

Airtable README - Move Orthopedics Timesheets - v5

aka Corali

Intention of Airtable base is to give accounts manager overview of how to invoice/collect employees' salary from each of the doctors (Hardcastle, Wever, Groenewald) as result of proportion of time shared employee has spent at their practice.

Additionally, (in amendment to v1), takes into consideration if employee has received a rebate amount from hospital in which case respective amounts due by each practice is reduced.

⚡ for each variation of [Location](#) and [Type of Work](#) a new form must be submitted

therefore these fields set to "Single Select" type fields (previously "Multiple Select")

✍ Idea for Future Version (and more generic)

refer to refer to v3 - grouped

- extract hours worked from start and stop time (*split fields to Start Time and End Time - "Duration" type fields*)
 - determine overtime based thereon
- combine worksheets of different individuals working for same organization and use unique identifier (Name or ID Number) to split entries
 - could make [Recon](#) sheet for each OR have group by this identifier in [Recon](#) sheet
 - can then also determine overtime on individual basis ie. X works 6h and Y 10h

⌚ Requires Standardization

[Location](#)

[Type of Work](#)

How Base Works

Tables

Data

Date

(identifier)

Week Number

Type: Formula

```
WEEKNUM(Date)
```

Location

Type: Single Select

- Hardcastle
- Wever
- Groenewald
- Practise
divided equally
- SICK
- LEAVE

Type of Work

Type: Multiple Select

- Meeting
- Theatre Preparation
- ...

Location Notes

Type: Long Text

Start-Stop Time

Type: Single Line Text

Admin Hours

Type: Duration

Format: duration (h:mm)

Theatre Hours

Type: Duration

Format: duration (h:mm)

Overtime

Type: Duration

Format: duration (h:mm)

Total Hours

Type: Formula

Description: calculates the sum of regular and overtime hours for each record.

```
Hours + Overtime
```

Format: duration (h:mm)

Queries

Type: Long Text

MRH Rebate Amount Due

Type: Formula

```
{Theatre Hours} / 3600 * 134.17
```

NOTICE: assumes fixed hourly rate

MHR Payment Received

Type: Checkbox

NOTICE: used for [Recon > Balance Due](#)

Calendar Month

Type: Formula

Description: displays the full month name (e.g., January, February) from the Date field

based on field [Date](#) entered in [Timesheet Form](#)

```
IF(DATETIME_FORMAT({Date Created}, 'MMMM') = {Billing Month},  
    'No',  
    'Yes'  
)
```

Billing Month

refer to v1 for more detail

once satisfied with base - delete this field

Link to Recon

Type: Link to Other Records

NOTICE: do not allow linking to multiple records in [Recon](#) to prevent counting duplicated work

Date Created

Type: Built-In

NOTICE: enables [Is Manual Intervention Required?](#) and can be used to verify if individual claims to have completed the form in timely manner

Is Manual Intervention Required?

Type: Formula

Description: indicates if manual intervention is needed by comparing the record's creation month to the Billing Month.

if time entry logged/created is after automation trigger, will need to link data to "Recon" sheet using "MANUALLY FIX RECON" view

NB - link based on Billing Month (or Calendar Month if this option chosen)

```
IF(
```

```
    AND(
```

```
        DATETIME_FORMAT({Date}, 'MMMM') = {Billing Month},
```

```
        {Link to Recon} = BLANK()
```

```
),
```

```
    'Yes',
```

```
    'No'
```

)

Recon

Trigger Date Time

(identifier)

Previous Month aka Billing Month

Type: Formula

Description: used in automation [for future Recon updates](#) to group all records from previous month in [Data sheet](#) into same [Recon](#) sheet record

```
DATETIME_FORMAT(
```

```
DATEADD({Trigger Date Time}, -1, 'month'),  
'MMMM'
```

)

The screenshot shows a Zoho Automation trigger configuration. The trigger is set to "At a scheduled time" (Every 1 month on the 5th and last day at 11:59 PM). The actions section includes a "Create record" step in the "In Recon" base, followed by a "Find records" step where the condition is "Where Billing Month is Previous Month aka Billing Month". This leads to a "Repeat for each in" loop, which contains an "Update a record" step. The "ACTION DETAILS" panel on the right shows the "Action type" as "Find records", "LABELS" as "Description", and "CONFIGURATION" settings for "Action will run..." (Always), "Table" (Data), and "Find records based on" (Condition). The condition is set to "Where Billing Month is Previous Month aka Billing Month". The "TEST STEP" panel indicates a test step to confirm the trigger's configuration.

