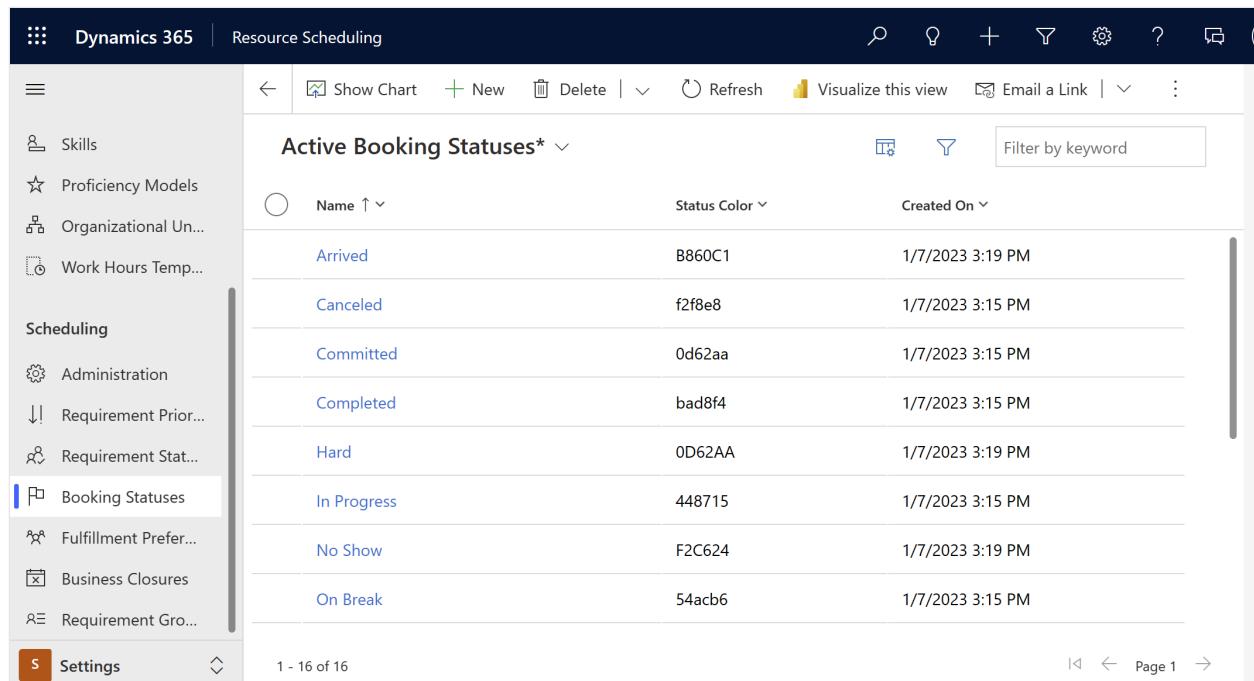
 No

[Provide product feedback ↗](#)

# Set up booking statuses

Article • 05/28/2024

Booking statuses allow you to create multiple substatuses mapped to each of your booking statuses in order to more precisely define your company's unique business processes. On the schedule board, booking statuses show in alphabetical order and grouped by their parent status. To change the order, update the name accordingly.



Name	Status Color	Created On
Arrived	B860C1	1/7/2023 3:19 PM
Canceled	f2f8e8	1/7/2023 3:15 PM
Committed	0d62aa	1/7/2023 3:15 PM
Completed	bad8f4	1/7/2023 3:15 PM
Hard	0D62AA	1/7/2023 3:19 PM
In Progress	448715	1/7/2023 3:15 PM
No Show	F2C624	1/7/2023 3:19 PM
On Break	54acb6	1/7/2023 3:15 PM

1. In the Field Service application, change to the **Resources** area.
2. Under **Booking Settings**, select **Booking Statuses**.
3. Select **New** on the **Active Booking Statuses** view.
4. Fill in your information.
  - Choose a custom **Status Color** value in the **Common** tab to specify which color the resource schedule time slot will be displayed on the schedule board.
  - Map the new status to a parent status in the **Field Service** tab by choosing a value in the **Field Service Status** field.

## Tip

You can also set a default status for scheduled and canceled bookings. Go to **Resources > Administration > Booking Setup Metadata Wizard**. Select

msdyn\_workorder and configure the default settings. For more information:

[Configure default settings](#)

---

## Feedback

Was this page helpful?

 Yes

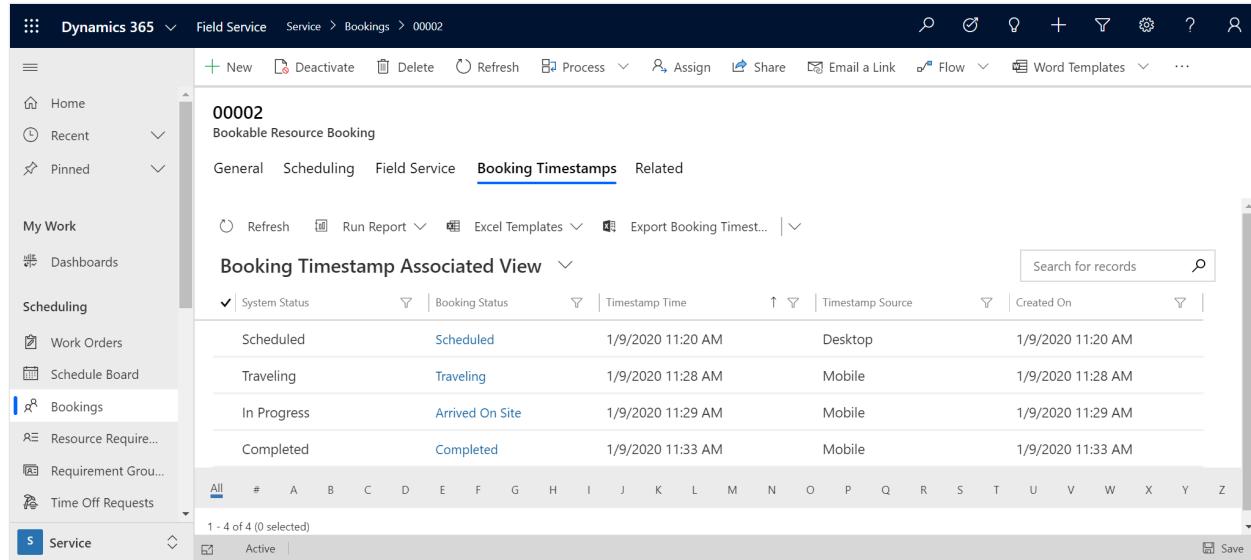
 No

[Provide product feedback !\[\]\(5d22036c339e10a9e6423d95a432e9e9\_img.jpg\)](#)

# Booking timestamps and booking journals

Article • 11/27/2024

Booking timestamps record the date and time that a *Booking Status* changes. They log how technicians spend their time, provide details for invoicing, and enable booking journals.



The screenshot shows the Dynamics 365 Field Service interface. The top navigation bar includes 'Dynamics 365', 'Field Service', 'Service > Bookings > 00002', and various action buttons like 'New', 'Deactivate', 'Delete', 'Refresh', 'Process', 'Assign', 'Share', 'Email a Link', 'Flow', 'Word Templates', and a search bar. The left sidebar has sections for 'Home', 'Recent', 'Pinned', 'My Work', 'Dashboards', 'Scheduling', 'Work Orders', 'Schedule Board', 'Bookings' (which is selected), 'Resource Requirements', 'Requirement Groups', 'Time Off Requests', and 'Service'. The main content area shows a 'Booking Timestamp Associated View' grid with columns: System Status, Booking Status, Timestamp Time, Timestamp Source, and Created On. The grid contains four rows of data corresponding to the booking status changes: Scheduled, Traveling, In Progress, and Completed. A 'Search for records' bar is at the top of the grid, and a navigation bar at the bottom shows '1 - 4 of 4 (0 selected)' and 'Active'.

## Tip

You can use booking timestamps to automatically generate [time entries](#). To enable that feature, [set the Field Service setting](#) Time Entry Generation Strategy to *Auto Generate from Booking Timestamps*.

## Prerequisites

Create or customize [booking statuses](#) that match your business and reporting needs. Booking statuses map to your [work order lifecycle and statuses](#).

## Create timestamps

By default, the system creates timestamps automatically when the booking status on an entity changes the related Field Service status. Admins can [change this behavior on the Field Service Settings](#) in the **Timestamp Frequency** field.

Think of a default work order process.

1. A dispatcher schedules a work order to a technician. When scheduling the booking, its booking status changes to *Scheduled*.
2. The technician sees the work order on their mobile device and updates the booking status to reflect progress. While they travel to the work site, they set it to *Traveling*.
3. When the work starts, they update it to *In Progress*.
4. After the technician completes the work, they set the booking status to *Completed*.

The system saves every change of the booking status to keep a track record.

### 💡 Tip

Bookings have a duration of at least one minute.

## View timestamps

To view timestamps, open a bookable resource booking and select **Related > Booking Timestamps**.

Each booking timestamp includes the following information:

- **System Status:** The related Field Service Status for the work order process.
- **Booking Status:** Booking status of the bookable resource booking.
- **Timestamp Time:** The date and time of status change.
- **Timestamp Source:** The device type originating the status change. When updating from the Dynamics 365 Field Service mobile app, the timestamp source is *Desktop* if the changes occurred while the app is running in online mode, and *Mobile* when the app is running in offline mode.

## Generate booking journals

Booking journals calculate total travel time and working time for a booking. When a bookable resource booking status changes to *Completed*, the system creates booking journals per the booking timestamps. If another user updates the booking status to *Completed* on behalf of a technician, the booking's end time preserves the previous end time value. *Canceled* bookings won't have any booking journal records.

There are a few types of booking journals:

- **Travel:** Duration when the resource is traveling to the site.

- **Working Hours:** Duration when the booking is in progress during the resource's work hours.
- **Break:** Duration when the resource is on break.
- **Overtime:** Duration when the booking is in progress outside of the resource's work hours.
- **Business Closure:** Duration when the booking is in progress business closure hours.

## Example of a booking journal

A resource's work hours are 8 AM to 3 PM. While completing a work order, they go through the following sequence.

