

SUPERCARGE

Resource Management with the power of APIs



smartsheet

ENGAGE

v.07/16

Legal

Certain information set forth in this presentation may be “forward-looking information.” Except for statements of historical fact, information contained herein may constitute forward-looking statements. Forward-looking statements are not guarantees of future performance and undue reliance should not be placed on them.

Such forward-looking statements necessarily involve known and unknown risks and uncertainties, many of which are and will be described in Smartsheet filings with the US Securities and Exchange Commission, and these risks and uncertainties may cause actual performance and financial results in future periods to differ materially from any projections of future performance or results expressed or implied by such forward-looking statements. Although forward-looking statements contained herein are based upon what Smartsheet management believes are reasonable assumptions, there can be no assurance that forward-looking statements will prove to be accurate, as actual results and future events could differ materially from those anticipated in such statements. Smartsheet undertakes no obligation to update forward-looking statements except as required by law.

This presentation is proprietary to Smartsheet and the content herein is confidential and intended for permitted internal use only. This content shall not be disclosed to any third party that is not under an obligation of confidentiality to Smartsheet.

Smartsheet is a registered trademark of Smartsheet Inc. The names and logos of actual companies and products used in this presentation are the trademarks of their respective owners and no endorsement or affiliation is implied by their use.

Session Speakers



David Murray
Technical Solutions Engineer



Matt Nulton
Technical Solutions Engineer



Session Agenda

1

Introduction

2

APIs

3

Advanced
Automation

1

Introduction

Who this talk is for, Advanced Analytics Reports, what we're covering

 smartsheet
ENGAGE



What you know

Resource Management core features

Technical expertise NOT required

What you'll take away

1. Knowledge of RM Advanced Reports
2. How to evaluate external systems for connecting with Resource Management
3. Better understanding of APIs and how they can be used with Resource Management
4. Use case ideas and inspiration

Quick Sidebar on Resource Management

RM at a glance

New! Advanced Reporting capability



Resource Management

Brief Overview

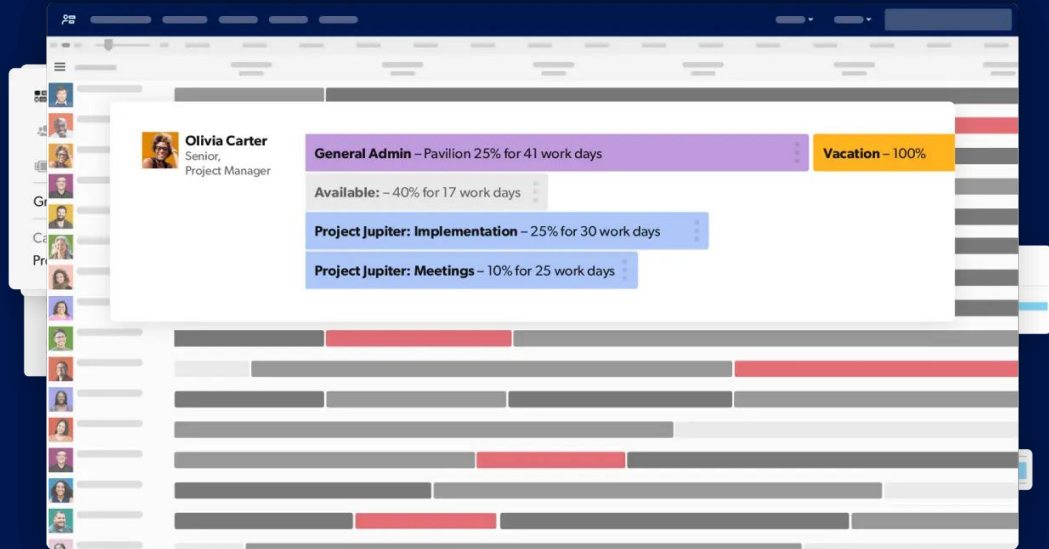
Improve productivity by centralizing your people

Balance workloads to achieve sustained high performance

Boost project results by always building the right team

Leverage time tracking to improve project estimates

Forecast hiring needs for more informed decision-making



New Advanced Reporting

Actionable Insights from RM Data

- Create Smartsheet assets for typically used reports and visualization
- Automate ingestion of critical RM data into Smartsheet assets
- Enhance Smartsheet solutions with RM insights
- Refresh your reporting data automatically or on-demand

Create advanced report ✕

Send your data to a source sheet and Smartsheet Report. You can update the report and adjust the settings from the Analytics page.

Name

Time & Fees: Hours by Project

Destination

RM Projects ✕ ▼

Data refresh settings ⓘ

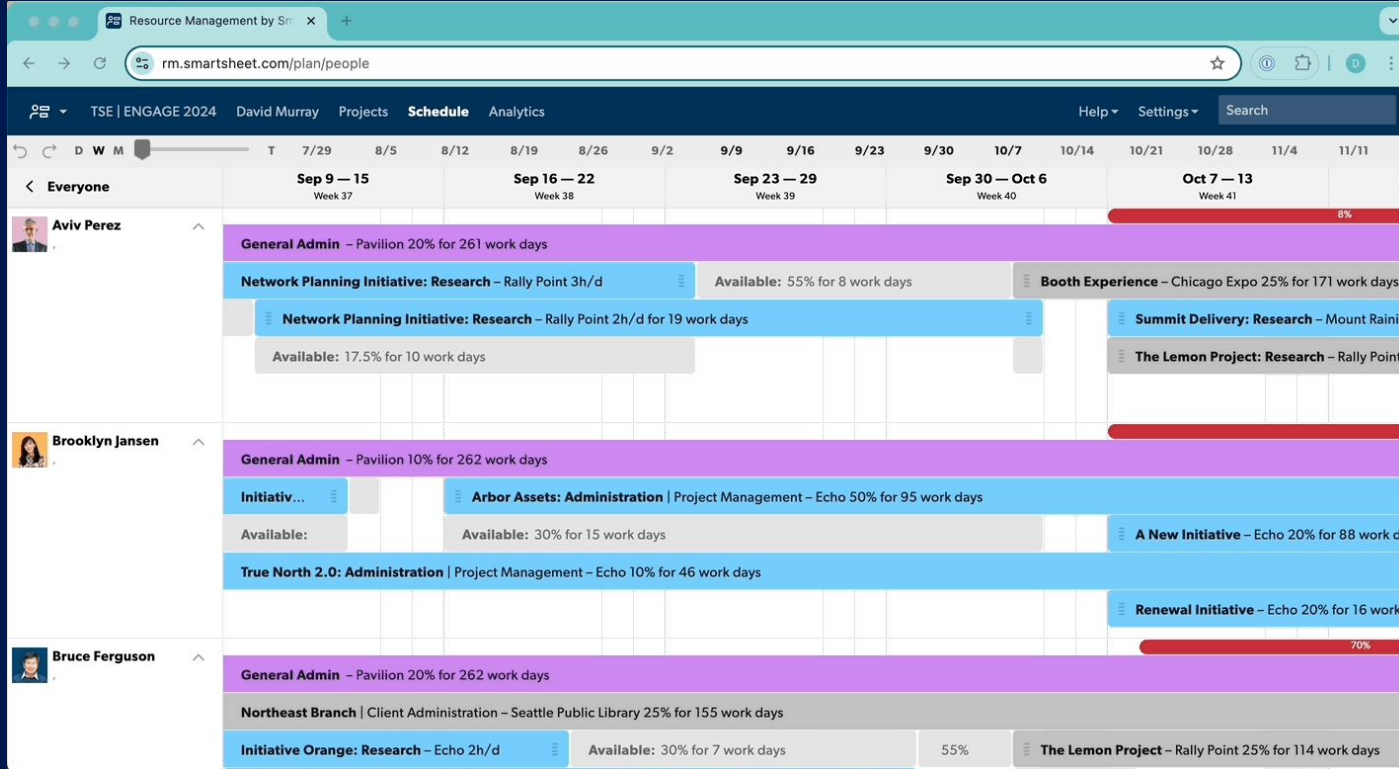
Automatic
Advanced report data will refresh every 24 hours

Manual
Data must be manually refreshed from the Analytics page

Cancel Create

Time & Fees Category	Hours	Rate	Total
Administrative	4.00	4.00	0.00
Client Work	8.00	0.00	0.00

Create Advanced Reports



View Advanced Reports

Resource Management by Smartsheet

rm.smartsheet.com/analytics

TSE | ENGAGE 2024 David Murray Projects Schedule Analytics Help Settings Search

TSE | ENGAGE 2024

New Report

Who's available?

This week, remaining

- David Murray
- Matt Nulton
- mbf admin
- Nathan Lloyd

Active Projects Tracked in Currency

Project Budget Incurred Future Scheduled

Project Name	Project Budget	Incurred	Future Scheduled
Echo A New Initiative	High	Medium	Low
Echo Arbor Assets	High	High	Medium
Chicago Expo Booth Experience	High	High	Medium
Mount Rainier Education Film Campaign	Low	Low	Low
Pavilion Estimate for App Design	High	Medium	Low

Show 22 more projects

Forecasted Budget

Project Name	Variance	Percentage
Echo A New Initiative	\$75,159.20 over budget	or 41%
Echo Arbor Assets	\$153,300 over budget	or 45%
Chicago Expo Booth Experience	\$179,376 over budget	or 60%
Mount Rainier Education Film Campaign	\$18,492 remaining	or 26%
Pavilion Estimate for App Design	\$57,316 over budget	or 29%

Active Projects Tracked in Days

No active projects to display

Saved Reports

Search

What we'll cover next

How to evaluate APIs

Use cases and inspiration
with a couple demos



2

Unlocking Potential

Harnessing APIs to Elevate your Smartsheet Solutions

 smartsheet
ENGAGE



What you know

Data from external systems can make Smartsheet solutions more impactful

Technical expertise NOT required

What you'll take away

1. What is an API
2. How do you use an API
3. How do you determine if an API can be utilized to solve for your use case

Our API Journey

1

Why should we care about APIs?

