



ENGAGE

Rocket fuel for project teams

A deep dive into API integrations and
automations with Smartsheet



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’s 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.

Introductions



Sergio Dominguez

Senior Solutions Architect, Specialized Services



Learning Objectives

1

Understand Smartsheet's API offerings

2

Learn how to get started with the Smartsheet API

3

Gain insight into Integrations and Advanced Automations

4

Understand Smartsheet Webhooks

5

Understand Smartsheet's SDK offerings

1

Smartsheet's API offerings



Smartsheet APIs

Products with API offerings



**Resource
Management**

by  smartsheet



Resource Management

10kft.github.io/10kft-api

The screenshot shows the Smartsheet interface for managing project members. The top navigation bar includes the Smartsheet logo, 'Creative Company', 'Aviv Perez', 'Projects', 'Schedule', and 'Analytics'. On the right, there are links for 'Help', 'Settings', and a search bar.

Project Members

Manage project membership and project access levels. Access levels are exclusive to this project. Some people can edit without membership based on their [account permissions](#).

[Add members](#)

Manage Members (4)

Search...

Name	Access	
<input type="checkbox"/> Aviv Perez	Editor	Project Owner
<input type="checkbox"/> Brooklyn Jansen	Editor	
<input type="checkbox"/> Devon Gomez	Editor	
<input type="checkbox"/> Olivia Carter	Editor	

On the left side of the interface, there is a sidebar menu with the following items: Project Settings, **Project Members**, Phases, Research, Administration, Design, and Production.

Brandfolder

developers.brandfolder.com

Brandfolder

Review Complete

- Download
- Open proof in new tab
- Send to Brandfolder...
- View in Brandfolder

smartsheet
Insights Data

Connect

All Asset Activity

- 7715 Total View Count
- 6921 Total Share Count
- 2199 Total Download Count

Asset Type *
Select or enter value

Description *

Status
Not Started

Start Date

Due Date *

Requested By *
Select or enter value

Priority
Select or enter value

Image Height

Image Width

Image Ratio
Select

Send me a copy of my responses

Submit

Smartsheet REST API

Public documentation

What's available

How it works

Limits

Samples in different programming languages



A screenshot of a web browser displaying the Smartsheet API Reference (2.0.0) documentation. The browser's address bar shows "smartsheet.redoc.ly". The page has a dark blue header with the Smartsheet logo. A left-hand navigation menu lists various sections like "Introduction", "API Basics", "Code Walkthrough", "Error Code List", "OAuth Walkthrough", "Resources", "Security", "Smartsheet Gov", "Smartsheet Regions Europe", "Troubleshooting", "Work at Scale", "Authentication", "ALTERNATE EMAILS", "Alternate Email Addresses Basics", "Objects", "Methods", "ATTACHMENTS", and "Attachments Basics". The main content area is titled "Smartsheet API Reference (2.0.0)" and includes a "Download OpenAPI specification: [Download]" button and the text "Updated 2023-07-20". The "Introduction" section is currently selected and shows text about accessing organizational information and a list of actions: "Read and update sheets", "Manage folders and workspaces", and "Administer user accounts and organization account". It also mentions viewing code examples in the Developer Reference section and provides "NOTES" regarding code samples, feedback, and terms of use. The "API Basics" section is partially visible below, with the "Assume User" subsection starting to appear.

2

Get started with the Smartsheet API



API Documentation: Access

To make API calls, you'll need:

- An API client (Postman/cURL).
- Access tokens.

Remember that tokens act on your behalf.

- Create one for each use case (each app, each project, etc.).
- Keep track of them somewhere safe and revoke/replace periodically.
- Use service account tokens when possible.

Raw Token Requests

If you want to get started quickly, or are developing a standalone application that can run with your credentials, follow these instructions:

1. Click the "Account" button in the lower-left corner of the Smartsheet screen, and then click "Personal Settings".
2. Click the "API Access" tab.
3. Click the "Generate new access token" button to obtain an access token.