[\[+\] Expand table](#)

Booking status	Time stamp
Scheduled	8:00 AM
Traveling	9:00 AM
In Progress	10.30 AM
On Break	12:00 PM
In Progress	1:00 PM
Completed	4:00 PM

Based on these time stamps, the system creates the corresponding booking journals.

[\[+\] Expand table](#)

Time frame	Duration	Booking journal type
Travel (9 AM - 10:30 AM)	90 minutes	Travel
In Progress (10:30 AM - 12 PM)	90 minutes	Working Hours
On Break (12 PM - 1:00 PM)	60 minutes	Break
In Progress (1:00 PM - 3:00 PM)	120 minutes	Working Hours
Exceeded working hours of resource (3:00 PM - 4:00 PM)	60 minutes	Overtime

## View booking journals

To view booking journals, open a bookable resource booking record and select **Related** > **Booking Journals**.

## Next steps

- [Work order summary report](#)
  - [Resource preferences](#)
- 

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) 

# Show appointments on the schedule board

Article • 05/28/2024

Organizations often use appointments in Dynamics 365 applications. For example, when a customer service rep creates a service appointment to help a customer with an issue.

You can use the schedule board to show [Dynamics 365 appointments](#) alongside bookings. Include appointments on the schedule board and during the scheduling process so dispatchers see the availability.

## Environment setting

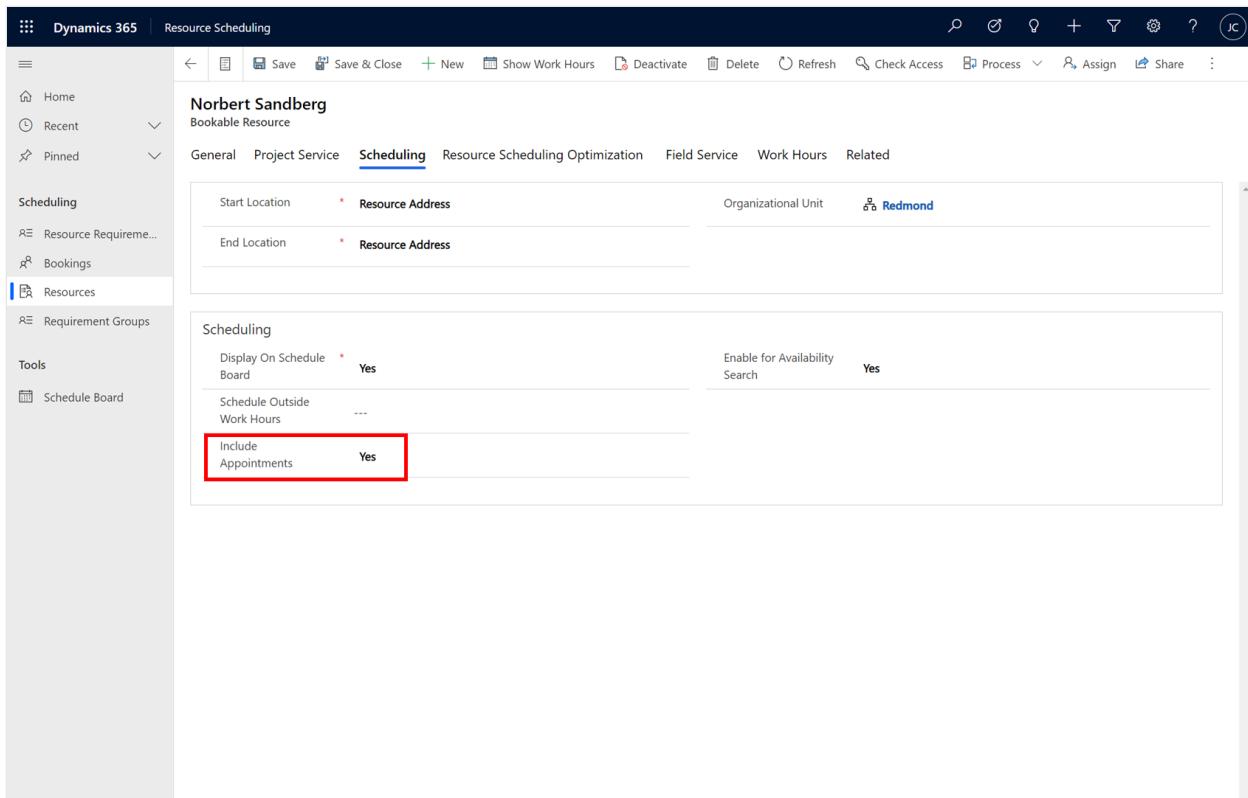
This setting enables appointment scheduling for all resources in your organization.

1. Open the **Resource Scheduling** app.
2. Change to the **Settings** area and go to **Administration > Scheduling Parameters**.
3. Set **Include Appointments** to **Yes**.
4. Confirm your change and select **Save & Close**.

## Resource setting

With appointment enabled on the environment, you can change the setting for individual resources.

Open a bookable resource form, select the **Scheduling** tab and set **Include Appointments** to **Yes**.



## Create an appointment

There are several ways to create Dataverse appointments. For example:

- Dynamics 365 Customer Service Hub, using Activities
- Dynamics 365 Sales Professional to manage tasks, appointments, email, or phone calls
- In Outlook, using [server-side synchronization](#) and the Tracked to Dynamics 365 category so it appears as a Dataverse appointment record.

## View appointments on the schedule board

Appointments on the schedule board are read-only and you can't move them. They show for all required attendees and the owner, if they're bookable resources. Hover over an appointment to see more details.

Appointments that are marked *Private* in Outlook and synced to Dynamics 365 won't show their subject on the schedule board.

## Customize appointment colors on the schedule board

The schedule board uses colors defined in [Dataverse appointment](#) metadata.

Customizing the **Color** of each **StatusCode** will change the appointment colors that show on the schedule board.

 **Note**

Customizing appointment tooltips on the Schedule Board is not supported.

## Hide canceled appointments on the schedule board

The [Show canceled setting in Board view settings](#) controls the visibility of canceled bookings and appointments.

## Automatically schedule around appointments

Resource Scheduling Optimization respects appointments when a planned optimization schedule runs, and for ad-hoc optimizations from the schedule board. The system considers appointments with statuses *Busy* and *Completed* as unavailable for scheduling operations.

Automated and assisted scheduling operations consider appointments as location-agnostic. If an appointment had related bookings, the system hides those bookings and won't consider them for scheduling operations.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Define intervals and time groups with fulfillment preferences

Article • 11/27/2024

Fulfillment preferences customize how the schedule assistant displays results for single-day requirements. By default, the schedule assistant shows results based on resource schedules and the earliest available time. Fulfillment preferences include intervals and time groups, or a combination of them.

## Intervals

Intervals define the length of the times slots the schedule assistant results are displayed in, and the start time for subsequent bookings. For example, with 30-minute intervals, the schedule assistant displays a resource available at 9:27 AM as available at 9:30 AM and books the start time (arrival time) for 9:30 AM. Travel time is included for onsite requirements and work orders, meaning travel time begins before 9:30 AM, and a field resource would arrive and start work at 9:30 AM.

## Time groups

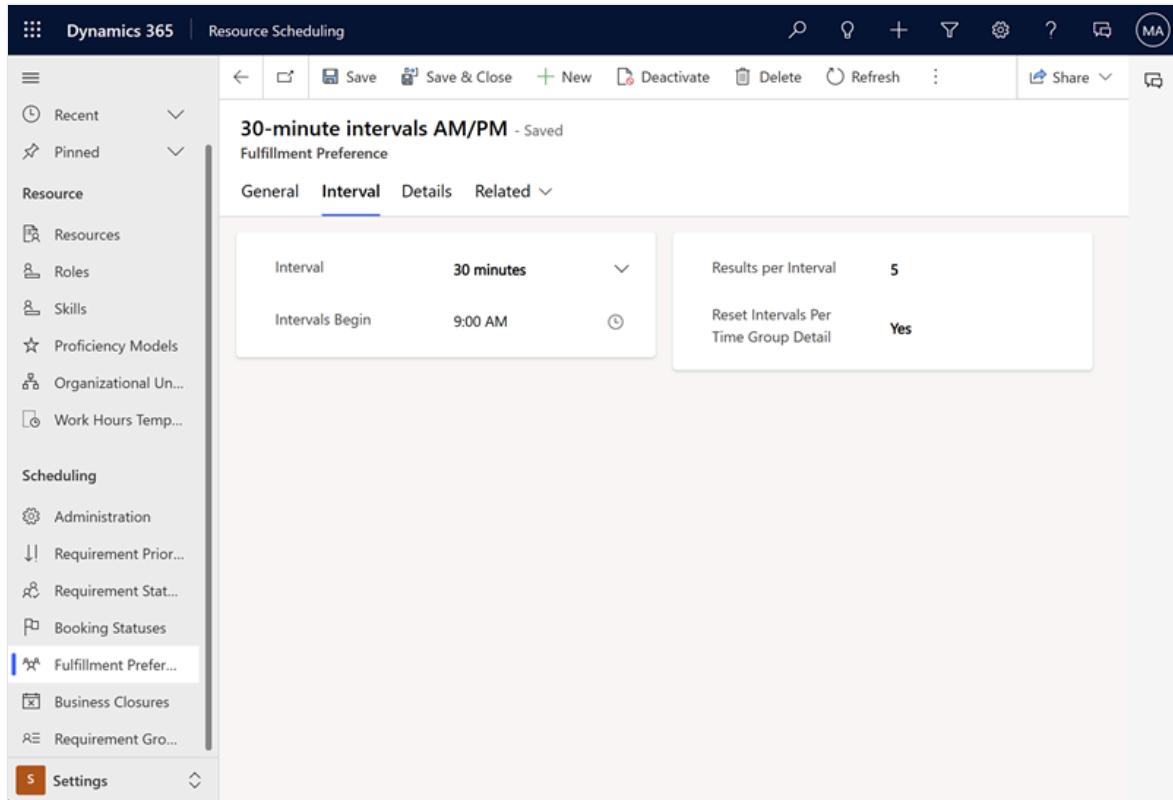
Time groups enable schedulers to search and view results as blocks of time when using the schedule assistant. Unlike intervals, time groups don't influence the start time of subsequent booking. Time groups organize results, but leave the start time/arrival time as-is, based on the particular resource's schedule.