2

What is an API, really though?

3

Are there different types of APIs?

4

API Request & Response

5

How to Evaluate an API

6

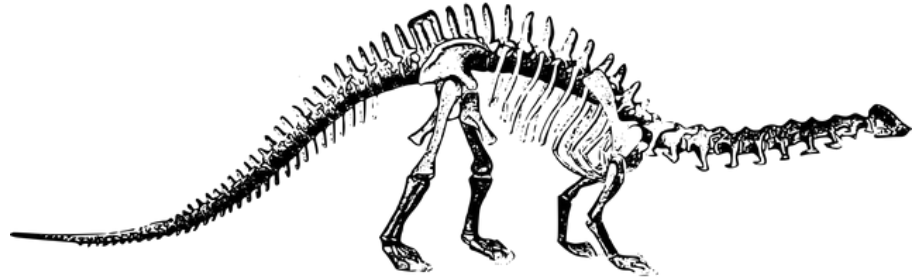
API Documentation Review

7

Sample API Calls

8

API Orchestration
(Putting it all together)



APIs are the
backbone
of modern digital
ecosystems

Why APIs Matter



Integrate

Break down silos



Automate

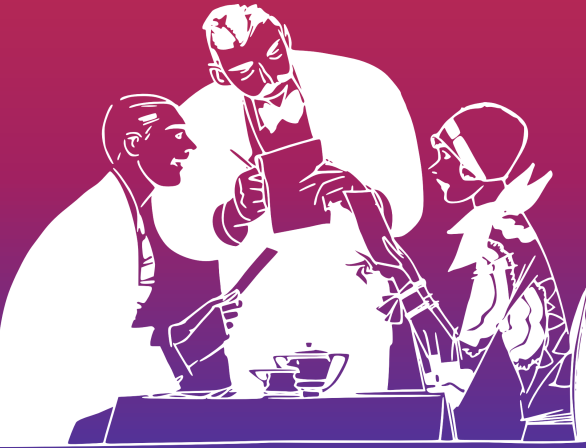
Streamline workflows



Innovate

Support organization growth





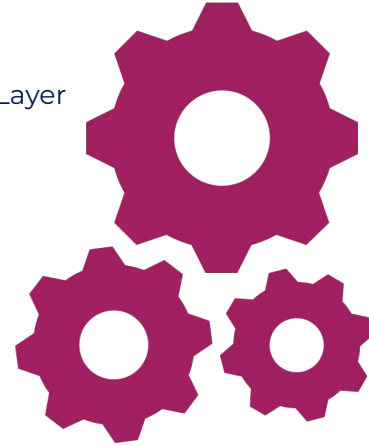
What is an API

Application Programming Interface

APIs enable software to communicate and work together

Custom Integration Layer
(leveraging APIs)

Software A



Software B

Types of APIs



REST

Simple & Scalable

Flexible & widely supported

Various improvements to increase speed and efficiency

Outputs JSON, typically, a more simple and small format

Implementation may vary greatly between platforms



SOAP

Formerly the most widely used type, usage has declined

More rigid, provides an additional layer of security

Generally speaking, higher overhead, slower performance

Outputs only XML data (more complex, larger size)



GraphQL

Released in 2015

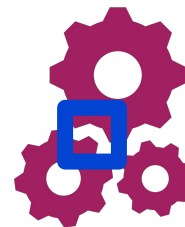
Extremely efficient at fetching specific pieces of data from massive repositories

Detailed Query Language

High degree of complexity

REST API Connection Lifecycle

Request & Response



Interface Client
(Diner)

API
(Waiter)



Request

Response

Destination Service

URL (before the /)

Destination Endpoint

URL (after the /)

Request Details

Method (GET, PUT, POST, DELETE, etc.)

Query Parameters

Headers (Authentication, Content Type, etc.)

Request Body

Typically used only for PUT/POST

These methods **update/add** data.

Response Timing

Synchronous vs. Asynchronous

Response Status

Successful vs. Unsuccessful

200, 201, 401, 403, 404 etc.

Response Body

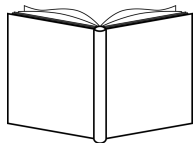
Matches format requested by HEADERS

Returns data requested or created, updated etc.

Examples typically provided in documentation

Evaluating an API

Important Considerations



Documentation

Clear and **detailed** guides are vital to get started quickly.

Insufficient documentation or overly complex, disorganized resources that obscure essential information are both significant **red flags**.

Published **SDKs** aide in development of **API Orchestration**



Security

APIs should protect data with appropriate security measures

Validate types of AUTH available (**None, Basic, API Token, OAuth** etc.)

Ensure that organization supports use of available methods



Use Case Specifics

Do the correct **Methods & Endpoints** exist to achieve your desired outcome? Does the the **response** of those calls provide all needed data?

Does your desired frequency, and needed calls have any potential to exceed **rate limits**?

Tools & Resources



Postman

Quickly test API Endpoints

Verify functionality, observe data returned, compare to expected return

Obtain data to test against (data transformation)



Community Forums

Embrace the power of community!

Smartsheet Community

GitHub

Stack Exchange

Example

Documentation

OpenWeather API

Call current weather data

How to make an API call

API call

```
https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}
```

Parameters

<code>lat</code>	required	Latitude. If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
<code>lon</code>	required	Longitude. If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
<code>appid</code>	required	Your unique API key (you can always find it on your account page under the "API key" tab)
<code>mode</code>	optional	Response format. Possible values are <code>xml</code> and <code>html</code> . If you don't use the <code>mode</code> parameter format is JSON by default. Learn more
<code>units</code>	optional	Units of measurement. <code>standard</code> , <code>metric</code> and <code>imperial</code> units are available. If you do not use the <code>units</code> parameter, <code>standard</code> units will be applied by default. Learn more
<code>lang</code>	optional	You can use this parameter to get the output in your language. Learn more

ENGAGE 2024 / Get Current Weather

Service Endpoint

GET Send

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> lat	27.77		
<input checked="" type="checkbox"/> lon	-82.63		
<input checked="" type="checkbox"/> appid	95963db6d3752097786cf7ab81e60834		
<input checked="" type="checkbox"/> units	imperial		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 273 ms Size: 865 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "coord": {
3     "lon": -82.63,
4     "lat": 27.77
5   },
6   "weather": [
7     {
8       "id": 802,
9       "main": "Clouds",
10      "description": "scattered clouds",
11      "icon": "03d"
12    }
13  ],
14  "base": "stations",
15  "main": {
16    "temp": 86.14,
17    "feels_like": 98.74,
18    "temp_min": 84.02,
19    "temp_max": 88.11,
20    "pressure": 1016,
21    "humidity": 79,
22    "sea_level": 1016,
23    "grnd_level": 1016
24  },
25  "visibility": 10000,
26  "wind": {
27    "speed": 6.91,
28    "deg": 140
29  },
30 }
```

Example

Postman Call

Example

Documentation

OpenWeather API

Call current weather data

How to make an API call

API call

```
https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}
```

Parameters

<code>lat</code>	required	Latitude. If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
<code>lon</code>	required	Longitude. If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
<code>appid</code>	required	Your unique API key (you can always find it on your account page under the "API key" tab)
<code>mode</code>	optional	Response format. Possible values are <code>xml</code> and <code>html</code> . If you don't use the <code>mode</code> parameter format is JSON by default. Learn more
<code>units</code>	optional	Units of measurement. <code>standard</code> , <code>metric</code> and <code>imperial</code> units are available. If you do not use the <code>units</code> parameter, <code>standard</code> units will be applied by default. Learn more
<code>lang</code>	optional	You can use this parameter to get the output in your language. Learn more

Smartsheet API

High Level Documentation Review

<https://smartsheet.redoc.ly/>

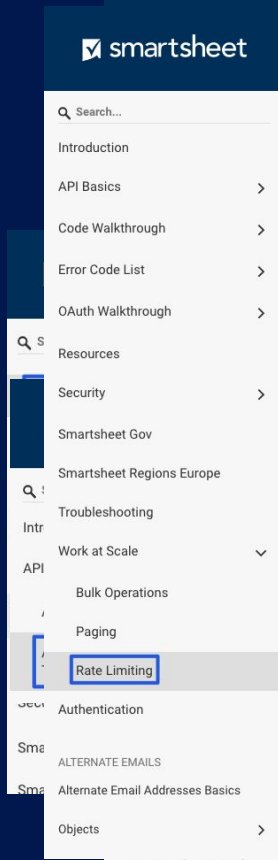
Authentication

HTTP (Headers, Methods, Status Codes)

SDK Availability

Security

Rate Limiting



The image shows a navigation menu for the Smartsheet API documentation. The menu is organized into several sections: Introduction, API Basics, Code Walkthrough, Error Code List, OAuth Walkthrough, Resources, Security, Smartsheet Gov, Smartsheet Regions Europe, Troubleshooting, Work at Scale, API, Bulk Operations, Paging, Rate Limiting (highlighted with a blue box), Authentication, ALTERNATE EMAILS, Alternate Email Addresses Basics, and Objects.

- smartsheet
- Search...
- Introduction
- API Basics >
- Code Walkthrough >
- Error Code List >
- OAuth Walkthrough >
- Resources
- Security >
- Smartsheet Gov
- Smartsheet Regions Europe
- Troubleshooting
- Work at Scale >
- API
 - Bulk Operations
 - Paging
 - Rate Limiting**
- Authentication
- ALTERNATE EMAILS
- Alternate Email Addresses Basics
- Objects >

Rate Limiting

Handle "Rate limit exceeded" Error

To prevent abuse and undue stress on the Smartsheet servers, Smartsheet reserves the right to enforce some limits depending on the load on our systems. This reduction is sometimes called rate limiting or throttling. Certain operations, such as attaching a file and getting cell history, are resource intensive.

The Smartsheet API implements "rate limiting" to protect the system. When API calls exceed an acceptable load, an HTTP **429** status will be returned along with the following response body:

```
{
  "errorCode": 4003,
  "message": "Rate limit exceeded."
}
```

Smartsheet recommends that you design your integration to gracefully handle this rate limit error. One way of doing that would be to have your integration *sleep* for a minimum of 60 seconds when this error is encountered, and then subsequently retry the request.

Alternatively, you might choose to implement [exponential backoff](#) (an error handling strategy whereby you periodically retry a failed request with progressively longer wait times between retries, until either the request succeeds or the certain number of retry attempts is reached). Note that the SDKs implement this behavior.

Avoid Executing "Rapid Fire" Updates

If the only thing your integration does is execute an [Update Rows](#) request once every second for the same sheet, that would only amount to a total of 60 requests per minute – well within rate limiting guidelines. However, updating the same object in such rapid succession could result in save errors that negatively impact both your integration as well as user experience within the Smartsheet app. To avoid this scenario, design your integration such that API requests are never executed with rapid-fire succession against the same Smartsheet object. For maximum efficiency, consider batching up changes and submitting them in a single request using a [bulk operation](#) (for example, [Update Rows](#) or [Add Columns](#)).

Execute Requests Serially

Executing multiple API requests in parallel to update a specific Smartsheet object results in reduced performance and often results in errors due to save collisions. To avoid this scenario, design your integration such that API requests to update a specific Smartsheet object are always executed serially (that is, execute one request at a time, not beginning the next request until the previous request has completed).

NOTE: Attempts to perform multiple concurrent updates to a sheet may result in [error code 4004](#).

Use the Smartsheet SDKs

[The SDKs provide default backoff and retry to accommodate rate limiting responses from the API.](#) Note that the default maximum retry duration is typically 30 seconds. You may wish to increase this if your application is making many API calls in quick succession. For specific instructions per language, see the [Readme](#) for the respective SDK.

Smartsheet API

GET Sheet Method

<https://smartsheet.redoc.ly/>

Authorization/Path Params/Include

Filters

Header Params

Documentation Example Calls

Response Samples

smartsheet

Get Sheet

Gets a sheet in the format specified, based on the sheet Id.

AUTHORIZATIONS: (APIToken) OR (OAuth2 (READ_SHEETS))

PATH PARAMETERS

sheetId	number
<i>required</i>	Sheet Id of the sheet being accessed.

QUERY PARAMETERS

accessApiLevel	number
Default: 0	Allows COMMENTER access for inputs and return values. For backwards-compatibility, VIEWER is the default. For example, to see whether a user has COMMENTER access for a sheet, use accessApiLevel=1.

Response samples

200 default

Content type: application/json

```
{  "version": 0}
```

Copy Expand all Collapse all

Enum: "attachments" "columnType" "crossSheetReferences" "discussions" "filters" "filterDefinitions" "format" "ganttConfig" "objectValue" "ownerInfo" "rowPermalink" "source" "writerInfo"

Example: rowsModifiedSince=2020-01-30T13:23:32-07:00

ENGAGE 2024 / Get Sheet

Service & Version Endpoint

GET https://api.smartsheet.com/2.0/sheets/5543646149431172 Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Headers Hide auto-generated headers

Key	Value	Description	Bulk Edit	Presets
Authorization			

Body Cookies Headers (17) Test Results 200 OK 649 ms 1.88 KB Save Response

Pretty Raw Preview Visualize JSON

```
{
  "id": 5543646149431172,
  "name": "Engage 2024 Get Sheet Example",
  "version": 1,
  "totalRowCount": 10,
  "accessLevel": "OWNER",
  "effectiveAttachmentOptions": [
    "EVERNOTE",
    "ONEDRIVE",
    "DROPBOX",
    "EGNYTE",
    "LINK",
    "GOOGLE_DRIVE",
    "FILE",
    "BOX_COM"
  ],
  "gantEnabled": false,
  "dependenciesEnabled": false,
  "resourceManagementEnabled": false,
  "resourceManagementType": "NONE",
  "cellImageUploadEnabled": true,
  "userSettings": {
    "criticalPathEnabled": false,
    "displaySummaryTasks": true
  },
  "userPermissions": {
    "summaryPermissions": "ADMIN"
  },
  "hasSummaryFields": false,
  "permalink": "https://app.smartsheet.com/sheets/92MMVcpgrv66h2hrrhVPMzxXc4iHFQ0GRmqcvH1",
  "createdAt": "2024-09-12T19:13:37Z",
  "modifiedAt": "2024-09-12T19:15:10Z",
  "isMultiPicklistEnabled": true,
  "columns": [
```

Get Sheet

Postman Call

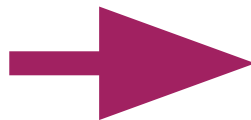
Get Sheet

Response Comparison

Engage 2024 Get Sheet Example ☆



Make	Model	Year	VIN
Toyota	Xtra	1993	WBABD33404J246587
GMC	Yukon	2005	JTJBM7FX2C5035685
Mitsubishi	Chariot	1986	19XFB2E50FE115167
Land Rover	Discovery	2002	KNDKG3A2XA7881613
Volkswagen	rio	2000	5UXZV8C50D0555792
Mazda	RX-7	1985	WAUML44E64N549312
Mazda	CX-9	2011	2G4GS5EV9D9881010
GMC	Yukon Denali	2006	1N4AB7APXEN368480
Mazda	Miata MX-5	2002	1G4HE5EM7AU174313
Audi	4000s	1985	1C4NJPBA0ED994974



```
"rows": [
  {
    "id": 6504226019053444,
    "rowNumber": 1,
    "expanded": true,
    "createdAt": "2024-09-12T19:15:10Z",
    "modifiedAt": "2024-09-12T19:15:10Z",
    "cells": [
      {
        "columnId": 3511176737804164,
        "value": "Toyota",
        "displayValue": "Toyota"
      },
      {
        "columnId": 8014776365174660,
        "value": "Xtra",
        "displayValue": "Xtra"
      },
      {
        "columnId": 696426970697604,
        "value": 1993.0,
        "displayValue": "1993"
      },
      {
        "columnId": 5200026598068100,
        "value": "WBABD33404J246587",
        "displayValue": "WBABD33404J246587"
      }
    ]
  }
]
```

RM API

High Level Documentation Review

<https://10kft.github.io/10kft-api/>

Authentication

Collections & Objects

Optional Fields & Date/Time Formatting

Webhooks

Rate Limiting

The screenshot displays the 'Optional Fields' section of the SmartSheet API documentation. On the left is a navigation sidebar with a tree structure. The main content area includes a text box explaining optional fields, a terminal window with a curl command and its output, a JSON response snippet, and a text box explaining how to request multiple optional fields. A 'Date & Time Formatting' section is partially visible at the bottom.

RESOURCE MANAGEMENT BY SMARTSHEET API

- Resource Management by Sm...
- Overview
- Need help?
- Endpoints
- Webhook Support
- Zapier Integration
- Third party libraries
- Questions?

SECTIONS

- Before starting to use the Re...
- Being notified about importa...
- Collections & Objects
- Pagination
- Filtering
- Authentication**
 - Optional Fields
 - Date & Time Formatting
 - Error Handling
 - Throttling & Rate-Limiting
- Getting help & reporting pro...
- API Token Management
- Approvals
- Assignables
- Assignments
- Create Assignment with Subt...
- Subtask Counts
- Bill Rates
- Budget Item Categories
- General API Use Examples
- Project Budgets
- Custom Fields and Custom Fi...
- Disciplines

Optional Fields

To reduce the number of roundtrips that might be required to fetch all related data for a given collection or resource (e.g. fetching a user and all their tags), the API supports a concept of optional fields. These `fields` are a comma separated list of field names that are supported as a URL parameter when making a request to fetch a resource or a resource collection.

```
curl -X GET https://api.rm.smartsheet.com/api/v1/users?fields=tags \
-H "Content-Type: application/json"
-H "auth: TOKEN"
```

Each field requested is included in the response as nested collections. The page size for these nested collections will be the same as the page size used for the parent API request.

```
{
  "data": [
    {
      "id": 1,
      "first_name": "Tom",
      "last_name": "Perera",
      "tags": {
        "data": [
          { "id": 1, "value": "developer" },
          { "id": 2, "value": "javascript" },
        ],
        "paging": {
          "page": 1,
          "total": 2
        }
      },
      "paging": {
        "page": 1,
        "total": 1
      }
    }
  ]
}
```

Multiple optional fields can be requested in a single API call like `fields=f1,f2`.

Date & Time Formatting

The API handles or expects dates and date-time values in various scenarios, and expects the values to be in specific formats.

- Date** values provided as query parameters when making API calls must be in the `2013-01-31` format.
- Time** values are accepted in the `2013-09-31T22:10:24Z` format, which is a date and time value in UTC time.

RM API

PUT Project Method

<https://10kft.github.io/10kft-api/>

Update a Project

Response Samples

RESOURCE MANAGEMENT BY SMARTSH

RESOURCE MANAGEMENT BY SMARTSHEET API

- Project Budgets
- Custom Fields and Custom Fi...
- Disciplines
- Expense Items
- Holidays
- Leave Types
- Phases
- Placeholder Resources
- Tags per Project
- Project Membership
- Users per Project
- Projects
 - Project State
 - List Projects (Index)
 - List projects
 - List projects with archived
 - List projects with sorting
 - Filter projects
 - Show a project
 - Show a project with its phases
 - Create a project
 - Update a Project**
 - Locking Time Entries
 - Delete a Project
 - Notes:
 - Sample Response
- Reports API

Update a Project

Note: You cannot add thumbnails to the project through the API

```
PUT /api/v1/projects/1245
```

```
{  "id": 1245,  "name": "New Project Name",  "ends_at": "2015-05-31"}
```

Locking Time Entries

`timeentry_lockout` is used to control when new time entries are no longer accepted to a project.

Value	Description
-1	Not locked
0	Locked
7	Locked for entries more than 7 calendar days ago. This can be any positive integer value

Note that positive integer values in this attribute are numbers of days relative to today's date on the calendar. We do not support setting this lockout value to a specific date.

If your project has phases, note that this must also be set for each phase `assignable_id` for it to take effect as intended.

Delete a Project

