

To filter a SharePoint list in PowerApps to check if a room is already booked based on a selected room and a time range, you can use the SharePoint list's `Room Name` , `Start Time` , and `End Time` fields to identify overlapping reservations. Below is a step-by-step guide to implement this in PowerApps.

Scenario

- **SharePoint List:** A list named `Reservations` with columns:
 - `RoomName` (e.g., Single Line of Text or Choice)
 - `StartTime` (Date and Time)
 - `EndTime` (Date and Time)
- **Goal:** Check if a selected room is booked for a given time range (e.g., a new booking's start and end times).
- **PowerApps Setup:** Assume you have:
 - A dropdown or combo box (`DropdownRoom`) for selecting `RoomName` .
 - Date pickers or text inputs for the new booking's `StartTime` (`DatePickerStart`) and `EndTime` (`DatePickerEnd`).
 - A button to check availability or a gallery to display conflicts.

Logic for Checking Room Availability

A room is considered booked if the new booking's time range overlaps with any existing reservation. An overlap occurs when:

- The new booking's `StartTime` is before an existing booking's `EndTime` , AND
- The new booking's `EndTime` is after an existing booking's `StartTime` .

PowerApps Filter Formula

To filter the `Reservations` list for overlapping bookings for a selected room, use the `Filter` function in PowerApps. Here's an example formula:

```
Filter(  
    Reservations,  
    RoomName = DropdownRoom.Selected.Value, // Match the selected room  
    StartTime <= DatePickerEnd.SelectedDate + Time(Hour(DatePickerEnd.Selecte  
    EndTime >= DatePickerStart.SelectedDate + Time(Hour(DatePickerStart.Selec  
)
```

Explanation

- **Data Source:** `Reservations` is the SharePoint list.
- **Room Filter:** `RoomName = DropdownRoom.Selected.Value` ensures only bookings for the selected room are considered.
- **Time Overlap Check:**
 - `StartTime <= DatePickerEnd.SelectedDate + Time(...)` : Checks if the existing booking starts before the new booking ends.
 - `EndTime >= DatePickerStart.SelectedDate + Time(...)` : Checks if the existing booking ends after the new booking starts.
- **Time Handling:** The `Time` function ensures proper handling of time components from the date pickers or text inputs. Adjust `Hour` and `Minute` if your inputs are separate (e.g., `TextInputStartTime.Text`).

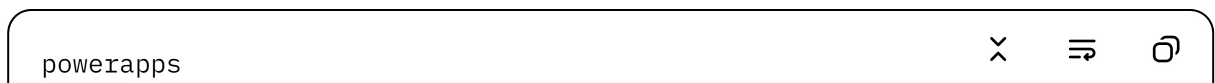
Example Implementation

1. Set Up Controls:

- **DropdownRoom:** Set `Items` to `Distinct(Reservations, RoomName)` to list available rooms.
- **DatePickerStart** and **DatePickerEnd:** Date pickers for selecting the start and end times of the new booking.
- **Gallery** (optional): To display conflicting bookings, set its `Items` property to the filter formula above.
- **Label** (optional): To show availability status (e.g., "Room is booked" or "Room is available").

2. Check Availability on Button Click:

- Add a button with the `OnSelect` property to check for conflicts and display a message:



```
If(
    IsEmpty(
        Filter(
            Reservations,
            RoomName = DropdownRoom.Selected.Value,
            StartTime <= DatePickerEnd.SelectedDate + Time(Hour(Date
                EndTime >= DatePickerStart.SelectedDate + Time(Hour(Date
                    )
                ),
            ),
            Notify("Room is available", NotificationType.Success),
            Notify("Room is already booked", NotificationType.Error)
        )
    )
```

3. Display Conflicts in a Gallery:




- Add a gallery and set its `Items` property to the filter formula above.
- Add labels in the gallery to show details like `ThisItem.RoomName`, `ThisItem.StartTime`, and `ThisItem.EndTime`.

Example Output

- If `DropDownRoom` is set to "Conference Room A", `DatePickerStart` is set to "10/14/2025 10:00 AM", and `DatePickerEnd` is set to "10/14/2025 11:00 AM", the filter will return any bookings for "Conference Room A" that overlap with 10:00 AM to 11:00 AM.
- If the gallery shows no items or `IsEmpty` returns `true`, the room is available.

Notes

- **Time Zone Considerations:** Ensure `StartTime` and `EndTime` in the SharePoint list and PowerApps are in the same time zone. SharePoint stores dates in UTC, so you may need to adjust using `DateAdd` or `TimeZoneOffset` if your app uses local time.
- **Input Validation:** Add checks to ensure `DatePickerEnd` is after `DatePickerStart`:

powerapps  Collapse  Wrap  Copy

```
If(
    DatePickerEnd.SelectedDate + Time(Hour(DatePickerEnd.SelectedDate), M
    DatePickerStart.SelectedDate + Time(Hour(DatePickerStart.SelectedDate
    Notify("End time must be after start time", NotificationType.Error)
)
```

- **Performance:** For large lists, consider delegating the filter. Ensure `RoomName` is indexed in SharePoint if possible, though time-based filters are not delegable in PowerApps.
- **Alternative Inputs:** If using text inputs for time (e.g., `TextInputStartTime`), parse them using `DateTimeValue` or `TimeValue`:

powerapps  Collapse  Wrap  Copy

```
DateTimeValue(DatePickerStart.SelectedDate & " " & TextInputStartTime.Text
```

If you need help with specific control configurations, handling time zones, or integrating with a booking form, let me know!

↳ [Handle time zones in filters](#)

↳ [Send booking confirmation email](#)