For example, a heating and cooling company wants to group air-conditioning installations in two blocks. They create morning (8:00 AM to 12:00 PM) and afternoon (1:00 PM to 5:00 PM) blocks, with a one-hour break for lunch in between. They communicate a morning or afternoon time range to customers for when the installation crew arrives.

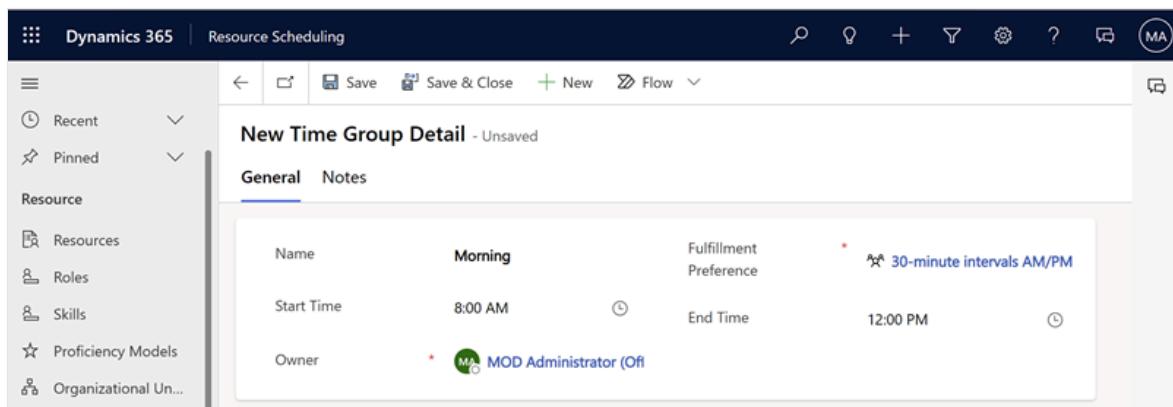
## Create a fulfillment preference

1. In Universal Resource Scheduling, open the **Settings** area. Under **Scheduling**, select **Fulfillment Preferences** and select **New**.
2. Enter a **Name** and **Save** the record.
3. To define an interval, go to the **Interval** tab.

- Choose an **Interval** to define the duration between two available time slots.
- Set the time for **Intervals begin** to define the start time for when options are available, depending on resource availability and working hours.
- Enter the number of **Results per Interval** to limit how many options the system presents for a time slot.



4. To add a time group, go to the **Details** tab and select **New Time Group Detail**. Enter a **Name** and choose a **Start Time** and **End Time**.



## Associate fulfillment preference to a requirement or requirement group

Open a requirement and go to the **Scheduling** tab to choose the **Fulfillment Preference**. For a requirement group, you can choose it in the **Fulfillment Preference** column.

## Book a requirement

Open a resource requirement and select **Book** to schedule the requirement. The start and end times match the fulfillment preference you defined for the requirement.

## Using intervals and time groups together

You can add both an interval and a time group to the same fulfillment preference, but first review these important considerations.

- Don't add a value for **Interval Begins**. The interval begins at the time of the earliest time group.
- If the **Reset Interval per Time Group Detail** option is set to **Yes**, the intervals reset once a new time group detail overlaps with an interval.

For example, a business offers appointments every 90 minutes, so intervals are set to 90 minutes. Additionally, they separate into morning and afternoon time groups of 8:00 AM to 12:00 PM and 1:00PM to 5:00 PM, with a one-hour lunch in between.

- If the **Reset Interval per Time Group Detail** is set to **No**, the appointments would be:

8:00 AM, 9:30 AM, 11:00 AM (not 12:30 PM because it's blocked for lunch), 2:00 PM, and 3:30 PM.

- If the **Reset Interval per Time Group Detail** is set to **Yes**, the appointments would be:

8:00 AM, 9:30 AM, 11:00 AM (not 12:30 PM because it's blocked for lunch), 1:00 PM (resetting for the next time group detail), 2:30 PM, and 4:00 PM.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback !\[\]\(4d0e8a6110a057cf149b615c463c4cef\_img.jpg\)](#)

# Set up booking rules

Article • 05/14/2024

Booking rules in Field Service create warning or error messages that users see when creating or editing a resource booking record, based on custom conditions. For example, a booking rule could be created to warn a user when they attempt to book a work order to a resource on the schedule board that doesn't have the skills required for the job.

Booking rules are custom JavaScript methods that will be executed prior to the bookable resource booking record being created or edited. The JavaScript method can accept a parameter that will contain information for the Bookable Resource Booking record being created and must return a JavaScript object with the required properties.

Set up booking rules to validate a booking when it's created or modified.

## Note

- Booking rules are only available for the hourly view, and not daily, weekly, or monthly views of the schedule board and schedule assistant. They are also available when a booking is created or updated via bookable resource booking form.
- Booking rules are not available on the bookable resource booking form, if it has business process flow enabled on the form.
- Booking rules are not available on the reassign functionality on the schedule board.
- Each custom booking rule can return only one error/warning. To return multiple messages, set up individual booking rules for each validation.

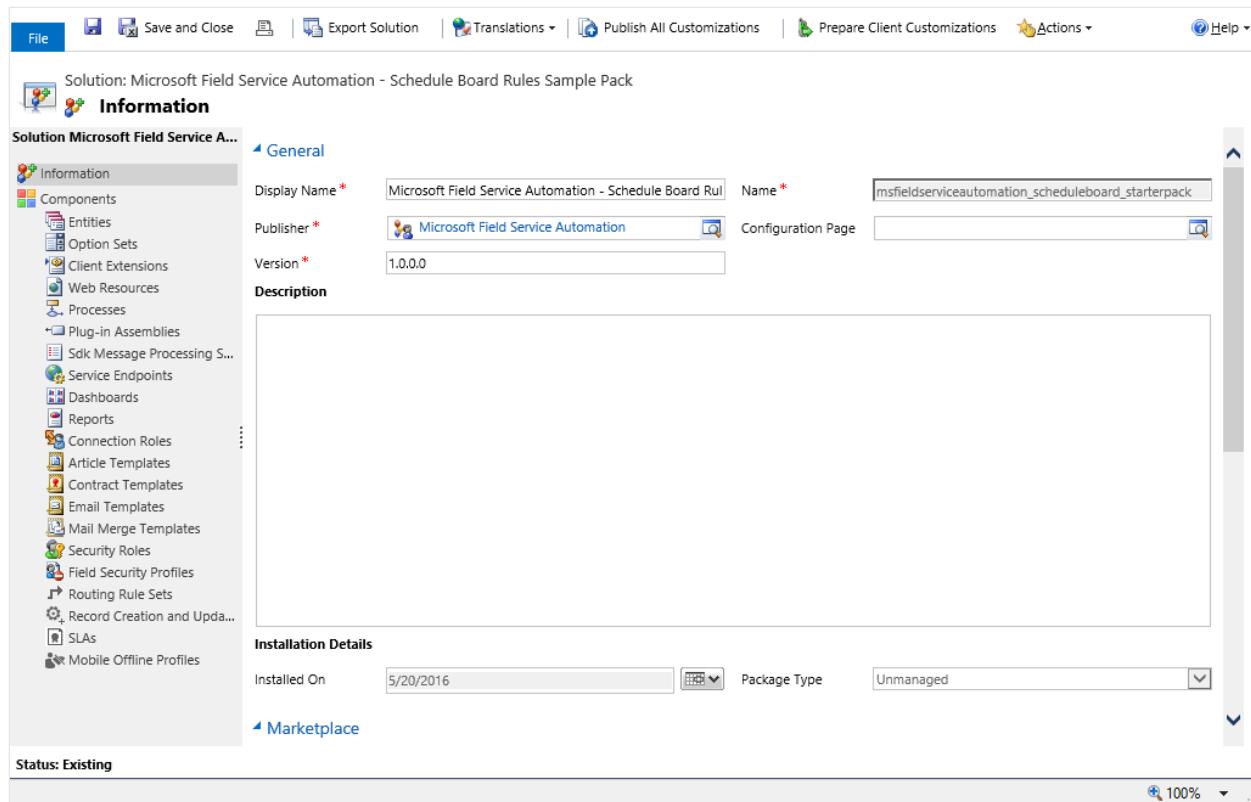
## Create a solution

The first step to setting up your booking rule is to create a custom JavaScript web resource. We recommend that you create a solution in CRM to add your custom JavaScript web resource or use an existing solution you may have for customizations.

Create CRM Solution

1. From **Settings > Solutions**, create a new solution for your booking rule JavaScript web resources.

The following screenshot shows a newly created solution. We recommend that your solution uses a unique publisher rather than the default publisher.



1. After creating the solution, select the Web Resources component and create a new web resource.
2. On the new web resource form, enter the following information: a. Name b. Display name c. Select **Script (Jscript)** as the type
3. Select the text editor option to enter your JavaScript code for the booking rule.
4. Select **Save** to save your web resource.
5. Select **Publish** to make sure the booking rule web resource has been published.

SAVE PUBLISH PUBLISH ALL CUSTOMIZA...

Solution: Microsoft Field Service Automation - Schedule Board Rules Sample Pack

**Web Resource: New**

**General**

Name\* msfsaeng\_ mybookingrule.js

Display Name mybookingrule.js

Description

**Content**

Type\* Script (JScript) Text Editor

Language

Upload File Browse...