The access token must be sent with every API call in an HTTP authorization header (except for the requests to Get Access Token or Refresh Access Token). Once you have an access token, include it in the **Authorization** header for every request you make:

```
Authorization: Bearer JKlMNOpQ12RStUVwxYZAbcde3F5g6hijkLM789
```


The header name is **Authorization** and the value of the header is **Bearer JKlMNOpQ12RStUVwxYZAbcde3F5g6hijkLM789**. Since the access token is being transmitted in clear text, all API calls are done over HTTPS.


NOTE: A best practice is to use a shared account, such as ticket-processor@example.com, rather than your individual work account.

API Documentation: Access


Endpoint

API Host and Version

Smartsheet Public API 2.0 / Serverinfo / **Get Server Info** 

GET  `https://api.smartsheet.com/2.0/serverinfo`

Params Authorization **Headers (8)** Body Pre-request Script Tests Settings

Headers  6 hidden

	Key	Value
<input type="checkbox"/>	Authorization	Bearer <code>{{token}}</code> → API Token
<input checked="" type="checkbox"/>	Content-Type	application/json
	Key	Value

Astrocats, Inc.

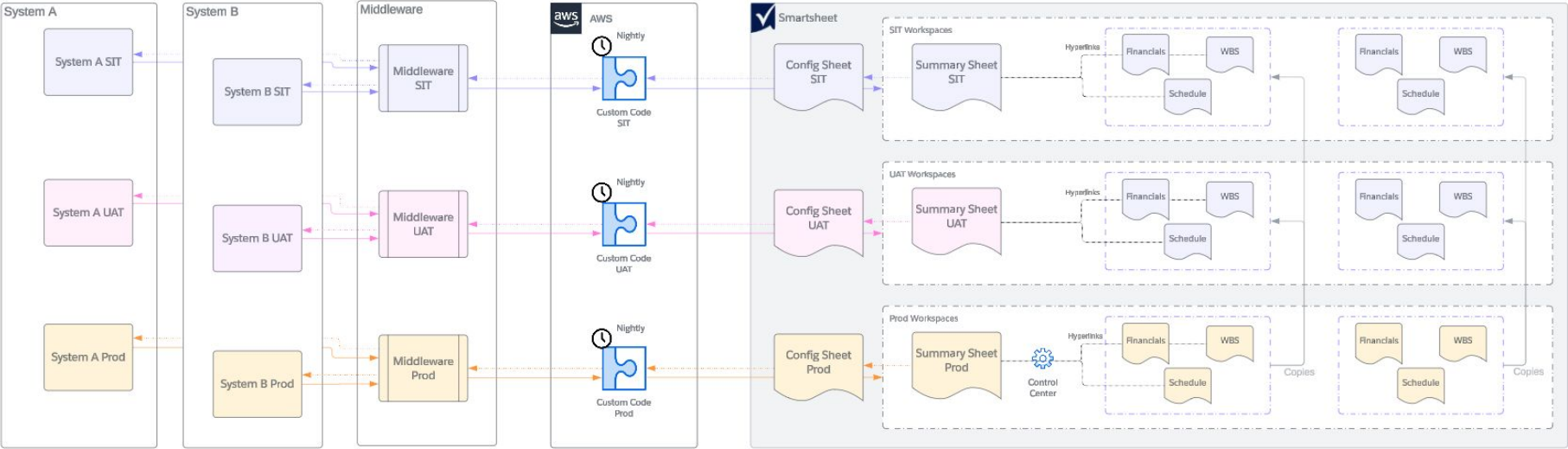
The Astrocats* embark on integrations



*We're not a real company

Astrocats, Inc.