```
DELETE /api/v1/projects/1245
```

ENGAGE 2024 / Update Project Service & Version Endpoint

PUT <https://api.rm.smartsheet.com/api/v1/projects/9713578> Send

Params Authorization Headers (10) **Body** Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** Beautify

```
1 {
2   "bill_rates": [
3     {
4       "id": 93192990,
5       "rate": 125.0,
6       "assignable_id": 9713578
7     },
8     {
9       "id": 93192992,
10      "rate": 125.0,
11      "assignable_id": 9713578,
12      "user_id": 3277558
13     },
14     {
15      "id": 93192993,
16      "rate": 125.0,
17      "assignable_id": 9713578,
18      "user_id": 3277561
19     },
20     {
21      "id": 93387194,
22      "rate": 125.0,
23      "assignable_id": 9713578,
24      "user_id": 3277536
25     },
26     {
27      "id": 93444140,
28      "rate": 125.0
29   }
30 }
```

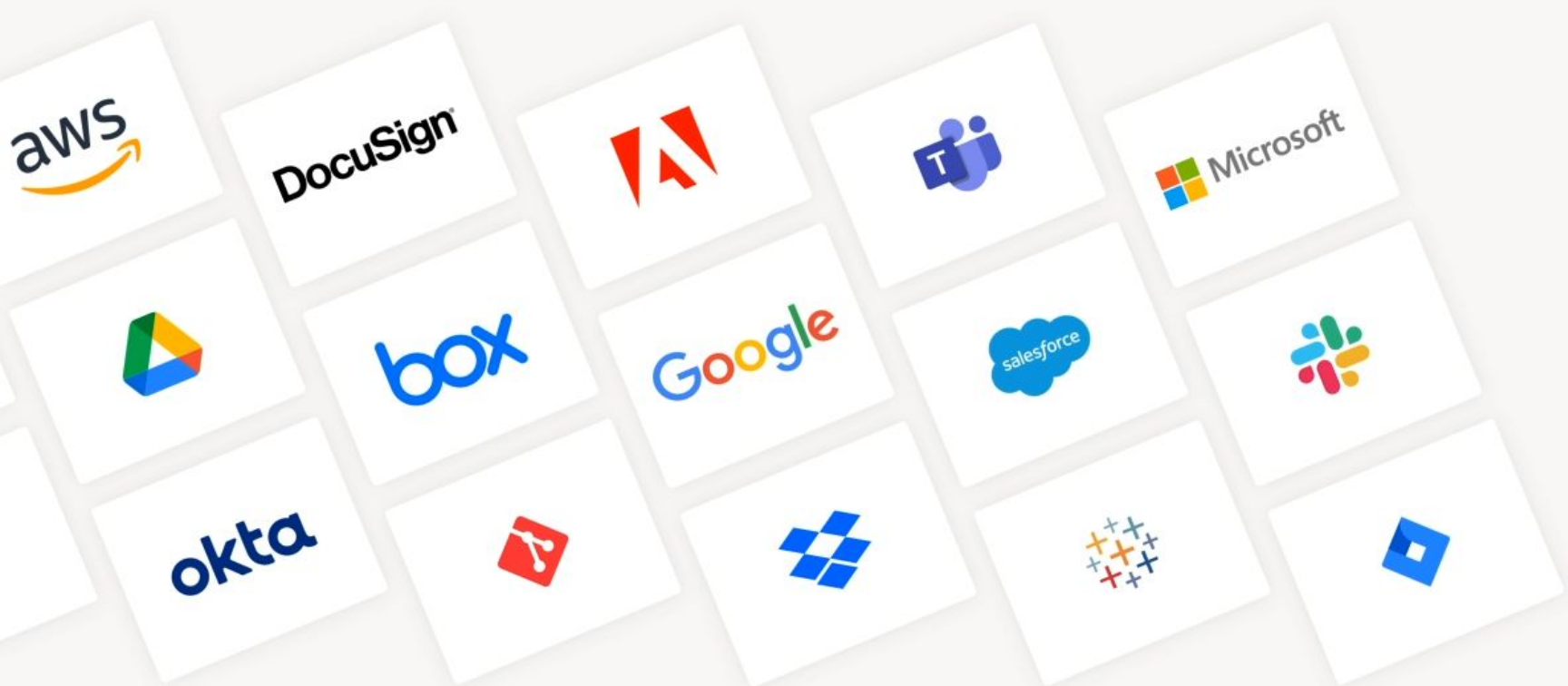
Body Cookies (2) Headers (23) Test Results **200 OK** 1075 ms 1.69 KB Save Response

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "id": 9713578,
3   "archived": false,
4   "archived_at": null,
5   "description": "",
6   "guid": "a71d38ed-a8c6-4ca6-a27d-70ba08ee2453",
7   "name": "Arbor Assets",
8   "parent_id": null,
9   "project_code": "0428",
10  "secureurl": "https://assets.rm.smartsheet.com/images/941116c8-e2da-46b3-a699-20e9ce08df8b.jpg",
```

Update Project

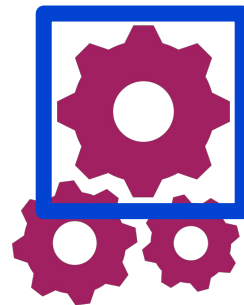
Postman Call



API Orchestration

Connects multiple tools or services to work together automatically.

API Orchestration



Automate Workflows

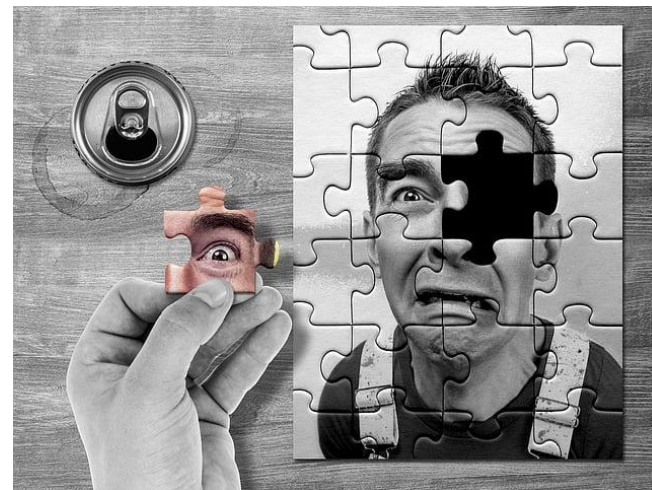
Save time by letting systems communicate with each other automatically.

Efficient Data Sharing

Move information between tools without manual effort.

... but how?

Listen for new information from external sources, or periodically fetch current data from external sources and **DO** something with it.



Orchestration Tools



iPaaS

Limited to prebuilt connections/workflows

Limited ability to transform data/handle attachments

Many workflows can be configured and maintained by non-technical users



Self-Hosted Custom Code

Purpose-built to handle unique use case

Requires hosting environment to run as built

Requires technical skills, programming knowledge, and solution architecture

3

Advanced Automations

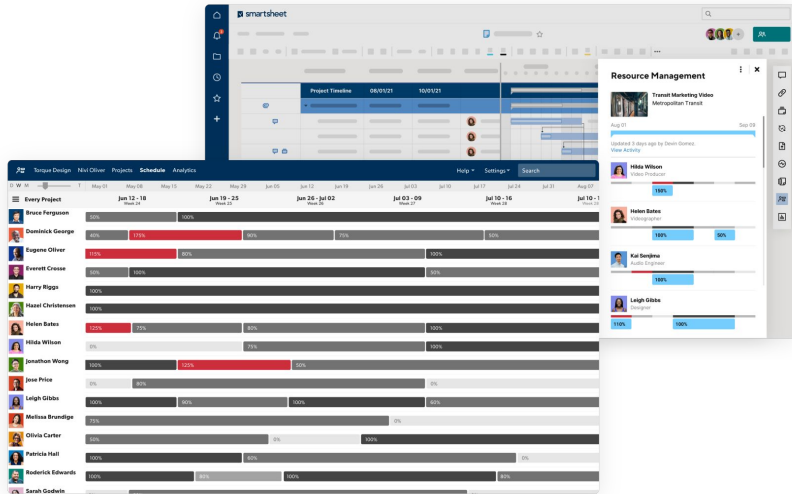
 smartsheet

ENGAGE

Advanced Automation Use Cases



SUPERCHARGED Examples



User Management

Updating People & Project Custom Fields

Managing PTO, Mat/Pat Leave

Integrate RM with an ERP

Managing Promotions at Scale

Dynamic Bill Rates deep dive

Automated Time Tracking deep dive

User Management

The Job

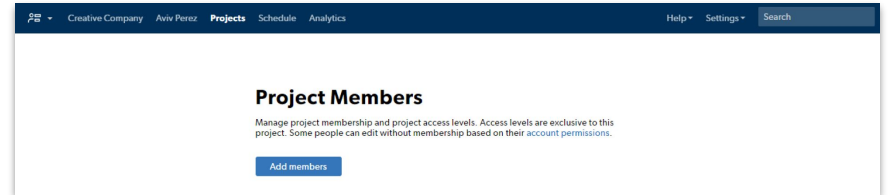
The User List in RM must be up to date so that people can be assigned and log time against assignments.

The Challenge

We manage all our people in our **HR system** and we don't want to duplicate efforts.

The Solution

Make changes in your HRS and automatically push the data into RM using API Orchestration.