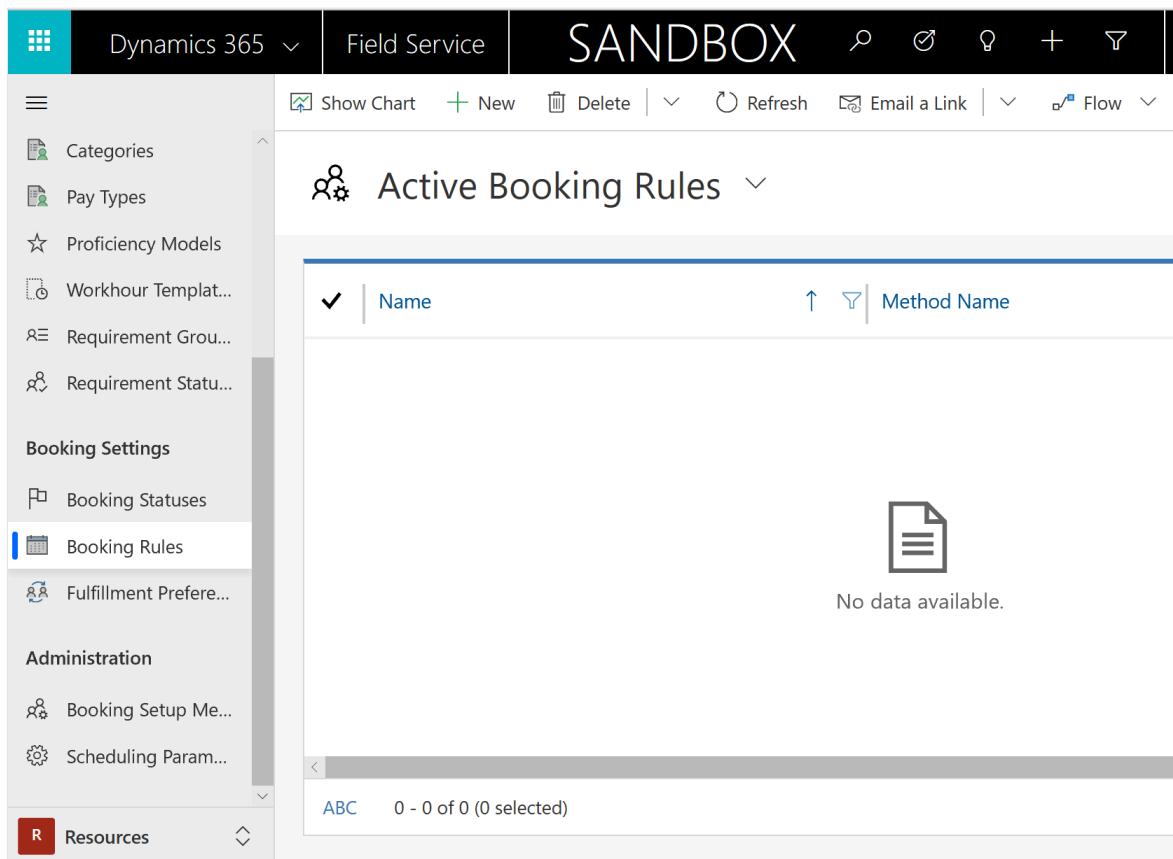
**URL**

URL

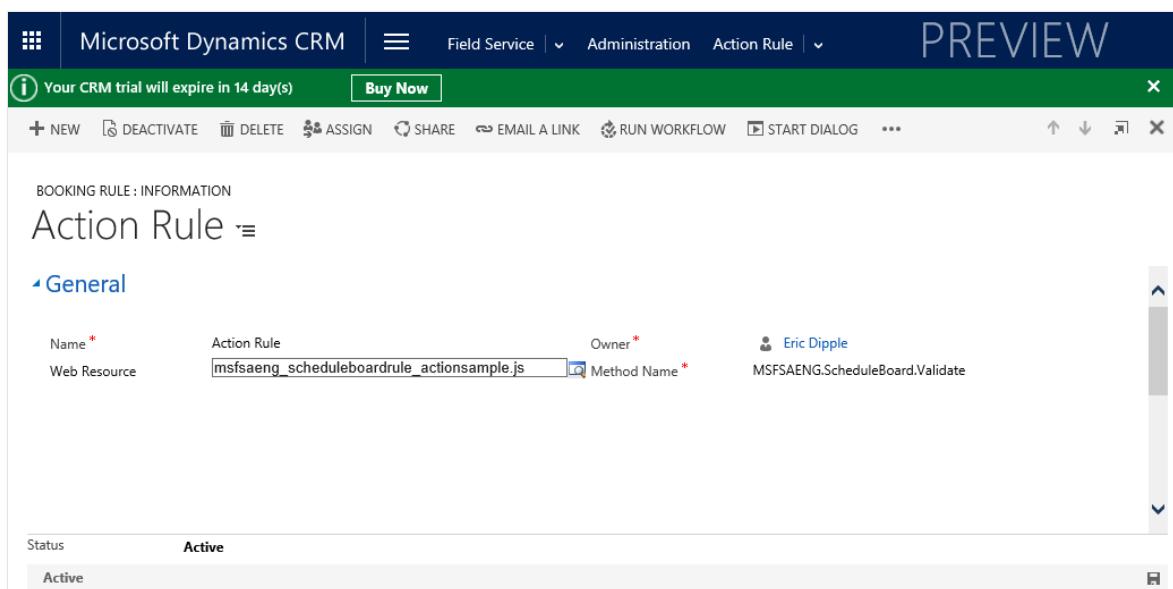
100%

## Set up a booking rule

1. From the main menu, go to **Field Service > Resources**, and then choose **Booking Rules** under **Booking Settings**.



2. Select **+New** to create a new booking rule.
3. From the booking rule form, enter the following information: a. Name b. Web resource (Select the web resource that you recently created). c. Enter the method name you defined in your JavaScript.



4. Save your booking rule. Once you save the booking rule, it will be used by the hourly view of schedule board and schedule assistant or the entity form. You can deactivate your booking rule record to keep the schedule board, schedule assistant, or the booking entity form, from executing the rule.

### (!) Note

The booking rules are currently only supported on the hourly view of the schedule board and schedule assistant. The booking rules are also supported when the bookings are created or updated using the bookable resource booking form. Booking rules do **not** execute on delete of a booking record. Booking rules don't work on forms when using multi-edit.

## Create a CRM action

In this section, we'll look at an example showing how you can use a custom CRM action to perform the validation as part of a booking rule.

When using a CRM action for the booking rule validation, you'll still need to create a custom web resource as defined above. The JavaScript that you'll define in your custom web resource will call the custom CRM Action and evaluate the results from the custom CRM action. See Attachment A at the end of this document for sample code that you can use to call a custom CRM action.

A custom CRM action will need to be created in CRM. We recommend that you use the CRM solution you've defined for your custom web resource to add your custom CRM action.

The custom CRM action should have the following input and output parameters. You can add more input and output parameters as your scenario requires. You'll need to ensure the JavaScript that you define to call your custom CRM action is updated to support your other input and output parameters.

Input parameters:

- originalScheduleStart – DateTime
- originalScheduleEnd – DateTime
- originalBookableResource – EntityReference
- originalScheduleSource – Picklist
- newScheduleStart – DateTime
- newScheduleEnd – DateTime
- isCreate – Boolean
- isUpdate – Boolean

Output Parameters:

- isError – Boolean

- isWarning – Boolean
- errorMessage – String
- warningMessage - String

The following screenshot shows an example custom CRM action. This sample is checking if the *newBookableResource* matches the preferred resource on the work order, and if the *newScheduleStart* is inside of the **Time From Promised** and **Time To Promised**. It's assumed the dates of the promised window are for a single date. Example: Time From Promised: 01/01/2016 8:00AM / Time To Promised: 01/01/2016 12:00PM.

The screenshot displays the configuration for a custom CRM action named "MS FSA Schedule Board Rule - Action 58".

**General Tab (Visible):**

- Process Name:** MS FSA Schedule Board Rule - Action 58
- Unique Name:** msfsaeng\_MSFSAScheduleBoardRuleAct
- Activate As:** Process
- Entity:** Work Order
- Category:** Action
- Enable rollback:**

**Workflow Log Retention:**  Keep logs for workflow jobs that encountered errors

**Hide Process Arguments:**

Name	Type	Required	Direction	Name *	Type *	Entity	Required	Direction	Description
originalScheduleStart	DateTime	Optional	Input	Name *	Boolean				
originalScheduleEnd	DateTime	Optional	Input	Type *					
originalBookableResource	EntityReference	Optional	Input	Entity					
originalScheduleSource	Picklist	Optional	Input	Required	<input type="checkbox"/>				
newScheduleStart	DateTime	Optional	Input	Direction	<input checked="" type="radio"/> Input <input type="radio"/> Output				
newScheduleEnd	DateTime	Optional	Input	Description					
newBookableResource	EntityReference	Optional	Input						
newScheduleSource	Picklist	Optional	Input						
isCreate	Boolean	Optional	Input						

**Validation Logic (Visible):**

- Default return parameter isError = false**  
Assign Value: isError [View properties](#)
- Default return parameter isWarning = false**  
Assign Value: isWarning [View properties](#)
- Perform Validations**

If Work Order:Preferred Resource contains data AND Work Order:Preferred Resource does not equal [newBookableResource (Bookable Resource):Bookable Resource], then:

- Preferred Resource Failed - Set isError = true**  
Assign Value: isError [View properties](#)
- Preferred Resource Failed - Set errorMessage**  
Assign Value: errorMessage [View properties](#)

Otherwise, if Work Order:Time From Promised > [Arguments-newScheduleStart] OR Work Order:Time To Promised < [Arguments-newScheduleStart], then:

- Promise Window Failed - Set isError = true**  
Assign Value: isError [View properties](#)
- Promise Window Failed - Set errorMessage**  
Assign Value: errorMessage [View properties](#)

Otherwise:

- For Demo: set isWarning = true**  
Assign Value: isWarning [View properties](#)
- For Demo: set warningMessage**  
Assign Value: warningMessage [View properties](#)

## Sample code

The JavaScript function you created can accept a single parameter, which is considered the booking context. The passed booking context parameter isn't a typical CRM context used in client-side scripting.

Booking context schema:

```
export type BookingRuleContext = {
  oldValues: BookingRuleBookingRecord;
  newValues: BookingRuleBookingRecord;
  isCreate: boolean;
  isUpdate: boolean;
};

export type BookingRuleBookingRecord = {
  ResourceRequirementId?: string;
  ResourceId?: string;
  StartTime?: Date;
  EndTime?: Date;
  ResourceScheduleSource?: string;
};
```

The booking context parameter will have the following JavaScript definition.

 **Note**

It's *not* necessary to include this JavaScript code in the custom web resource for the booking rule.