Smartsheet Integration Environments High Level Design



API Documentation: Sheets

Get Sheet - /sheets/{sheetId}

```
1  [
2  "id": 4225731770247044,
3  "name": "test",
4  "version": 19,
5  "totalRowCount": 6,
6  "accessLevel": "OWNER",
7  > "effectiveAttachmentOptions": [
16 ],
17 "ganttEnabled": false,
18 "dependenciesEnabled": false,
19 "resourceManagementEnabled": false,
20 "resourceManagementType": "NONE",
21 "cellImageUploadEnabled": true,
22 > "userSettings": {
25 },
26 > "userPermissions": {
28 },
29 "hasSummaryFields": false,
30 "permalink": "https://app.smartsheet.com/sheets/V9MVX9QpaxmCgcX7Xphc7GWhpgP3J7C6XV6x9m31",
31 "createdAt": "2021-07-23T18:17:27Z",
32 "modifiedAt": "2023-01-11T15:29:25Z",
33 "isMultiPicklistEnabled": true,
34 > "columns": [
105 ],
106 > "rows": [
306 ]
307 ]
```


API Documentation: Sheets

Common endpoints

- Get sheet (list rows, get holidays, aggregate data),
- Get sheet version,
- Copy sheet (save as new),
- List sheets,
- Get sheet as CSV,
- List shares,
- Share sheet

The screenshot displays the Smartsheet API documentation for the 'Get Sheet' endpoint. The interface is divided into several sections:

- Navigation:** A sidebar on the left contains a search bar and a list of API endpoints, with 'Get Sheet' highlighted.
- Endpoint Overview:** The main header 'Get Sheet' is followed by a description: 'Gets a sheet in the format specified, based on the sheet Id.'
- Authorizations:** A section titled 'AUTHORIZATIONS:' specifies '(APIToken) OR (OAuth2 (READ_SHEETS))'.
- Path Parameters:** A section titled 'PATH PARAMETERS:' lists 'sheetId' as a required number representing the 'Sheet Id of the sheet being accessed.'
- Query Parameters:** A section titled 'QUERY PARAMETERS:' lists 'accessApiLevel' (Default: 0) which allows COMMENTER access, and 'include' (string) which is a comma-separated list of optional elements to include in the response. The 'include' section lists various options: attachments, columnType, crossSheetReferences, discussions, filters, filterDefinitions, format, ganttConfig, objectValue, ownerInfo, rowPermalink, source, and writerInfo.
- Request Samples:** A section titled 'Request samples' shows examples for cURL, Node.js, Python, and Ruby, demonstrating how to request a sheet as JSON, Excel, PDF, or CSV.
- Response Samples:** A section titled 'Response samples' shows a 200 status code and a 'default' content type of 'application/json', with a partial JSON response snippet: { "version": 0 }.

Standard Cell Data

Strings **ABC**

Text
Hello, world 1

```
{  
  "columnId": 525094551244676,  
  "value": "Hello, world 1",  
  "displayValue": "Hello, world 1"  
},
```

Numbers **123**

Numbers
123.456

```
{  
  "columnId": 5028694178615172,  
  "value": 123.456,  
  "displayValue": "123.456"  
},
```

Booleans **True/False**

Boolean
<input checked="" type="checkbox"/>

```
{  
  "columnId": 2776894364929924,  
  "value": true  
},
```

*value can be absent

```
{  
  "columnId": 120343275720580  
},
```

Standard Cell Data

Dates

2024-10-08

Standard Date
10/08/24

```
{ } Untitled-1 • { } no params ↔ ?level=2&include=objectValue • { } Untitled-2 •
276 {
277   "columnId": 1896707389607812,
278   "value": "2024-10-08"
279 }
335 {
336   "columnId": 1896707389607812,
337+   "value": "2024-10-08",
338+   "objectValue": {
339+     "objectType": "DATE",
340     "value": "2024-10-08"
341+   }
342 }
```

Standard Cell Data

Start/End Dates (w/ Dependencies)