Updating People Custom Fields

The Job

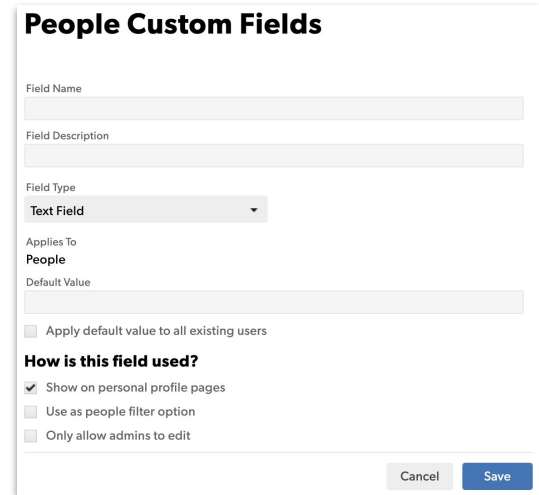
RM can make selecting the right resource for an assignment easier by utilizing tags to indicate a resource's skill set or location.

The Challenge

This type of data is often maintained in an **HR system** and we want to ensure data integrity between the two systems.

The Solution

API Orchestration between the HRS and RM allow for changes made in the HRS to automatically be reflected in RM.



The screenshot shows a configuration form titled "People Custom Fields". It contains the following fields and options:

- Field Name:** A text input field.
- Field Description:** A text input field.
- Field Type:** A dropdown menu currently set to "Text Field".
- Applies To:** A dropdown menu currently set to "People".
- Default Value:** A text input field.
- Apply default value to all existing users
- How is this field used?**
 - Show on personal profile pages
 - Use as people filter option
 - Only allow admins to edit

At the bottom right, there are "Cancel" and "Save" buttons.

Updating Project Custom Fields

The Job

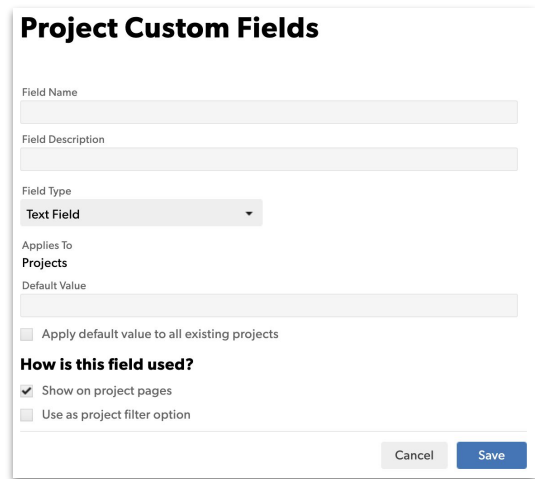
When **Smartsheet Control Center** is used in conjunction with RM, project metadata must be updated in the SCC environment.

The Challenge

SCC project metadata is only pushed to RM *at provisioning*; future updates are made manually.

The Solution

Create API Orchestration that leverages the Smartsheet and RM APIs to update RM project metadata fields when changes in Smartsheet are made.



The screenshot shows a form titled "Project Custom Fields" with the following fields and options:

- Field Name: [Text input field]
- Field Description: [Text input field]
- Field Type: [Dropdown menu with "Text Field" selected]
- Applies To: [Text input field containing "Projects"]
- Default Value: [Text input field]
- Apply default value to all existing projects
- How is this field used?**
 - Show on project pages
 - Use as project filter option

At the bottom right, there are "Cancel" and "Save" buttons.

Integrate RM with an ERP

The Job

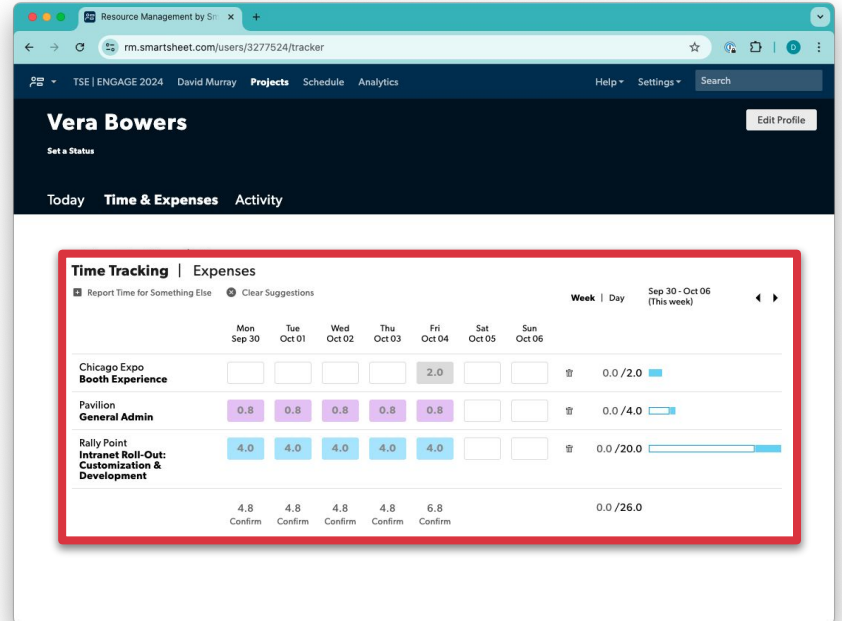
You use an **external ERP** time tracking solution for invoicing; all time tracking data is in that external system.

The Challenge

RM can only show planned costs if RM time tracking is not used. The full picture is not shown.

The Solution

API Orchestration between the two systems allows for actuals to automatically flow into RM.



Reflecting Team Member PTO/Leave in RM

The Job

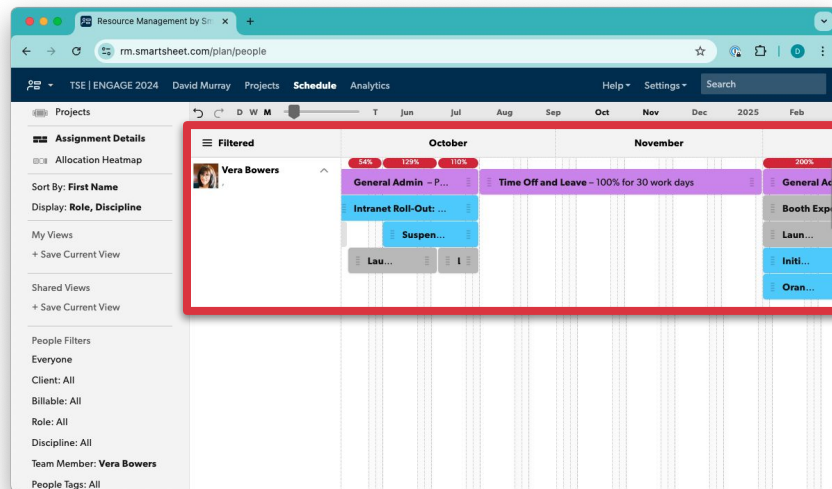
Team members need time off! These requests typically live in the **HR system**.

The Challenge

It can be cumbersome to replicate this data in RM, without it assignments and forecasting are not accounting for the time away.

The Solution

Configure a People Leave RM Project, and utilize API orchestration to automatically create leave assignments.



Managing Promotions at Scale

The Job

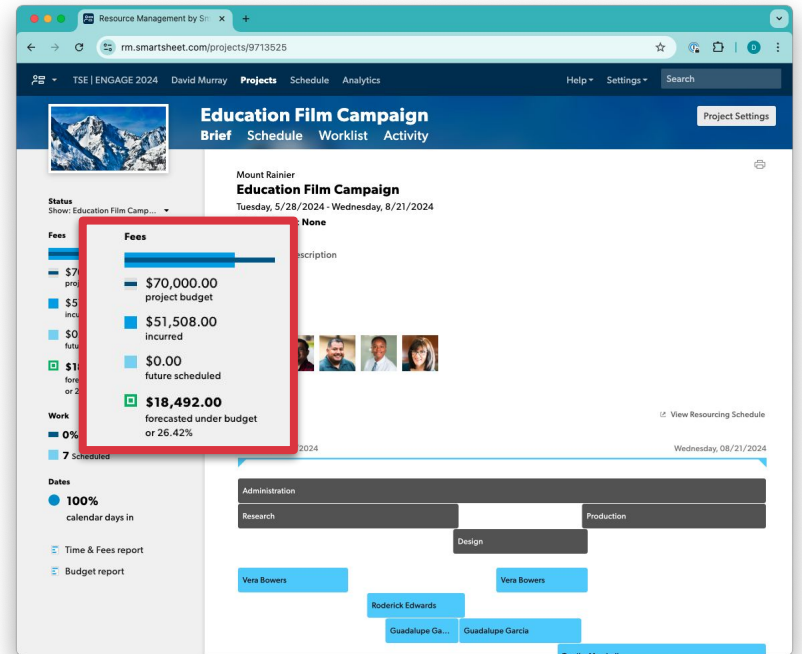
A team member is promoted with a new bill rate, role etc. but will maintain work on current project(s).

The Challenge

If care is not taken, the new bill rate/role will be reflected in the resources previous assignments.

The Solution

Automate the complex **RM** process of creating a new profile, splitting assignments between the two profiles, and archiving the original profile.



Advanced Automation Use Case

Dynamic Bill Rates in Resource Management



Resource Management Bill Rates

Brief RM Capabilities Overview

- Use of bill rates in RM enable things like:
 - Tracking cost of work vs. profitability
 - Budget forecasting
 - Accurate billing of customers
- RM enables bill rates be applied at these levels:
 - **Globally:** Role, Discipline, or User
 - **Per Project:** Role, Discipline, User, or Phase
- Accurate bill rates are critical for achieving business goals!