The possible values for *ResourceScheduleSource* are from the resource schedule source global option set. You can make use of this property to know if the booking rule is being triggered from the schedule board or scheduling assistant.

```
var sbContext = {
  oldValues: {
    StartTime: "01/01/2016 08:00AM",
    EndTime: "01/01/2016 05:00PM",
    ResourceId: "00000000-0000-0000-0000-00000000",
    ResourceScheduleSource: 690970001
  },
  newValues: {
    StartTime: "01/01/2016 08:00AM",
    EndTime: "01/01/2016 05:00PM",
    ResourceId: "00000000-0000-0000-0000-00000000",
    ResourceScheduleSource: 690970001
  },
  isCreate: true,
  isUpdate: false
};
```

Your validation method must return a JavaScript object with the following definition.

## (!) Note

It's *not* necessary to include this JavaScript code in the custom web resource for the booking rule.

```
var ruleResult = {
  IsValid: false,
  Message: "Some Message Here",
  Type: "error" // this can be either "error" or "warning"
};
```

Example JavaScript function definition. The following JavaScript code is the only JavaScript code you'll need to include in your custom web resource.

```
function Validate(ctx) {
  var url = Xrm.Page.context.getClientUrl();
  var ruleResult = {
    IsValid: false,
    Message: '',
    Type: 'error'
  };

  //
  // perform some lookups or other validation logic here.
  //

  ruleResult.IsValid = false;
  ruleResult.Message = 'Some Error Message Here.';
  ruleResult.Type = 'error';

  return ruleResult;
}
```

The following JavaScript can be used to call a custom CRM action that has the same input and output parameters as the previous sample.

On the booking rule record, the **Method Name** must be:

*MSFSAENG.ScheduleBoard.Validate*. For reference, see the screenshot in the "Setting Up A Booking Rule" section of this article.

```

/// <reference path="xrm.d.ts" />
function brErrorCallback(sb) {
// Add custom error handeling here if desired.
    return;
}
function brWarningCallback(sb) {
// Add custom warning handeling here if desired.
    return;
}
function brSuccessCallback(sb) {
// add custom sucess handeling here if desired.
    return;
}
var MSFSAENG;
(function (MSFSAENG) {
MSFSAENG.ScheduleBoard = {
    url: Xrm.Page.context.getClientUrl() + "/api/data/v8.1/",
    actionName: "msfsaeng_MSFSAScheduleBoardRuleActionSample",
    actionInputParameters: function (ctx) {
        var inputParameters = {};
        if (ctx.isUpdate) {
            inputParameters = {
                "originalScheduleStart": ctx.oldValues.StartTime,
                "originalScheduleEnd": ctx.oldValues.EndTime,
                "originalBookableResource": {
                    "@odata.type": "Microsoft.Dynamics.CRM.bookableresource",
                    "bookableresourceid": ctx.oldValues.ResourceId,
                    "name": ""
                },
                "originalScheduleSource": ctx.oldValues.ResourceScheduleSource,
                "newScheduleStart": ctx.newValues.StartTime,
                "newScheduleEnd": ctx.newValues.EndTime,
                "newBookableResource": {
                    "@odata.type": "Microsoft.Dynamics.CRM.bookableresource",
                    "bookableresourceid": ctx.newValues.ResourceId,
                    "name": ""
                },
                "newScheduleSource": ctx.newValues.ResourceScheduleSource,
                "isCreate": ctx.isCreate,
                "isUpdate": ctx.isUpdate
            };
        }
        else {
            inputParameters = {
                "newScheduleStart": ctx.newValues.StartTime,
                "newScheduleEnd": ctx.newValues.EndTime,
                "newBookableResource": {
                    "@odata.type": "Microsoft.Dynamics.CRM.bookableresource",

```

```

        "bookableresourceid": ctx.newValues.ResourceId,
        "name": ""
    },
    "newScheduleSource":
ctx.newValues.ResourceScheduleSource,
        "isCreate": ctx.isCreate,
        "isUpdate": ctx.isUpdate
    };
}
return JSON.stringify(inputParameters);
},
ctx: null,
ruleResult: {
    IsValid: true,
    Message: "",
    Type: ""
},
outputParameters: {
    isError: false,
    isWarning: false,
    errorMessage: "",
    warningMessage: ""
},
Validate: function (context) {
    this.ctx = context;
    ScheduleBoardHelper.callActionWebApi(this);
    return this.ruleResult;
},
errorCallback: brErrorCallback,
warningCallback: brWarningCallback,
successCallback: brSuccessCallback
};
var ScheduleBoardHelper = (function () {
    function ScheduleBoardHelper() {
    }
    ScheduleBoardHelper.callActionWebApi = function (sb) {
        var oDataEndpoint = sb.url + sb.actionName;
        var req = new XMLHttpRequest();
        req.open("POST", oDataEndpoint, false);
        req.setRequestHeader("Accept", "application/json");
        req.setRequestHeader("Content-Type", "application/json;
charset=utf-8");
        req.setRequestHeader("OData-MaxVersion", "4.0");
        req.setRequestHeader("OData-Version", "4.0");
        req.onreadystatechange = function () {
            if (req.readyState == 4) {
                req.onreadystatechange = null;
                if (req.status == 200) {
                    sb.outputParameters = JSON.parse(req.response);
                    if (sb.outputParameters.isError) {
                        sb.ruleResult.IsValid = false;
                        sb.ruleResult.Message =
sb.outputParameters.errorMessage;
                        sb.ruleResult.Type = 'error';
                        if (sb.errorCallback)

```

```

                sb.errorCallback(sb);
                return;
            }
            else if (sb.outputParameters.isWarning) {
                sb.ruleResult.IsValid = false;
                sb.ruleResult.Message =
sb.outputParameters.warningMessage;
                sb.ruleResult.Type = 'warning';
                if (sb.warningCallback)
                    sb.warningCallback(sb);
                return;
            }
            else {
                sb.ruleResult.IsValid = true;
                sb.ruleResult.Message = '';
                sb.ruleResult.Type = '';
                if (sb.successCallback)
                    sb.successCallback(sb);
                return;
            }
        }
        else {
            alert('Error calling Rule Action. Response = ' +
req.response + ', Status = ' + req.statusText);
        }
    }
};

req.send(sb.actionInputParameters(sb.ctx));
};

return ScheduleBoardHelper;
}();
})(MSFSAENG || (MSFSAENG = {}));

```

## Additional notes

The bookable resource booking is enabled to use booking rules to create warnings or error messages that users see when creating or editing a resource booking record, based on custom conditions. The system uses [preventDefault in booking rules](#).

Therefore, business process flows and other custom scripts bond to the `onSave` event can't be used on the bookable resource booking entity with booking rules enabled.

However, the processing of booking rules can be disabled on the save of the Booking form by enabling the below setting, which would let the users use the business process flows. The client side APIs can be used to enable this setting at an environment level.

Read current value of the setting `msdyn_DisableProcessBookingRulesOnSaveBookingForm`.

```
Xrm.Utility.getGlobalContext().getCurrentAppSettings()  
["msdyn_DisableProcessBookingRulesOnSaveBookingForm"]
```

Enable the setting `msdyn_DisableProcessBookingRulesOnSaveBookingForm`.

```
Xrm.Utility.getGlobalContext().saveSettingValue("msdyn_DisableProcessBooking  
RulesOnSaveBookingForm",true,).then(() => {a = "success"}, (error) => {a =  
error})
```

Disable the setting `msdyn_DisableProcessBookingRulesOnSaveBookingForm`.

```
Xrm.Utility.getGlobalContext().saveSettingValue("msdyn_DisableProcessBooking  
RulesOnSaveBookingForm",false,).then(() => {a = "success"}, (error) => {a =  
error})
```

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) ↗

# Edit work hour calendars by using APIs

Article • 12/02/2024

Organizations often need to programmatically create, edit, or delete work hours on the calendars of their resources. Calendars show working hours, time off, and breaks that determine the availability of a resource when work is being scheduled. Those resources must be scheduled in specific time zones, might or might not observe business closures, and can have variable capacity. For information about defining work hours in the Field Service app, go to [Add work hours to a bookable resource](#).

In addition to using the Field Service app, you can use the following APIs to modify calendar rules for selected record types:

- The Save Calendar API (`msdyn_SaveCalendar`) creates or updates calendar records on a selected entity, based on the inputs passed as the request.
- The Delete Calendar API (`msdyn_DeleteCalendar`) deletes all [inner calendar rules](#) of a calendar on a selected entity, based on the inputs passed as the request.
- The Save/Delete Calendar API V2 (`msdyn_SaveCalendar/msdyn_DeleteCalendar`, pass flag `UseV2`) allows multiple work hour recurrences simultaneously by altering the logic for overlapping rules. For more information, see [What happens if there are overlapping rules?](#).

This article contains details about each API's input (request) and output (response), and their usage, with examples.

## Prerequisites

- Platform version 9.2.21055 or above with Universal Resource Scheduling version 3.12.45.7.
- Usage of one of the following record types:
  - Bookable resource (`bookableresource`)
  - Resource requirement (`msdyn_resourcerequirement`)
  - Work hour template (`msdyn_workhourtemplate`)
  - Project (`msdyn_project`)

## Calendar event types

When you create a calendar, you define how many times a [work hour type](#) occurs—once, all day, every week, or every day, or you can create a custom recurrence. For more information about these calendar events, go to the [examples later in this article](#).

## Occurrence

When a work hour type occurs only once in the entity's calendar, it's called an *occurrence*.

For example, consider a resource working from 5:00 AM to 10:00 AM on May 26, 2021. These APIs only support this type of occurrence, which starts and ends within the same day. For another example, consider a resource working from May 26, 2021 at 8:00 PM 10:00 AM on May 27, 2021. You can't create this occurrence by using just one call of the `msdyn_SaveCalendar` API; you need to make two calls instead.

## All-day occurrence

When a work hour type occurs for one or more whole days, starting at midnight (12:00 AM) of the start date, it's an *all-day occurrence*. The maximum duration for an all-day occurrence is five years.

For example, a resource works all day from May 26, 2021 to the end of the day on May 30, 2021. This is an all-day occurrence that lasts five days.

## Weekly recurrence

When a work hour type occurs at the same time on selected days of every week, it's called a *weekly recurrence*.

For example, a resource works from 5:00 AM to 10:00 AM every Monday, Tuesday, and Wednesday.

## Daily recurrence

When a work hour type occurs at the same time every day, it's called a *daily recurrence*.

For example, a resource works from 5:00 AM to 10:00 AM every day of the week.

## Custom recurrence

When a work hour type occurs at certain times on certain days of the week, but the times are different on different days, you can create a *custom recurrence*.