2024-10-23T16:59:59

End
10/15/24
10/23/24

```
{ } Untitled-1 • { } no params ↔ ?level=2&include=objectValue • { } Untitled-2 •
252 {
253   "columnId": 1650994458087300,
254   "value": "2024-10-23T16:59:59"
255 },
284 {
285   "columnId": 1650994458087300,
286+   "value": "2024-10-23T16:59:59",
287+   "objectValue": {
288+     "objectType": "ABSTRACT_DATETIME",
289     "value": "2024-10-23T16:59:59"
290+   }
291 },
```

Standard Cell Data

Dropdowns

"A, B, C"



```
{ } Untitled-1 • {} no params ↔ ?level=2&include=objectValue • {} Untitled-2 •
261 {
262   "columnId": 894167533440900,
263-  "value": "A, B, C",
264  "displayValue": "A, B, C",
265 },
298 {
299   "columnId": 894167533440900,
300+  "objectValue": {
301+    "objectType": "MULTI_PICKLIST",
302+    "values": [
303+      "A",
304+      "B",
305+      "C"
306+    ]
307+  },
308   "displayValue": "A, B, C",
309 }
```

Exceptional Cell Data

```
{
  "columnId": 120343275720580,
  "value": "1",
  "objectValue": {
    "objectType": "PREDECESSOR_LIST",
    "predecessors": [
      {
        "rowId": 6795987509546884,
        "rowNumber": 1,
        "type": "FS",
        "inCriticalPath": true
      }
    ]
  },
  "displayValue": "1"
},
```

```
{
  "columnId": 4623942903091076,
  "value": "6d",
  "objectValue": {
    "objectType": "DURATION",
    "days": 6.0
  },
  "displayValue": "6d"
},
```

```
{
  "columnId": 6435560108543876,
  "value": "sergio.dominguez@smartsheet.com",
  "objectValue": {
    "objectType": "CONTACT",
    "email": "sergio.dominguez@smartsheet.com",
    "name": "Sergio Dominguez",
    "imageId": "u!1!aUsrmj4dccc!nhEqv54nF2s!bc4z3Y5N9Mr"
  },
  "displayValue": "Sergio Dominguez"
},
{
  "columnId": 4183760294858628,
  "objectValue": {
    "objectType": "MULTI_CONTACT",
    "values": [
      {
        "objectType": "CONTACT",
        "email": "sergio.dominguez@smartsheet.com",
        "name": "Sergio Dominguez",
        "imageId": "u!1!aUsrmj4dccc!nhEqv54nF2s!bc4z3Y5N9Mr"
      },
      {
        "objectType": "CONTACT",
        "email": "sergio.dominguez+internaladmin@smartsheet.com",
        "name": "sergio.dominguez+internaladmin@smartsheet.com"
      }
    ]
  },
  "displayValue": "Sergio Dominguez, sergio.dominguez+internaladmin@smartsheet.com"
},
```

3

Integrations and Advanced Automations

A primer

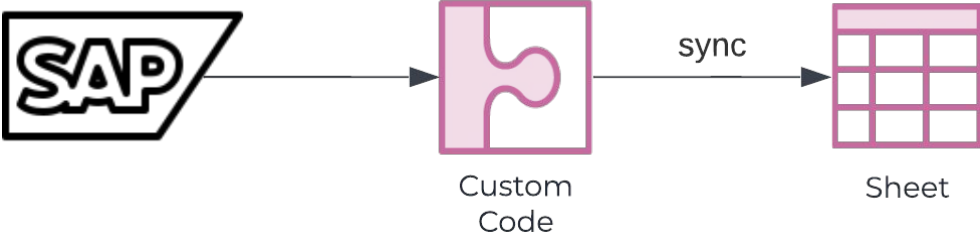
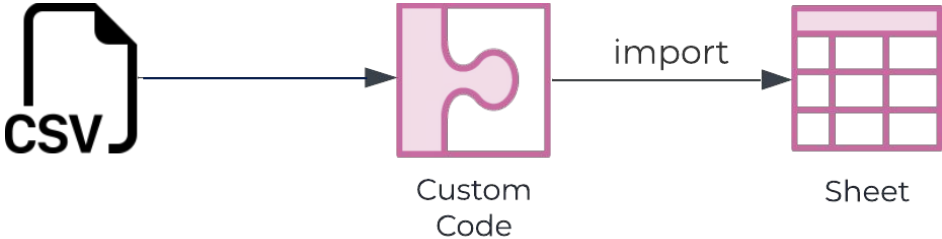


What can I do with the Smartsheet API?