Year

Type: Formula

```
YEAR({Trigger Date Time})
```

used to generate a view

The screenshot shows a Power BI Data view titled "2025 Recon Sheet". The view has a filter bar at the top with "Filtered by Year" set to 2025. The main area contains a table with the following columns: "A Trigger Date Time", "Month", "Start Time", "End Time", "Total Hours", and "Hardcastle". The table data is as follows:

A Trigger Date Time	Month	Start Time	End Time	Total Hours	Hardcastle
2025-07-27T09:00:00.058Z	March	07:30	09:00	1:30	Groenewald
2025-07-29T09:00:00.098Z	January	08:00	09:00	1:00	17:00
2025-07-29T09:00:00.133Z	April	08:00	09:00	1:00	32:00
2025-07-29T09:00:00.137Z	July	08:00	09:00	1:00	14:00
2025-07-29T09:00:00.168Z	August	08:00	09:00	1:00	48:00
2025-07-29T09:00:00.228Z	September	08:00	09:00	1:00	16:30
2025-07-31T21:30:00.041Z					156:20
2025-08-21T21:45:00.035Z					55:00
2025-09-21T21:45:38.938Z					133:00
					38:00
					10:00
					39:30
					33:00
					121:00
					19:00
					57:00

Hardcastle | Total Admin Hours

Type: Rollup

using [Admin Hours](#)

with condition based on [Location](#)

Formula: sum(values)

Format: duration (h:mm)

Hardcastle | Total Theatre Hours

Type: Rollup

using [Theatre Hours](#)

with condition based on [Location](#)

Formula: sum(values)

Format: duration (h:mm)

Wever | Total Admin Hours

Type: Rollup

using [Admin Hours](#)

with condition based on [Location](#)

Formula: sum(values)

Format: duration (h:mm)

Wever | Total Theatre Hours

Type: Rollup

using [Theatre Hours](#)

with condition based on [Location](#)

Formula: sum(values)

Format: duration (h:mm)

Groenewald | Total Admin Hours

Type: Rollup

using [Admin Hours](#)

with condition based on [Location](#)

Formula: sum(values)

Format: duration (h:mm)

Groenewald | Total Theatre Hours

Type: Rollup

using [Theatre Hours](#)

with condition based on [Location](#)

Formula: sum(values)

Format: duration (h:mm)

Practise | Total Admin Hours

Type: Rollup

using [Admin Hours](#)

with condition based on [Location](#)

Formula: sum(values)

Format: duration (h:mm)

Practise | Total Theatre Hours

Type: Rollup

using [Theatre Hours](#)

with condition based on [Location](#)

Formula: sum(values)

Format: duration (h:mm)

Total Hours

Type: Formula

```
{Hardcastle | Total Hours} + {Wever | Total Hours} + {Groenewald | Total
```

Hours}

Format: duration (h:mm)

Practise | Total Hours

Type: Formula

```
{Practise | Total Admin Hours} + {Practise | Total Theatre Hours}
```

Format: duration (h:mm)

Hardcastle | Total Hours

Type: Formula

```
{Hardcastle | Total Theatre Hours} + {Hardcastle | Total Admin Hours}  
+ (1/3)*{Practise | Total Hours}
```

Format: duration (h:mm)

Wever | Total Hours

Type: Formula

```
{Wever | Total Theatre Hours} + {Wever | Total Admin Hours} + (1/3)*  
{Practise | Total Hours}
```