For example, a resource works from 5:00 AM to 10:00 AM every Monday, and 12:00 PM to 3:00 PM every Wednesday.

# Work hour types

These APIs support create, update, and delete operations for the following work hour types:

- [Working hours](#)
- [Nonworking hours](#)
- [Breaks](#)
- [Time off](#)
- [Business closure](#)

## Working hours

Working hours are times during which an entity is available to perform work.

Using these APIs, you can do the following:

- Create, edit, or delete a working hour occurrence.
- Create, edit, or delete a working hour daily recurrence.
- Create, edit, or delete a working hour weekly recurrence.
- Create, edit, or delete a working hour custom recurrence.
- Create, edit, or delete all-day working hours.
- Create or edit capacity on working hours.
- Edit a single working hour occurrence in a recurrence.
- Edit *This and Following occurrences* in a recurrence.
- Change a working hour occurrence to a recurrence.
- Change the time zone of the calendar rule.

Using this API, you can't do the following:

- Delete a single working hour occurrence from a recurrence.
- Create an occurrence that spans 24 hours but doesn't start and end at midnight (12:00 AM).
- Create, edit, or delete an all-day recurrence.

## Nonworking hour

These are times during which the entity is unavailable to work due to an unspecified reason.

Using these APIs, you can do the following:

- Create or edit all-day nonworking hours.

- Create or edit a nonworking hour occurrence.
- Change the time zone of the calendar rule.

Using these APIs, you can't do the following:

- Create or edit a nonworking hour recurrence.

## Break

These are times in a working day during which an entity is taking a break and is unavailable for work. Breaks can't exist without working hours; they have to occur between two working hours in a day. Breaks can't overlap with working hours.

Using these APIs, you can do the following:

- Create or edit breaks during working hours.

Using these APIs, you can't do the following:

- Delete only breaks from an occurrence or recurrence of working hours.

## Time off

These are times during which an entity is unavailable to work due to a vacation. The reason for time off can be specified.

Using these APIs, you can do the following:

- Create or edit time off with a label.
- Change the time zone of the calendar rule.

Using these APIs, you can't do the following:

- Create or edit a time-off hour recurrence.

## Business closure

You can [create business closure entities](#) that define the times the business is closed.

Using the `msdyn_SaveCalendar` API, you can set every entity to observe or ignore the organization's business closure times by using the optional **ObserveClosure** key. When they're set to observe these closures, the entities are unavailable for work.

# Save Calendar API

## Input

The request contains only one attribute—**CalendarEventInfo**, which is a **String** type. It contains several other attributes that are all embedded in this string.

### ⓘ Note

In the following table, **Type** represents the format expected to make a successful request. However, the whole request is parsed as a single string.

## CalendarEventInfo

[\[+\] Expand table](#)

Name	Type	Required	Description
EntityLogicalName	String	Yes	This key describes the entity from which the API is being called. The calendar of this entity is to be created or edited.
CalendarId	GUID	Yes	This key contains the calendar ID related to the entity described above. When any of these entities are created, a calendar record is also automatically created. These APIs edit that calendar record by adding rules or editing existing rules.
RulesAndRecurrences	RulesAndRecurrences	Yes	This key is an array, and each element contains multiple attributes as listed in the table in the following section. The size of the array should be at least one.
IsVaried	Boolean	No	This key should be set to <code>true</code> for custom recurrence scenarios.
IsEdit	Boolean	No	This key should be set to <code>true</code> for editing existing rules.
TimeZoneCode	Integer	No	This key takes an integer value corresponding to the time zone for the calendar rules. For the mapping, go to <a href="#">Time zone codes</a> ,

Name	Type	Required	Description
			later in this article. The default value is the user's time zone.
InnerCalendarDescription	String	No	This key is only needed if the calendar rule is for time off. It should contain the reason for the time off.
ObserveClosure	Boolean	No	This key is specific to recurrences. If it's set to <code>true</code> , the entity observes business closure.
RecurrenceEndDate	DateTime	No	This key is specific to recurrences. It contains the end date for the recurrence. If the timestamp is 08:00:00 or earlier, the recurrence end date is one day before the specified date. If the timestamp is 08:00:01 or later, the date is respected as-is. The default value for occurrences is null. The default value for recurrences is 30 Dec 9999, 23:59:59 hours, UTC.
RecurrenceSplit	Boolean	No	This key is specific to recurrences. It's set to <code>true</code> for editing "This and following occurrences" of a recurrence.
ResourceId	GUID	No	This key contains the <b>SystemUserId</b> or <b>ResourceId</b> and is only to be passed when the entity associated with this call is a bookable resource of type <b>SystemUser</b> . This is necessary to check for <b>OwnCalendar</b> privileges on the <b>Service Management</b> tab.
UseV2	Flag	No	Passing this flag enables the V2 version of the work hour calendar, with an enhanced overlapping rules logic allowing for multiple recurrences. For more information, see <a href="#">What happens if there are overlapping rules?</a>

## RulesAndRecurrences

[Expand table](#)

Name	Type	Required	Description
Rules	Rules	Yes	This key is an array, and each element contains multiple attributes as listed in the table in the following section. The size of the array should be at least one.
RecurrencePattern	String	No	This key is specific to recurrences. We currently only support this pattern: <code>FREQ=WEEKLY; INTERVAL=1; BYDAY=SU, MO, TU, WE, TH, FR, SA.</code> <code>BYDAY</code> can be changed to include fewer days; however, <code>FREQ</code> and <code>INTERVAL</code> can't be changed.
InnerCalendarId	GUID	No	This key is specific to editing. If a rule is being edited, the <code>InnerCalendarId</code> needs to be passed here. If an <code>InnerCalendarId</code> isn't passed, the API creates a new rule, even if the <code>IsEdit</code> key is set to true.
Action	Integer	No	This key is specific to custom recurrences. If a custom recurrence is being created or edited, one of the following numbers should be entered: <ul style="list-style-type: none"><li>• (1) Adding a day to the recurrence</li><li>• (2) Deleting a day from the recurrence</li><li>• (3) Editing only the start or end dates or times, or editing capacity</li><li>• (4) Editing anything other than the keys mentioned in (3)</li></ul>

## Rules

[Expand table](#)

Name	Type	Required	Description
StartTime	DateTime	Yes	This key contains a datetime entry in <a href="#">ISO format</a> . For example, <code>\"2021-05-15T12:00:00.000Z\"</code> . The time portion determines the start time of the work hour in the time zone specified earlier. The date portion determines the start date of the work hour. Here, May 15, 2021 is the date of the occurrence or the starting date of the recurrence. If the pattern was <code>BYDAY=TU, WE</code> , but May 15 (a Saturday) is the date, the API will automatically create or edit rules for all Tuesdays and Wednesdays following May 15. This is case where the rule doesn't have to have the date corresponding to the day.

Name	Type	Required	Description
EndTime	DateTime	Yes	<p>This contains a datetime entry in <a href="#">ISO format</a>. For example, <code>\"2021-05-15T12:00:00.000Z\"</code>. The time portion determines the end time of the work hour in the time zone specified earlier. The date portion <i>must</i> contain the same date as the date portion of the <b>StartTime</b>. The only exceptions are:</p> <ul style="list-style-type: none"> <li>• If it's an all-day occurrence. In this case, the date portion should reflect the end date of the all-day occurrence.</li> <li>• The occurrence ends at the end of the day, that is, 12:00 AM of the following day. In this case, the date should be <code>\"2021-05-16T00:00:00.000Z\"</code>. To specify the end date of the recurrence, modify the <b>RecurrenceEndDate</b> attribute.</li> </ul>
WorkHourType	Integer	Yes	<p>This key contains a number corresponding to one of the following options:</p> <ul style="list-style-type: none"> <li>• (0) Working</li> <li>• (1) Break</li> <li>• (2) Nonworking</li> <li>• (3) Time Off</li> </ul>
Effort	Integer	No	<p>This key determines the capacity of the entity. It must be a whole number. The default value is 1.</p>

## Output

This POST API creates or modifies calendar rule records for the selected entity. It also gives the following output.

[\[+\] Expand table](#)

Name	Type	Description
InnerCalendarIds	String	An array of <b>InnerCalendarIds</b> GUIDs that are a result of the POST operation.

## Delete Calendar API

### Input

[\[+\] Expand table](#)

Name	Type	Required	Description
EntityLogicalName	String	Yes	This field describes the entity whose calendar rules are to be deleted.
InnerCalendarId	GUID	Yes	This field describes the ID of the <b>InnerCalendarId</b> that needs to be deleted. If there are multiple <b>InnerCalendarIds</b> associated with a single rule, any one ID is sufficient here. More information about inner and outer calendars: <a href="#">Calendar entities</a>
CalendarId	GUID	Yes	This field describes the <b>CalendarId</b> of the entity.
IsVaried	Boolean	No	This field is specific to recurrences, and is set to <code>yes</code> if a custom recurrence rule is being deleted.
UseV2	Flag	No	Passing this flag enables the V2 version of the work hour calendar, with an enhanced overlapping rules logic allowing for multiple recurrences. For more information, see <a href="#">What happens if there are overlapping rules?</a> .

## Output

This POST API deletes calendar rule records for the selected entity. Additionally, it gives the following output.

[\[+\] Expand table](#)

Name	Type	Description
InnerCalendarIds	String	An array of <b>InnerCalendarIds</b> GUIDs that are a result of the POST operation.

## Load Calendar API

### Input

Name: msdyn\_LoadCalendars

Type: Action

Description: Returns calendars for given LoadCalendarsInput.

Name: msdyn\_LoadCalendars.LoadCalendarsInput

Type: Parameter

Description: String in the following JSON format:

JSON

```
{  
  StartDate: string,  
  EndDate: string,  
  CalendarIds: string[]  
}
```

Name: msdyn\_LoadCalendarsResponse

Type: ComplexType

Description: Contains the response from the msdyn\_loadCalendars action.

Name: msdyn\_LoadCalendarsResponse.CalendarEvents

Type: Property

Description: String in the following JSON format:

JSON

```
{  
  "calendarId": CalendarEventSlot[]  
}
```

Where calendarId is a proper guid representing Guid of the Calendar and

CalendarEventSlot is an object of following format:

JSON

```
{  
  CalendarId: string,  
  InnerCalendarId: string,  
  Start: string,  
  End: string,  
  Effort: double  
}
```

## How to call the APIs

These APIs can be called by using the browser.