Dynamic Bill Rates

The Problem

- Bill rates can change over time, and often change while a project is in flight
- Options for modifying bill rates for in flight projects can be time consuming
- Potential for inefficient use of time adjusting bill rates manually
- Introduce risks such as inaccurate forecasting or incorrect billing of customers



Dynamic Bill Rates

The Solution

- Leverage Smartsheet and Resource Management APIs to update bill rates dynamically
- Take advantage of RM API capabilities to establish bill rates based on effective date ranges
- Provide ability to have more granularity when managing bill rates
- Improve accuracy for budget tracking



Dynamic Bill Rates

Example Walkthrough

- Bill rates are established with a default value based on role/discipline
- Bill rates cannot have effective start/end rate established
- Adjusting rates across projects for a given user can be time consuming

Bill Rates

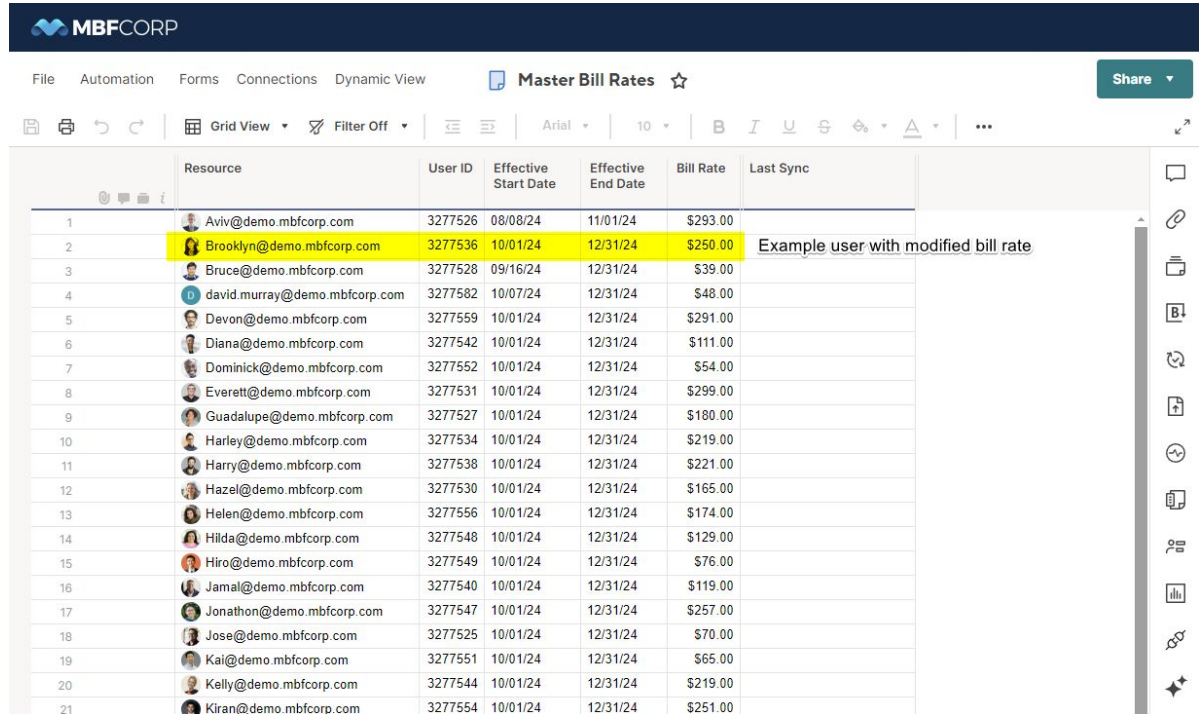
Bill rates for people currently on the project. These rates are derived from the current default bill rates that are specified below or set within a user profile. Rates in **Red** deviate from the default.

Brooklyn Jansen	\$125
Design	\$125
Harry Riggs	\$125
Paul Finley	\$125
Project Management	\$125
Shirley Huson	\$125
Tamika Marshall	\$200

Dynamic Bill Rates

Example Walkthrough

- Establish a Master Bill Rates sheet in Smartsheet
- Enter bill rates for each resource including the effective dates for a given bill rate



The screenshot shows a Smartsheet interface for a sheet titled "Master Bill Rates". The sheet contains a table with the following columns: Resource, User ID, Effective Start Date, Effective End Date, Bill Rate, and Last Sync. Row 2 is highlighted in yellow, and a note "Example user with modified bill rate" is placed in the Last Sync column for that row.

	Resource	User ID	Effective Start Date	Effective End Date	Bill Rate	Last Sync
1	Aviv@demo.mbfcorp.com	3277526	08/08/24	11/01/24	\$293.00	
2	Brooklyn@demo.mbfcorp.com	3277536	10/01/24	12/31/24	\$250.00	Example user with modified bill rate
3	Bruce@demo.mbfcorp.com	3277528	09/16/24	12/31/24	\$39.00	
4	david.murray@demo.mbfcorp.com	3277582	10/07/24	12/31/24	\$48.00	
5	Devon@demo.mbfcorp.com	3277559	10/01/24	12/31/24	\$291.00	
6	Diana@demo.mbfcorp.com	3277542	10/01/24	12/31/24	\$111.00	
7	Dominick@demo.mbfcorp.com	3277552	10/01/24	12/31/24	\$54.00	
8	Everett@demo.mbfcorp.com	3277531	10/01/24	12/31/24	\$299.00	
9	Guadalupe@demo.mbfcorp.com	3277527	10/01/24	12/31/24	\$180.00	
10	Harley@demo.mbfcorp.com	3277534	10/01/24	12/31/24	\$219.00	
11	Harry@demo.mbfcorp.com	3277538	10/01/24	12/31/24	\$221.00	
12	Hazel@demo.mbfcorp.com	3277530	10/01/24	12/31/24	\$165.00	
13	Helen@demo.mbfcorp.com	3277556	10/01/24	12/31/24	\$174.00	
14	Hilda@demo.mbfcorp.com	3277548	10/01/24	12/31/24	\$129.00	
15	Hiro@demo.mbfcorp.com	3277549	10/01/24	12/31/24	\$76.00	
16	Jamal@demo.mbfcorp.com	3277540	10/01/24	12/31/24	\$119.00	
17	Jonathon@demo.mbfcorp.com	3277547	10/01/24	12/31/24	\$257.00	
18	Jose@demo.mbfcorp.com	3277525	10/01/24	12/31/24	\$70.00	
19	Kai@demo.mbfcorp.com	3277551	10/01/24	12/31/24	\$65.00	
20	Kelly@demo.mbfcorp.com	3277544	10/01/24	12/31/24	\$219.00	
21	Kiran@demo.mbfcorp.com	3277554	10/01/24	12/31/24	\$251.00	

Request Method GET Request URI https://api.smartsheet.com/2.0/sheets/966765454512004/rows/7881988040527748

Request Method Headers (12) Body Pre-request Script Tests Settings

Body Cookies Headers (17) Test Results Status: 200 OK Time: 402 ms Size: 983 B

Pretty Raw Preview Visualize JSON

```

{
  "id": 7881988040527748,
  "sheetId": 966765454512004,
  "rowNumber": 2,
  "siblingId": 8829886279252356,
  "version": 69,
  "expanded": true,
  "accessLevel": "OWNER",
  "createdAt": "2024-09-06T14:39:29Z",
  "modifiedAt": "2024-09-12T17:28:53Z",
  "cells": [
    {
      "columnId": 2254798671007620
    },
    {
      "columnId": 6758398298378116,
      "value": "brooklyn@demo.mbfcorp.com",
      "displayValue": "Brooklyn@demo.mbfcorp.com"
    },
    {
      "columnId": 1128898764164996,
      "value": 3277536.0,
      "displayValue": "3277536"
    },
    {
      "columnId": 5632498391535492,
      "value": "2024-10-01"
    },
    {
      "columnId": 3380698577860244,
      "value": "2024-12-31"
    },
    {
      "columnId": 7884298205220740,
      "value": 250.0,
      "displayValue": "$250.00"
    },
    {
      "columnId": 2689888930713476,
      "value": "2024-09-12"
    },
    {
      "columnId": 4941688744398724,
      "value": "2024-09-12T13:20:50-04:00",
      "displayValue": "2024-09-12T13:20:50-04:00"
    },
    {
      "columnId": 565948810743684
    },
    {
      "columnId": 5869540430114180
    }
  ]
}

```

Request/Response

Dynamic Bill Rates

GET Row Postman Example

GET https://api.rm.smartsheet.com/api/v1/assignments?per_page=500&from=2024-10-21&to=2024-12-31 Request URI

Request Method

Query Params

Key	Value
<input checked="" type="checkbox"/> per_page	500
<input checked="" type="checkbox"/> from	2024-10-21
<input checked="" type="checkbox"/> to	2024-12-31

Request Parameters

Body Cookies (2) Headers (24) Test Results Status: 200 OK Time: 536 ms Size: 85.92 KB

Pretty Raw Preview Visualize JSON

Dynamic Bill Rates

GET Assignments in Effective Range

```

1  {
2    "paging": {
3      "per_page": 500,
4      "page": 1,
5      "previous": null,
6      "self": "/api/v1/assignments?from=2024-07-31&per_page=500&to=2024-12-31&page=1",
7      "next": null
8    },
9    "data": [
10     {
11       "id": 64697422,
12       "allocation_mode": "percent",
13       "user_id": 3277527,
14       "assignable_id": 9713523,
15       "ends_at": "2024-08-23",
16       "starts_at": "2024-06-28",
17       "repetition_id": null,
18       "created_at": "2024-09-06T21:06:12Z",
19       "updated_at": "2024-09-06T21:06:13Z",
20       "all_day_assignment": true,
21       "resource_request_id": null,
22       "percent": 1.0,
23       "bill_rate": 125.0,
24       "bill_rate_id": 93192866,
25       "status": null,
26       "description": "",
27       "status_option_id": null,
28       "note": ""
29     },
30     {
31       "id": 64697425,
32       "allocation_mode": "hours_per_day",
33       "user_id": 3277529,
34       "assignable_id": 9713525,
35       "ends_at": "2024-08-21",
36       "starts_at": "2024-07-19",
37       "repetition_id": null,
38       "created_at": "2024-09-06T21:06:13Z",
39       "updated_at": "2024-09-06T15:06:23Z",
40       "all_day_assignment": true,
41       "resource_request_id": null,
42       "hours_per_day": 2.0,
43       "bill_rate": 20.0,
44       "bill_rate_id": 93192870,
45       "status": null,
46       "description": null,
47       "status_option_id": null,
48       "note": null
49     },
50     {
51       "id": 64697426,
52       "allocation_mode": "hours_per_day",
53       "user_id": 3277525,
54       "assignable_id": 9713525,
55       "ends_at": "2024-08-21",

```

Request Response

Notice, we filter by date range!