Format: duration (h:mm)

Groenewald | Total Hours

Type: Formula

```
{Groenewald | Total Theatre Hours} + {Groenewald | Total Admin Hours}  
+ (1/3)*{Practise | Total Hours}
```

Format: duration (h:mm)

Hardcastle | Admin vs Theatre

Description: calculates the ratio of total theatre hours to total admin hours; allowing for insight or review.

```
IF({Hardcastle | Total Admin Hours} = 0, IF({Hardcastle |  
Total Theatre Hours} = 0, "NO INFORMATION", "N/A"), ROUND(({Hardcastle |  
Total Admin Hours} / {Hardcastle |  
Total Hours}) * 100) & "% vs " & ROUND(({Hardcastle |  
Total Theatre Hours} / {Hardcastle | Total Hours}) * 100) & "%")
```

Wever | Admin vs Theatre

Description: calculates the ratio of total theatre hours to total admin hours; allowing for insight or review

```
IF(  
  
{Wever | Total Admin Hours} = 0,  
  
IF(  
  
{Wever | Total Theatre Hours} = 0,  
  
"NO INFORMATION",  
  
"N/A"  
  
,  
  
ROUND(({Wever | Total Admin Hours} / {Wever |  
Total Hours}) * 100) & "% vs " & ROUND(({Wever |  
Total Theatre Hours} / {Wever | Total Hours}) * 100) & "%"  
  
)
```

Groenewald | Admin vs Theatre

Description: calculates the ratio of total theatre hours to total admin hours; allowing for insight or review

```
IF({Groenewald | Total Admin Hours} = 0,  
  
IF({Groenewald | Total Theatre Hours} = 0,  
  
BLANK(),  
  
"0% vs 100%"
```

```
)  
  
IF({Groenewald |  
Total Theatre Hours} = 0, "100% vs 0%", ROUND(({Groenewald |  
Total Admin Hours} / {Groenewald |  
Total Hours}) * 100) & "% vs " & ROUND(({Groenewald |  
Total Theatre Hours} / {Groenewald | Total Hours}) * 100) & "%")  
  
)
```

Hardcastle | Proportion

Type: Formula

```
{Hardcastle | Total Hours} / {Total Hours}
```

Format: duration (h:mm)

Never | Proportion

Type: Formula

```
{Never | Total Hours} / {Total Hours}
```

Format: duration (h:mm)

Groenewald | Proportion

Type: Formula

```
{Groenewald | Total Hours} / {Total Hours}
```

Format: duration (h:mm)

Total Due

Type: Formula

```
{Total Hours} /3600 * 134.17
```

FORMAT: currency (R)

MRH Rebate Amount Due

ROLLUP

using [MRH Rebate Amount Due](#)

with condition [MHR Payment Received](#) "yes"

FORMULA: `sum(values)`

FORMAT: currency (R)