1. Open the browser and the org in which you need to make these calendar changes.

2. Open Developer Tools (select **Ctrl+Shift+I** in Microsoft Edge, select **F12** in Google Chrome).
3. In the console, enter the following function, after replacing [org-name] with org details (for example, `http://your_org.crm.dynamics.com`):

```

function CalendarAction(action, data) {
    let req = new XMLHttpRequest();
    req.open("POST", "**[org-name]**/api/data/v9.0/" + action, true);
    req.setRequestHeader("OData-MaxVersion", "4.0");
    req.setRequestHeader("OData-Version", "4.0");
    req.setRequestHeader("Accept", "application/json");
    req.setRequestHeader("Content-Type", "application/json;
charset=utf-8");
    req.setRequestHeader("Prefer", 'odata.include-annotations="*"');
    req.onreadystatechange = function () {
        if (this.readyState === 4) {
            req.onreadystatechange = null;
            if(this.status == 200) {
                console.log(JSON.parse(this.response));
            } else {
                console.error(JSON.parse(this.response));
            }
        }
    };
    req.send(JSON.stringify(data));
}

```

4. After this function is defined, you can call it to create, edit, or delete calendars by using the APIs. Enter the following call to save a calendar:

```

CalendarAction("msdyn_SaveCalendar",{
    "CalendarEventInfo"://{
        \"CalendarId\": \"df0857c4-50f5-4576-8e89-f236670ad2d5\",
        \"EntityLogicalName\": \"bookableresource\",
        \"TimeZoneCode\": 92, \"StartDate\": \"2021-04-
25T00:00:00.000Z\",
        \"IsVaried\": false,
        \"RulesAndRecurrences\":[{
            \"Rules\":[{
                \"StartTime\": \"2021-04-25T08:00:00.000Z\",
                \"EndTime\": \"2021-04-25T17:00:00.000Z\",
                \"Duration\": 540,
                \"Effort\": 1
            }]
        }]
    }
}

```

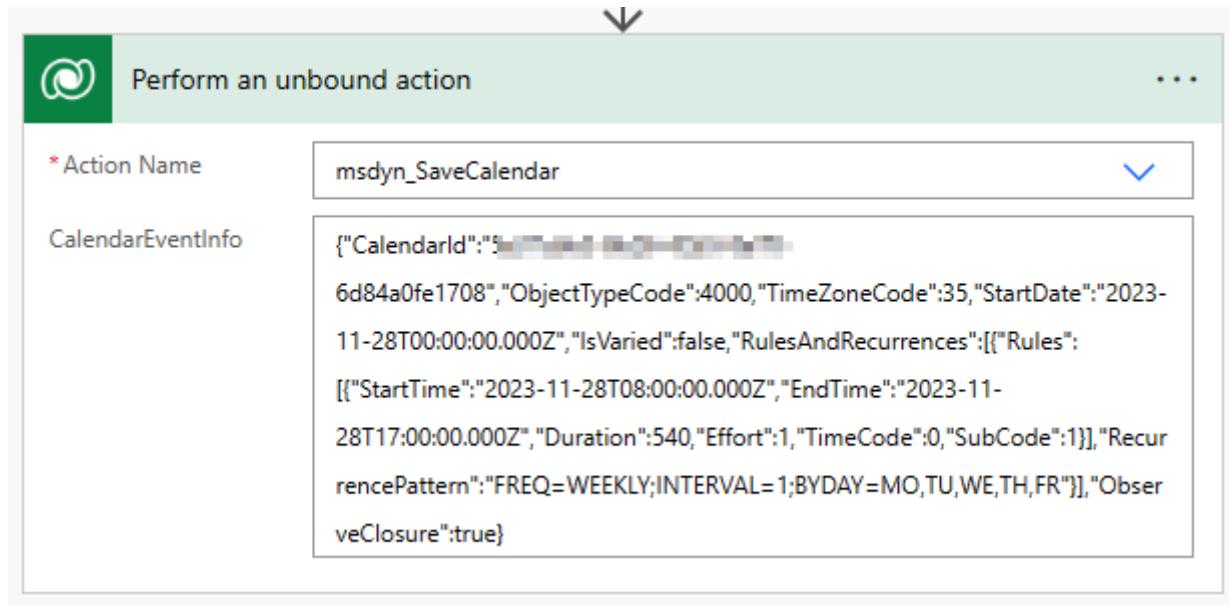
```
  }"  
})
```

Enter the following call to delete a calendar:

```
CalendarAction("msdyn_DeleteCalendar", {  
    "CalendarEventInfo": {  
        "\\"CalendarId\\":\\"8390358c-77d0-430f-b176-f27adadac8eb\\",  
        "\\"EntityLogicalName\\":\\"bookableresource\\",  
        "\\"InnerCalendarId\\":\\"cf508c2c-5c55-485c-be1e-d2ebcb385441\\"  
    }  
})
```

See the following section for examples of how to make different calls based on your needs. Replace the `action` of the function call in step 3 with `msdyn_SaveCalendar` or `msdyn_DeleteCalendar`, and replace `data` with the relevant `CalendarEventInfo`.

Also see the following screenshot for a Power Automate call for `msdyn_SaveCalendar` action:



## Example scenarios for API usage

Let's walk through some scenarios that you can use these APIs for.

Bob and Tim are delivery truck drivers for Contoso Enterprises in Bellevue, Washington. Their dispatcher, Debbie, is responsible for making changes to their work hour calendars. Debbie makes these changes by using the `msdyn_SaveCalendar` and `msdyn_DeleteCalendar` APIs.

## Create a working hour occurrence.

Bob is scheduled to drive around to deliver packages from 9:00 AM to 5:00 PM on May 15, 2021. Debbie uses the `msdyn_SaveCalendar` API.

### Request

```
{  
  "CalendarEventInfo": "{\"CalendarId\": \"d33263c7-c16b-4e3e-a56a-  
  20f7a66cafc1\", \"EntityLogicalName\": \"bookableresource\", \"TimeZoneCode\": 5  
  , \"RulesAndRecurrences\": [{\"Rules\": [{\"StartTime\": \"2021-05-  
  15T09:00:00.000Z\", \"EndTime\": \"2021-05-  
  15T17:00:00.000Z\", \"Effort\": 1, \"WorkHourType\": 0}]}]}"  
}
```

### Response

```
{  
  "InnerCalendarIds": ["f76cc333-cbbe-eb11-a81d-000d3a6e4359"]  
}
```

## Edit a working hour occurrence.

Bob's schedule then changes to start at 10:00 AM on May 15, 2021. Debbie uses the `msdyn_SaveCalendar` API.

### Request

```
{  
  "CalendarEventInfo": "{\"CalendarId\": \"d33263c7-c16b-4e3e-a56a-  
  20f7a66cafc1\", \"EntityLogicalName\": \"bookableresource\", \"IsEdit\": \"true\"  
  , \"TimeZoneCode\": 5, \"RulesAndRecurrences\": [{\"Rules\":  
  [{\"StartTime\": \"2021-05-15T10:00:00.000Z\", \"EndTime\": \"2021-05-  
  15T17:00:00.000Z\", \"Effort\": 1, \"WorkHourType\": 0}],  
  \"InnerCalendarId\": \"f76cc333-cbbe-eb11-a81d-000d3a6e4359\"}]}"  
}
```

### Response

```
{  
  "InnerCalendarIds": "[\"f76cc333-cbbe-eb11-a81d-000d3a6e4359\"]"  
}
```

## Delete a working hour occurrence.

A family emergency comes up, and Bob needs to cancel an entire day of work. Debbie uses the `msdyn_DeleteCalendar` API.

### Request

```
{  
  "CalendarEventInfo": "{\"CalendarId\": \"d33263c7-c16b-4e3e-a56a-  
20f7a66cafc1\", \"EntityLogicalName\": \"bookableresource\", \"InnerCalendarId\\  
\":\"f76cc333-cbbe-eb11-a81d-000d3a6e4359\"}"  
}
```

### Response

```
{  
  "InnerCalendarIds": "[\"f76cc333-cbbe-eb11-a81d-000d3a6e4359\"]"  
}
```

## Create a working hour daily recurrence

Starting May 20, 2021, Bob decides to work with Contoso all week from 8:00 AM to 5:00 PM, and will stop working there on July 15, 2021.

### Request

```
{  
  "CalendarEventInfo": "{\"CalendarId\": \"d33263c7-c16b-4e3e-a56a-  
20f7a66cafc1\", \"EntityLogicalName\": \"bookableresource\", \"TimeZoneCode\": 5  
, \"RecurrenceEndDate\": \"2021-07-15T00:00:00.000Z\", \"RulesAndRecurrences\":  
[{\\"Rules\": [{\"StartTime\": \"2021-05-20T08:00:00.000Z\", \"EndTime\": \"2021-  
05-  
20T17:00:00.000Z\", \"Effort\": 1, \"WorkHourType\": 0}], \"RecurrencePattern\": \\\n
```

```
"FREQ=WEEKLY;INTERVAL=1;BYDAY=SU,MO,TU,WE,TH,FR,SA\"}]}"  
}
```

## Response

```
{  
  "InnerCalendarIds": "[\"20f6cfa7-cfbe-eb11-a81d-000d3a6e4359\"]"  
}
```

## Edit a working hour daily recurrence with increased capacity

Bob decides to stop working for the entire week of June 15, 2021 to take a break. Until then, Bob will continue the all-week schedule as previously agreed to. Debbie makes these changes by using the `msdyn_SaveCalendar` API.