Request Method GET https://api.rm.smartsheet.com/api/v1/projects/9713578/fields=bill_rates Request URI

Query Params

Key	Value
fields	bill_rates

Request Parameters

Body

```
1 {
2   "id": 9713578,
3   "archived": false,
4   "archived_at": null,
5   "description": "",
6   "guid": "a71d38ed-a8c6-4ca6-a27d-78ba08ee2453",
7   "name": "Aabor Assets",
8   "parent_id": null,
9   "project_code": "0428",
10  "secureurl": "https://assets.rm.smartsheet.com/images/941116c8-e2da-46b3-a699-28e9ce88df8b.jpg",
11  "secureurl_expiration": null,
12  "settings": 0,
13  "timeentry_lockout": -1,
14  "ends_at": "2025-03-12",
15  "starts_at": "2024-09-16",
16  "deleted_at": null,
17  "created_at": "2024-09-05T21:06:43Z",
18  "updated_at": "2024-09-13T14:14:25Z",
19  "use_parent_bill_rates": false,
20  "type": "Project",
21  "project_state_id": 18668609,
22  "thumbnail": "https://assets.rm.smartsheet.com/images/941116c8-e2da-46b3-a699-28e9ce88df8b.jpg",
23  "owner_id": null,
24  "owner_name": null,
25  "phase_name": null,
26  "has_pending_updates": false,
27  "client": "Echo",
28  "project_state": "Confirmed",
29  "tags": {}
30 }
31
32 "bill_rates": {
33   "paging": {
34     "per_page": 20,
35     "page": 1,
36     "previous": null,
37     "self": "/api/v1/projects/9713578/bill_rates?per_page=20&page=1",
38     "next": null
39   },
40   "data": [
41     {
42       "id": 93192990,
43       "rate": 125.0,
44       "assignable_id": 9713578,
45       "discipline_id": null,
46       "role_id": null,
47       "user_id": null,
48       "starts_at": null,
49       "ends_at": null,
50       "created_at": "2024-09-05T21:06:43Z",
51       "updated_at": "2024-09-05T21:06:43Z",
52       "startdate": null,
53       "enddate": null
54     },
55     {
56       "id": 93192992,
57       "rate": 125.0,
58       "assignable_id": 9713578,
59       "discipline_id": null,
60       "role_id": null,
61       "user_id": 3277658,
62       "starts_at": null,
63       "ends_at": null,
64       "created_at": "2024-09-05T21:06:43Z",
65       "updated_at": "2024-09-05T21:06:43Z",
66       "startdate": null,
67       "enddate": null
68     }
69   ]
70 }
71
```

Status: 200 OK Time: 386 ms Size: 4.44 KB

Pretty Raw Preview Visualize JSON

Request Response

Dynamic Bill Rates

GET Project with Bill Rates

```
{
  "bill_rates": [
    {
      "assignable_id": 9713578,
      "discipline_id": null,
      "id": 93192998,
      "rate": 125,
      "role_id": null
    },
    {
      "assignable_id": 9713578,
      "discipline_id": null,
      "id": 93192992,
      "rate": 125,
      "role_id": null,
      "user_id": 3277858
    },
    {
      "assignable_id": 9713578,
      "discipline_id": null,
      "id": 93192993,
      "rate": 125,
      "role_id": null,
      "user_id": 3277561
    },
    {
      "assignable_id": 9713578,
      "discipline_id": null,
      "id": 93387194,
      "rate": 125,
      "role_id": null,
      "user_id": 3277536
    },
    {
      "assignable_id": 9713578,
      "discipline_id": null,
      "id": 93443827,
      "rate": 125,
      "role_id": null,
      "user_id": 3277560
    },
    {
      "assignable_id": 9713578,
      "discipline_id": null,
      "id": 93443828,
      "rate": 125,
      "role_id": null,
      "user_id": 3277537
    },
    {
      "assignable_id": 9713578,
      "discipline_id": null,
      "id": 93443829,
      "rate": 280,
      "role_id": null,
      "user_id": 3277529
    },
    {
      "assignable_id": 9713578,
      "discipline_id": null,
      "id": 93443830,
      "rate": 125,
      "role_id": null,
      "user_id": 3277538
    }
  ]
}
```

RequestBody Example

Dynamic Bill Rates

PUT Project with new Bill Rates

Dynamic Bill Rates

The Result: Bill Rates

- Elimination of manual effort and associated risks
 - As rates change with promotions, modifying scopes of work, annual increases, etc.
- Enable more precision when applying rates
 - Set multiple rates per resource on projects
- Improve accuracy for budget tracking and billing

Bill Rates

Bill rates for people currently on the project. These rates are derived from the current default bill rates that are specified below or set within a user profile. Rates in **Red** deviate from the default.

Brooklyn Jansen	\$125
Brooklyn Jansen (from October 01, 2024 to December 31, 2024)	\$250
Design	\$125
Harry Riggs	\$125
Paul Finley	\$125
Project Management	\$125
Shirley Huson	\$125
Tamika Marshall	\$200

Dynamic Bill Rates

The Result: Budget Reports

- Updated Bill Rates reflect in Budget reporting
- Bill rates with effective date ranges apply **only** to the assignments within the defined range

Project		Incurred	Future Scheduled	Budget
Arbor Assets	Amounts	\$0.00	\$526,300.00	\$375,500.00
Expenses		\$0.00	-	\$35,500.00
Time & Fees		\$0.00	\$526,300.00	\$340,000.00
Brooklyn Jansen		\$0.00	\$88,100.00	-
Date				
Monday, 09/16/2024		\$0.00	\$500.00	
Tuesday, 09/17/2024		\$0.00	\$500.00	
Wednesday, 09/18/2024		\$0.00	\$500.00	
Thursday, 09/19/2024		\$0.00	\$500.00	
Friday, 09/20/2024		\$0.00	\$500.00	
Monday, 09/23/2024		\$0.00	\$500.00	
Tuesday, 09/24/2024		\$0.00	\$500.00	
Wednesday, 09/25/2024		\$0.00	\$500.00	
Thursday, 09/26/2024		\$0.00	\$500.00	
Friday, 09/27/2024		\$0.00	\$500.00	
Monday, 09/30/2024		\$0.00	\$500.00	
Tuesday, 10/01/2024		\$0.00	\$1,000.00	
Wednesday, 10/02/2024		\$0.00	\$1,000.00	
Monday, 12/30/2024		\$0.00	\$1,000.00	
Tuesday, 12/31/2024		\$0.00	\$1,000.00	
Wednesday, 01/01/2025		\$0.00	\$500.00	
Thursday, 01/02/2025		\$0.00	\$500.00	

Advanced Automation Use Case

Automated Time Tracking



Automated Time Tracking

The Job

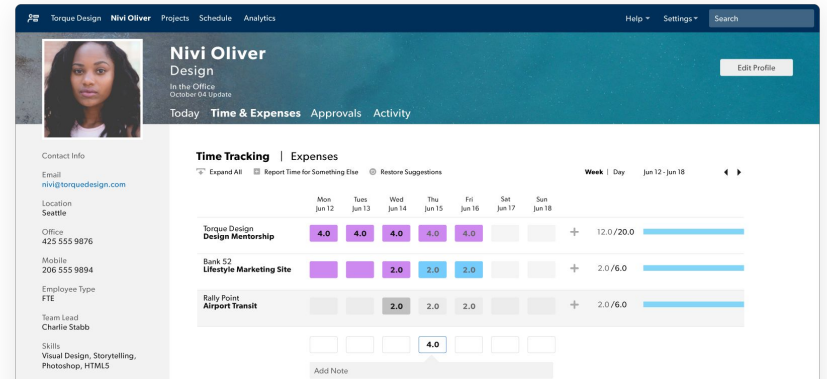
Most resources are juggling many tasks/assignments in a given workday.

The Challenge

Dynamically tracking time throughout the day switching between work streams can be difficult to manage efficiently - ever feel like you need to log the time spend logging time?

The Solution

There are unique services out there that make time tracking fast and nearly thoughtless that can be brought in with API Orchestration.