Balance Due

Type: Formula

```
{Total Due} - {MRH Rebate Amount Due}
```

Due by Hardcastle

Type: Formula

```
{Hardcastle | Proportion} * {Balance Due}
```

Due by Wever

Type: Formula

```
{Wever | Proportion} * {Balance Due}
```

Due by Groenewald

Type: Formula

```
{Groenewald | Proportion} * {Balance Due}
```

Forms

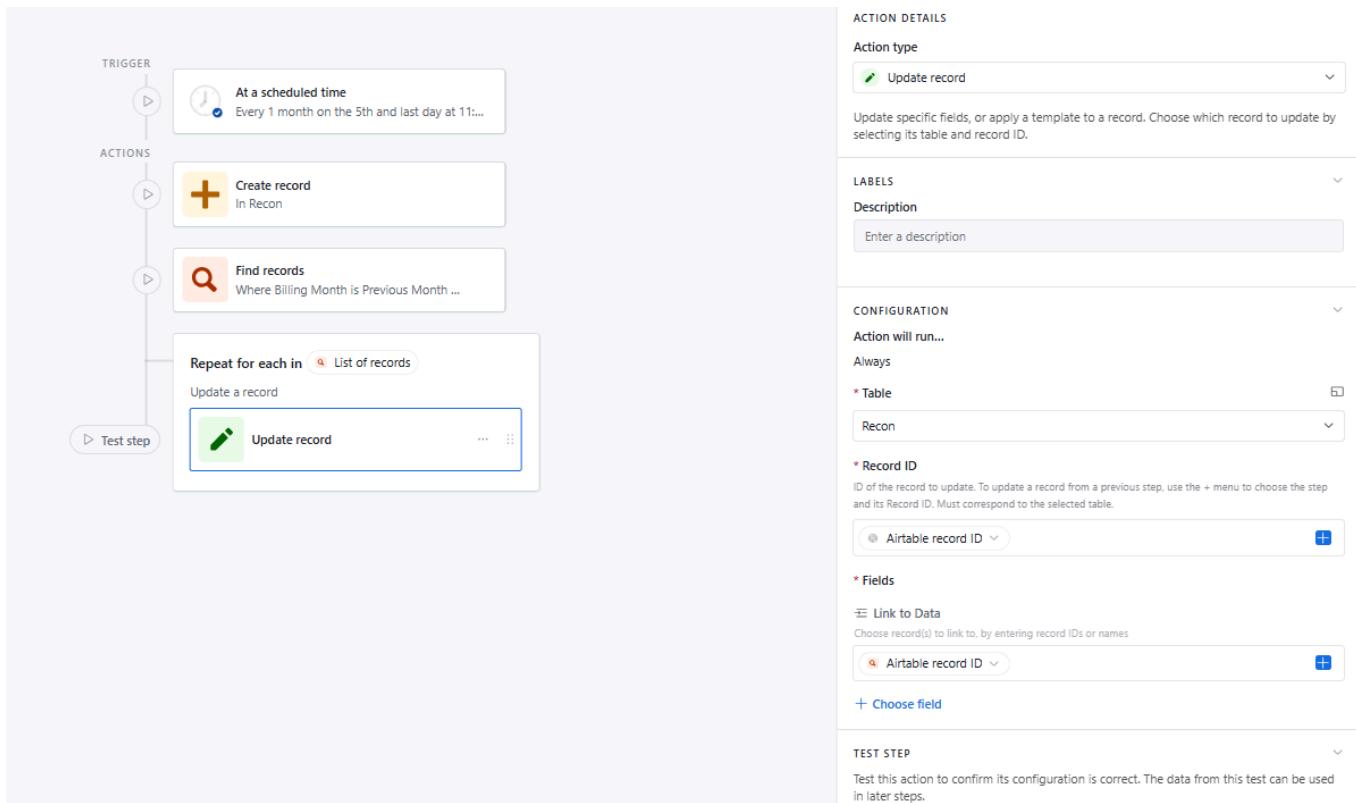
Timesheet Form

uses [Date](#), [Location](#), [Location notes](#), [Start-Stop Time](#), [Hours](#), [Overtime](#), [Type of Work](#)

Automations

for future Recon updates

eg. November 2025 - end



for back-dated Recon updates

 January 2025 

At a scheduled time, create a record, and 2 more ...

 February 2025 

At a scheduled time, create a record, and 2 more ...

 March 2025 

At a scheduled time, create a record, and 2 more ...

 May 2025 

At a scheduled time, create a record, and 2 more ...