## Request

```
{  
  "CalendarEventInfo": "{\"CalendarId\": \"d33263c7-c16b-4e3e-a56a-  
  20f7a66cafc1\", \"EntityLogicalName\": \"bookableresource\", \"TimeZoneCode\": 5  
  , \"RecurrenceEndDate\": \"2021-06-15T00:00:00.000Z\", \"RulesAndRecurrences\":  
  [{\"Rules\": [{\"StartTime\": \"2021-05-20T08:00:00.000Z\", \"EndTime\": \"2021-  
  05-  
  20T17:00:00.000Z\", \"Effort\": 1, \"WorkHourType\": 0}], \"InnerCalendarId\": \"2  
  0f6cfa7-cfbe-eb11-a81d-  
  000d3a6e4359\", \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=SU,MO,TU,  
  ,WE,TH,FR,SA\"}]}"  
}
```

## Response

```
{  
  "InnerCalendarIds": "[\"867a2461-cdbe-eb11-a81d-000d3a6e4359\"]"  
}
```

## Create a working hour weekly recurrence

Starting from June 16, 2021, Bob will work from 8:00 AM to 5:00 PM on Wednesdays and Fridays, and will take a break from 12:00 PM to 12:30 PM for lunch. Debbie uses the `msdyn_SaveCalendar` API, but makes a mistake, and schedules the break from 12:00 PM to 1:00 PM.

## Request

```
{  
  "CalendarEventInfo": "{\"CalendarId\":\"d33263c7-c16b-4e3e-a56a-  
  20f7a66cafc1\", \"EntityLogicalName\":\"bookableresource\", \"TimeZoneCode\":5  
  , \"RulesAndRecurrences\":[{\"Rules\":[{\"StartTime\":\"2021-06-  
  16T08:00:00.000Z\", \"EndTime\":\"2021-06-  
  16T12:00:00.000Z\", \"Effort\":1, \"WorkHourType\":0}, {\"StartTime\":\"2021-  
  06-16T12:00:00.000Z\", \"EndTime\":\"2021-06-  
  16T13:00:00.000Z\", \"Effort\":null, \"WorkHourType\":1},  
  {\"StartTime\":\"2021-06-16T13:00:00.000Z\", \"EndTime\":\"2021-06-  
  16T17:00:00.000Z\", \"Effort\":1, \"WorkHourType\":0}], \"RecurrencePattern\":\\"  
  FREQ=WEEKLY;INTERVAL=1;BYDAY=WE,TH,FR\\\"}]}"
```

## Response

```
{  
  "InnerCalendarIds": "[\"1f894441-d0be-eb11-a81d-000d3a6e4359\"]"  
}
```

## Edit a break from a working hour weekly recurrence

Debbie then corrects the mistake and changes the break to occur from 12:00 PM to 12:30 PM by using the `msdyn_SaveCalendar` API.

## Request

```
{  
  "CalendarEventInfo": "{\"CalendarId\":\"d33263c7-c16b-4e3e-a56a-  
  20f7a66cafc1\", \"EntityLogicalName\":\"bookableresource\", \"IsEdit\":\\\"true\\\"  
  , \"TimeZoneCode\":5, \"RulesAndRecurrences\":[{\"Rules\":[  
  {\"StartTime\":\"2021-06-15T08:00:00.000Z\", \"EndTime\":\"2021-06-  
  15T12:00:00.000Z\", \"Effort\":1, \"WorkHourType\":0}, {\"StartTime\":\"2021-  
  06-15T12:00:00.000Z\", \"EndTime\":\"2021-06-  
  15T12:30:00.000Z\", \"Effort\":null, \"WorkHourType\":1},  
  {\"StartTime\":\"2021-06-15T12:30:00.000Z\", \"EndTime\":\"2021-06-  
  15T17:00:00.000Z\", \"Effort\":1, \"WorkHourType\":0}], \"RecurrencePattern\":\\"  
  FREQ=WEEKLY;INTERVAL=1;BYDAY=WE,TH,FR\\\"}]}"
```

```
15T17:00:00.000Z\", \"Effort\":1, \"WorkHourType\":0}], \"InnerCalendarId\": \"1f894441-d0be-eb11-a81d-000d3a6e4359\", \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=WE,TH,FR\"}]}"  
}
```

## Response

```
{  
  \"InnerCalendarIds\": [\"1f894441-d0be-eb11-a81d-000d3a6e4359\"]}  
}
```

## Create a working hour custom recurrence

Tim works for Contoso on Mondays from 8:00 AM to 5:00 PM, and Wednesdays from 11:00 AM to 3:00 PM. Tim started working for Contoso on May 16, 2021. Debbie uses the `msdyn_SaveCalendar` API to create Tim's work hours.

## Request

```
{  
  \"CalendarEventInfo\": {\"CalendarId\": \"a68245c9-ba2e-4496-9c18-3bee75fda396\", \"EntityLogicalName\": \"bookableresource\", \"TimeZoneCode\": 5, \"IsVaried\": true, \"RulesAndRecurrences\": [{\"Rules\": [{\"StartTime\": \"2021-05-16T08:00:00.000Z\", \"EndTime\": \"2021-05-16T17:00:00.000Z\", \"Effort\": 1, \"WorkHourType\": 0}], \"Action\": 1, \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=MO\"}, {\"Rules\": [{\"StartTime\": \"2021-05-16T11:00:00.000Z\", \"EndTime\": \"2021-05-16T15:00:00.000Z\", \"Effort\": 1, \"WorkHourType\": 0}], \"Action\": 1, \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=WE\"}]}}  
}
```

## Response

```
{  
  \"InnerCalendarIds\": [\"9fb8c199-d1be-eb11-a81d-000d3a6e4359\",  
  \"a2b8c199-d1be-eb11-a81d-000d3a6e4359\"]}  
}
```

## Edit a working hour custom recurrence

Tim's schedule then changes to work hours from Wednesdays 5:00 PM to 8:00 PM, and Thursday 10:00 AM to 12:00 PM. Monday is removed from Tim's schedule. Debbie uses the `msdyn_SaveCalendar` API to achieve this.

## Request

```
{  
  "CalendarEventInfo": "{\"CalendarId\": \"a68245c9-ba2e-4496-9c18-  
  3bee75fda396\", \"EntityLogicalName\": \"bookableresource\", \"TimeZoneCode\": 5  
  , \"IsVaried\": true, \"IsEdit\": true, \"RulesAndRecurrences\": [{\"Rules\":  
  [{\"StartTime\": \"2021-05-16T08:00:00.000Z\", \"EndTime\": \"2021-05-  
  16T17:00:00.000Z\", \"Effort\": 1, \"WorkHourType\": 0}], \"Action\": 2, \"InnerCal  
  endarId\": \"9fb8c199-d1be-eb11-a81d-  
  000d3a6e4359\", \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=MO\"},  
  {\"Rules\": [{\"StartTime\": \"2021-05-16T17:00:00.000Z\", \"EndTime\": \"2021-  
  05-  
  16T20:00:00.000Z\", \"Effort\": 1, \"WorkHourType\": 0}], \"Action\": 3, \"InnerCal  
  endarId\": \"a2b8c199-d1be-eb11-a81d-  
  000d3a6e4359\", \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=WE\"},  
  {\"Rules\": [{\"StartTime\": \"2021-05-16T10:00:00.000Z\", \"EndTime\": \"2021-  
  05-  
  16T12:00:00.000Z\", \"Effort\": 1, \"WorkHourType\": 0}], \"Action\": 1, \"InnerCal  
  endarId\": null, \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=TH\"}]}  
}
```

## Response

```
{  
  "InnerCalendarIds": ["a2b8c199-d1be-eb11-a81d-000d3a6e4359",  
  "942bda0f-d3be-eb11-a81d-000d3a6e4359"]  
}
```

## Edit a working hour occurrence in a recurrence

On May 26, 2021, Tim is only able to work from 1:00 PM to 7:00 PM. Debbie uses the `msdyn_SaveCalendar` API here.

## Request

```
{  
  "CalendarEventInfo": "{\"CalendarId\": \"a68245c9-ba2e-4496-9c18-  
  3bee75fda396\", \"EntityLogicalName\": \"bookableresource\", \"TimeZoneCode\": 5  
  , \"IsVaried\": true, \"IsEdit\": true, \"RulesAndRecurrences\": [{\"Rules\":  
  [{\"StartTime\": \"2021-05-26T13:00:00.000Z\", \"EndTime\": \"2021-05-26T19:00:00.000Z\",  
  \"Effort\": 1, \"WorkHourType\": 0}], \"Action\": 2, \"InnerCal  
  endarId\": \"9fb8c199-d1be-eb11-a81d-  
  000d3a6e4359\", \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=MO\"},  
  {\"Rules\": [{\"StartTime\": \"2021-05-26T17:00:00.000Z\", \"EndTime\": \"2021-05-26T19:00:00.000Z\",  
  \"Effort\": 1, \"WorkHourType\": 0}], \"Action\": 3, \"InnerCal  
  endarId\": \"a2b8c199-d1be-eb11-a81d-  
  000d3a6e4359\", \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=WE\"},  
  {\"Rules\": [{\"StartTime\": \"2021-05-26T10:00:00.000Z\", \"EndTime\": \"2021-05-26T12:00:00.000Z\",  
  \"Effort\": 1, \"WorkHourType\": 0}], \"Action\": 1, \"InnerCal  
  endarId\": null, \"RecurrencePattern\": \"FREQ=WEEKLY;INTERVAL=1;BYDAY=TH\"}]}  
}
```

```
, \"RulesAndRecurrences\":[{\\"Rules\": [{\"StartTime\":\"2021-05-26T13:00:00.000Z\", \"EndTime\":\"2021-05-26T19:00:00.000Z\", \"Effort\":1, \"WorkHourType\":0}], \"InnerCalendarId\":\"a2b8c199-d1be-eb11-a81d-000d3a6e4359\"}]}]
```

## Response

```
{  
  \"InnerCalendarIds\": [\"a2b8c199-d1be-eb11-a81d-000d3a6e4359\"]}  
}
```

## Delete a working hour custom recurrence

Tim has decided to leave the company and has to delete their entire schedule. Debbie uses the `msdyn_DeleteCalendar` API here.

### Request

```
{  
  \"CalendarEventInfo\": {\"CalendarId\":\"a68245c9-ba2e-4496-9c18-3bee75fda396\", \"EntityLogicalName\":\"bookableresource\", \"InnerCalendarId\":\"34d2210c-9fb6-eb11-a820-000d3afb1dba\", \"IsVaried\":true}  
}
```

### Response

```
{  
  \"InnerCalendarIds\": [\"a2b8c199-d1be-eb11-a81d-000d3a6e4359\", \"942bda0f-d3be-eb11-a81d-000d3a6e4359\"]}  
}
```

## Create time off

Tim will be taking three days off for a family vacation starting on June 9, 2021.

### Request