So glad you asked...

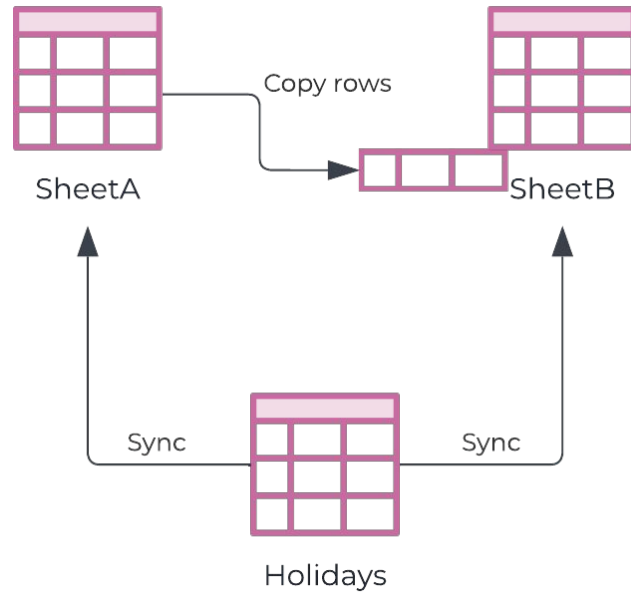


Transfer data

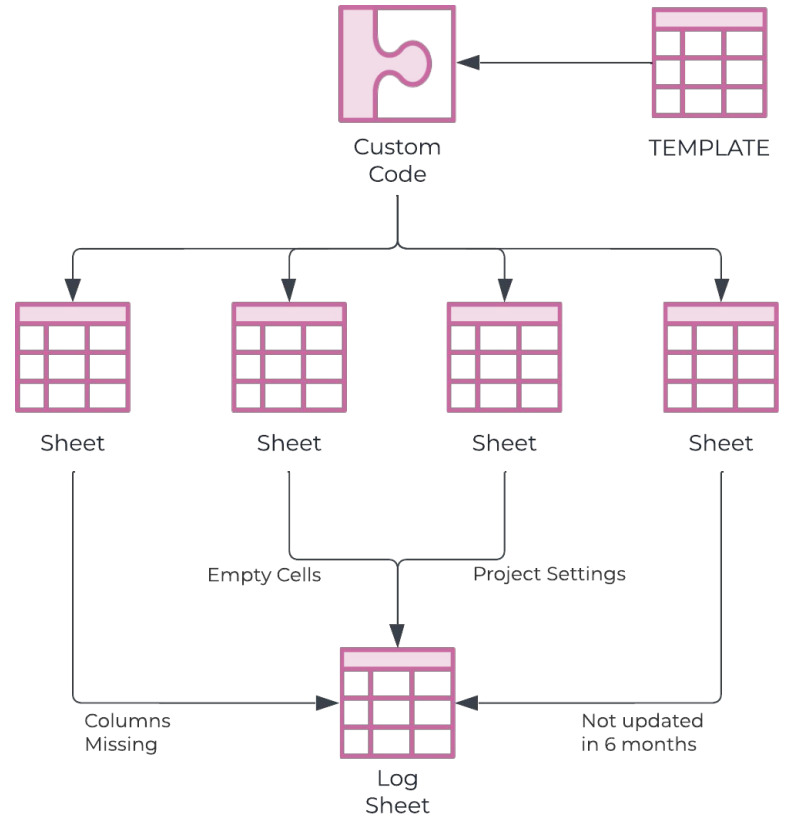
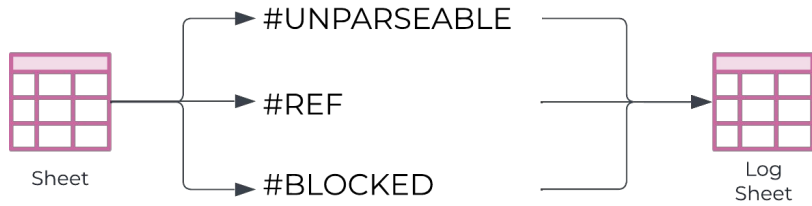