 April 2025 

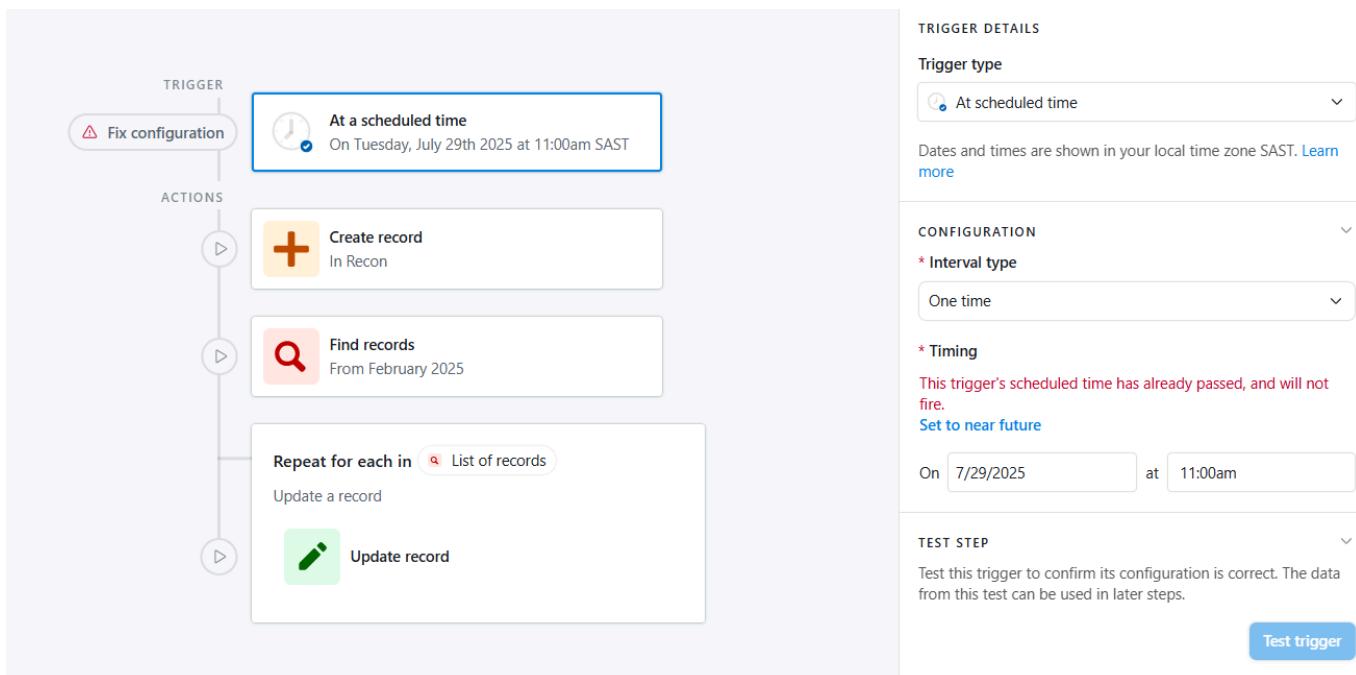
At a scheduled time, create a record, and 2 more ...

 June 2025 

At a scheduled time, create a record, and 2 more ...

 July 2025 

At a scheduled time, create a record, and 2 more ...



set trigger at specific time

use personal views in [Data](#) as filter for records to be linked to given record in [Recon](#)

The screenshot shows a software interface with a main list of records and a detailed view of one record.

Main List:

- + Create new...
- Billing Periods - required for back-dating Recon table
 - January 2025
 - February 2025
 - March 2025** (selected)
 - April 2025
 - May 2025
 - June 2025
 - July 2025
 - August 2025
 - September 2025

Detailed View (Record 15):

Date	Location Notes	Hours
March 7, 2025		
March 10, 2025		
March 11, 2025		
March 12, 2025		
March 12, 2025		08H00 - 16H00
March 12, 2025		08H00 - 17H00
March 18, 2025		08H00 - 17H00
March 19, 2025		08H00 - 12H00
March 19, 2025		12H00 - 14H00
March 19, 2025		15H00 - 17H00
March 20, 2025		08H00 - 17H00
March 21, 2025		13H00 - 15H00
March 24, 2025		08H00 - 17H00

Filtering Dialog:

In this view, show records

Where

- Year = 2025
- and Billing Month Is March

Add condition + Add condition group ⚡

Cancel Apply

End of Document

Last updated: November 06, 2025