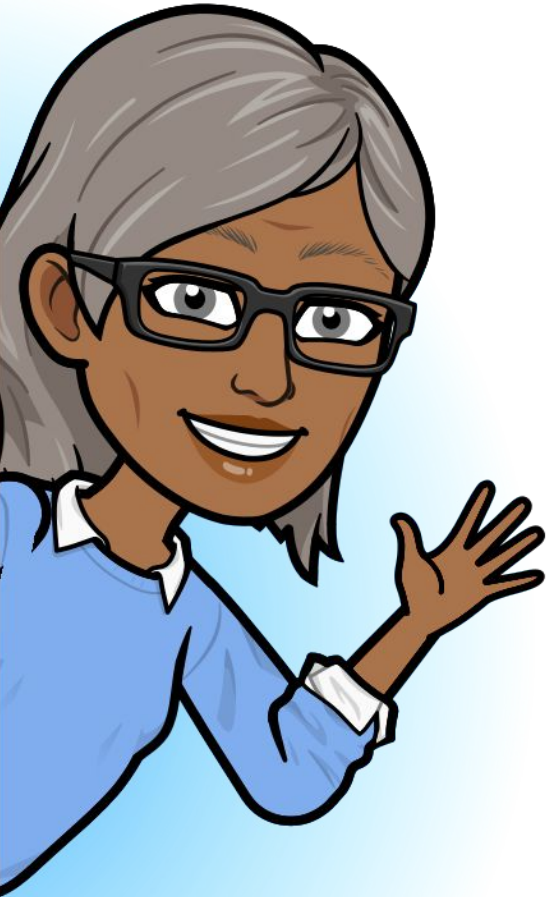


Meet Lex Quill, a
fictitious Software
Engineer

hi



Meet Dr. Evelyn Hawthorne, a fictitious Engineering Manager

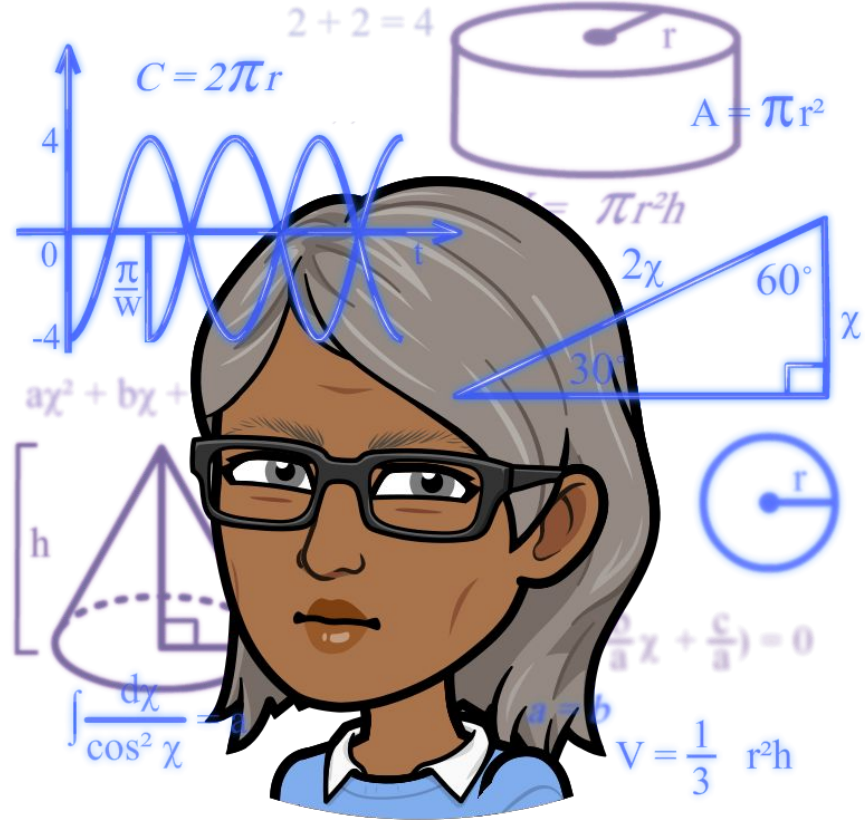




Lex often is working on countless projects and has a hard time keeping on top of his time entries in Resource Management.



Evelyn would like to be able to see clearly which projects are taking most of Lex's time.



Lex tends to wait until the end of the week to enter his time when he knows Evelyn will be looking at RM.

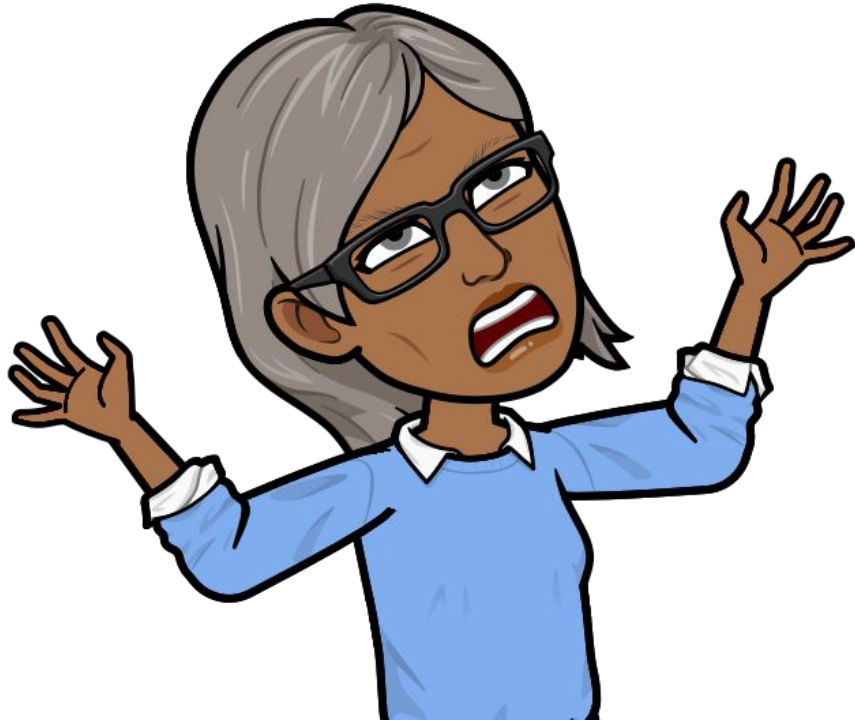




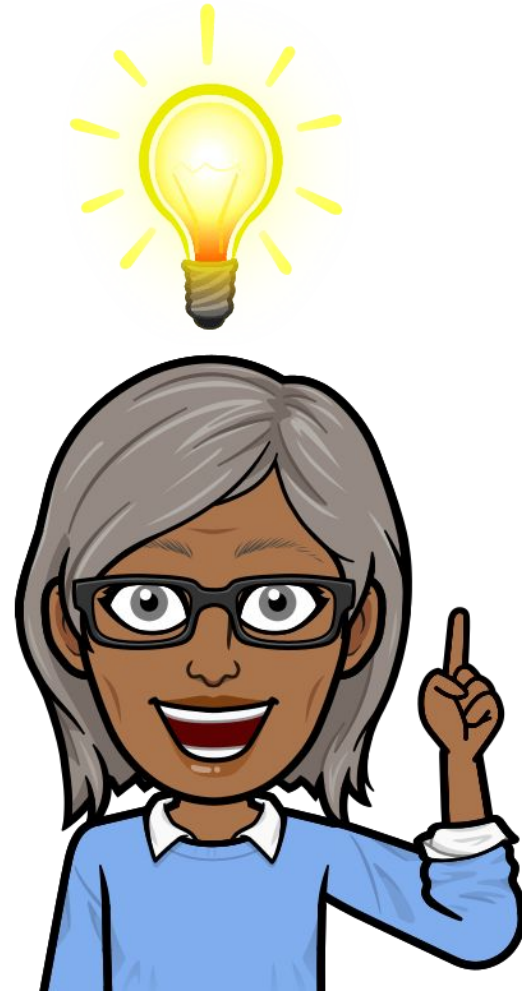
Oh no, Lex has forgotten to keep track! He now has to look back at his calendar and guesstimate how much time he spent on each project.



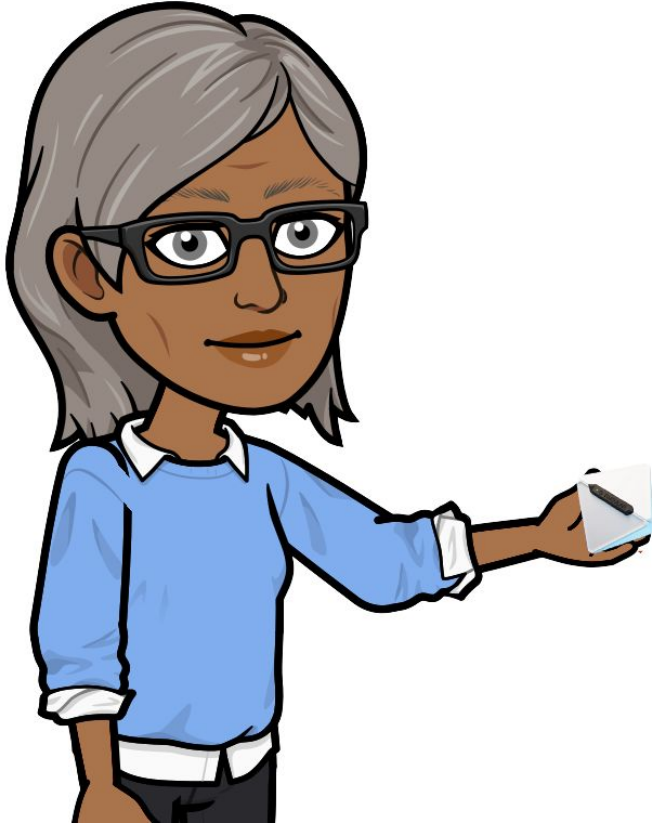
Evelyn is not happy.



She decides to get creative to help Lex better manage his time entry and to give her clear and accurate data to be able to make better decisions.



Evelyn buys Lex a Timeular Tracker



Everyone is happy!!!



demo time



What we covered

RM's new Advanced Reporting capability

How to evaluate external systems for connecting with RM

How APIs can be used to integrate systems and supercharge your solution

Given you some real world use cases, ideas, and inspiration

Next steps...

- **Explore the Power of Advanced Analytics in Resource Management**
 - Unlock hidden potential in your data—start today!
- **Assess Your Current Tools for API Integration Potential**
 - Transform your workflows by connecting the right tools.
- **Implement Smartsheet APIs to Automate and Optimize Forecasting**
 - Take your project planning to the next level with smarter automation.

Take the survey

We'd love to hear your thoughts on the session.

**Open this session in the mobile app, click “Survey,”
and answer two questions — it's that easy!**

Thank you.

 smartsheet

ENGAGE

Share your experience at ENGAGE

by participating in the conversation on social media!
Use **#SmartsheetENGAGE** and tag Smartsheet
in your posts all week long.



@smartsheet



@smartsheetplatform



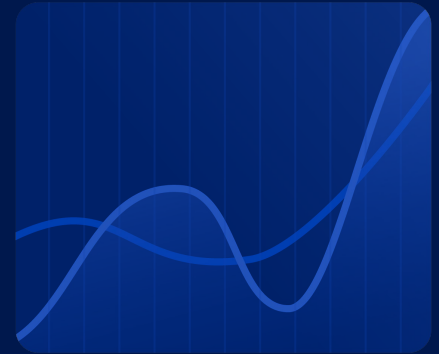
@smartsheet



@smartsheet



@smartsheet



 smartsheet

ENGAGE