Automate your Mondays

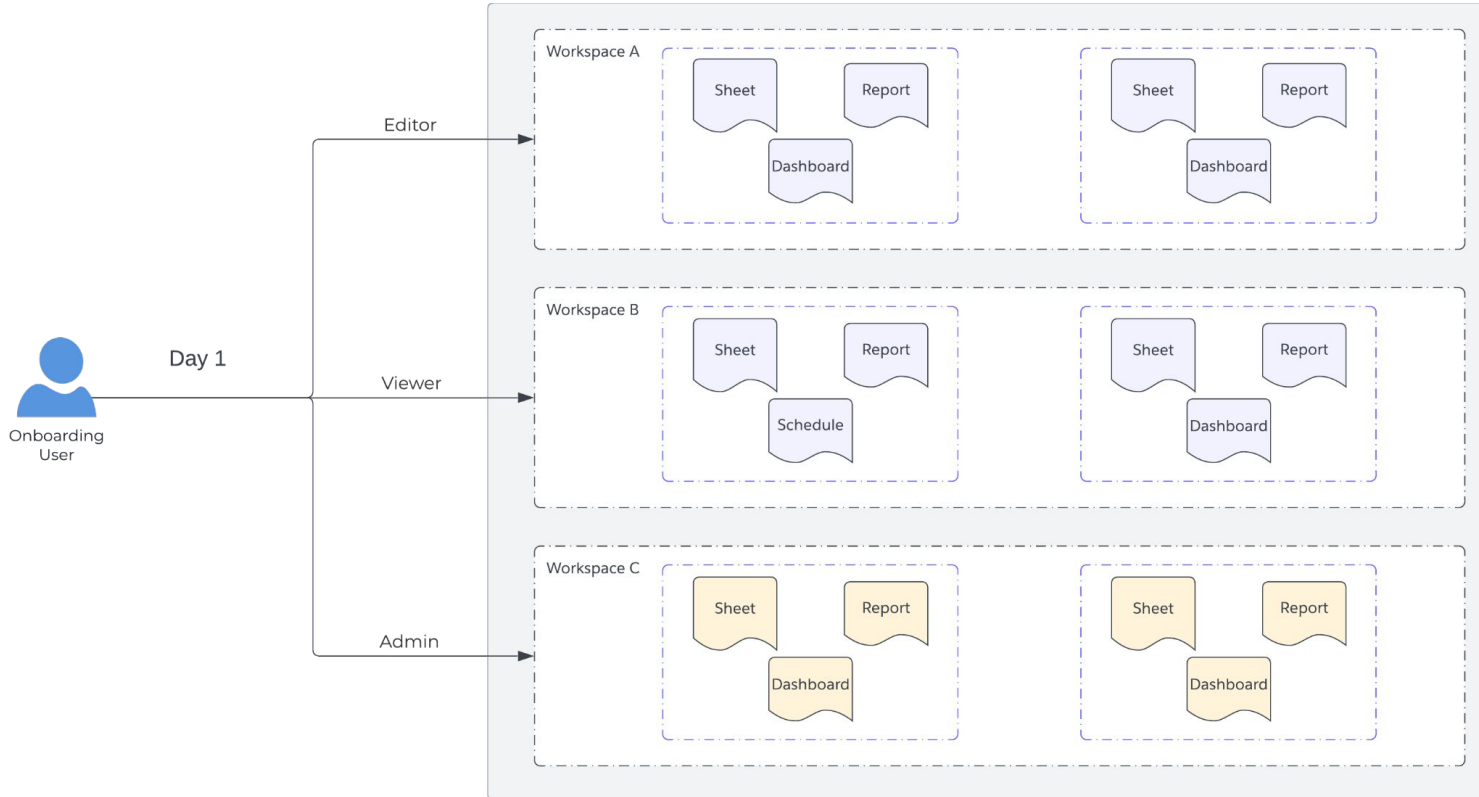
If you've done it manually more than once, it should be automated.



Quality checks



Manage permissions





“

With great

power

comes great responsibility.”

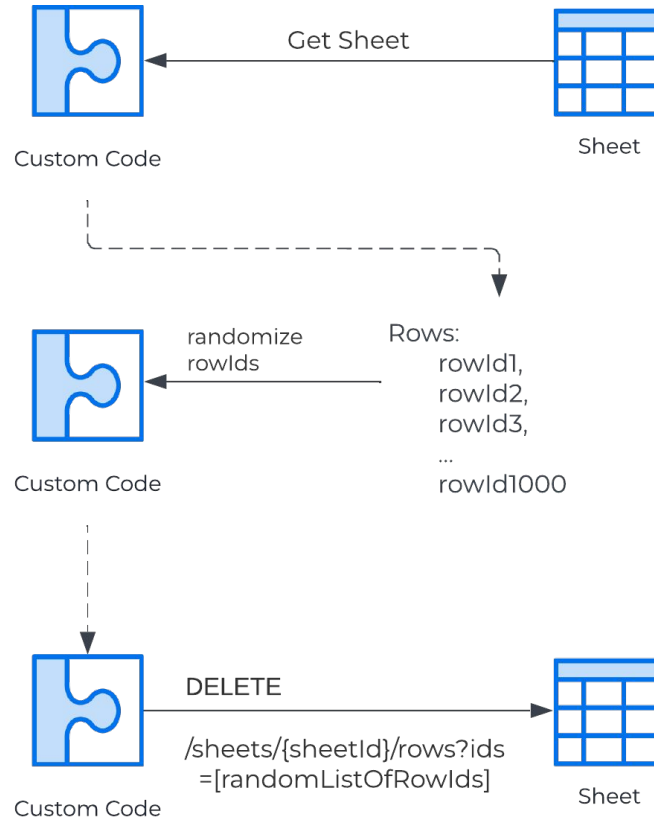
— “Ben Parker”

— Sergio Dominguez

 smartsheet
ENGAGE

Be an agent of chaos

But, seriously, don't.



Developing API integrations

Things you'll need:

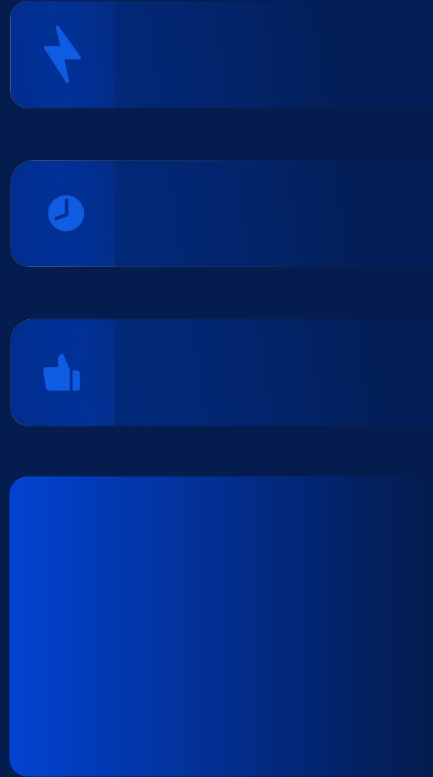
- Software development experience (Python, JavaScript, C#, Java, Ruby).
- Resources to develop the application(s).
- Resources to maintain the application(s).
- Hosting solutions (AWS, Google Cloud, Microsoft Azure).

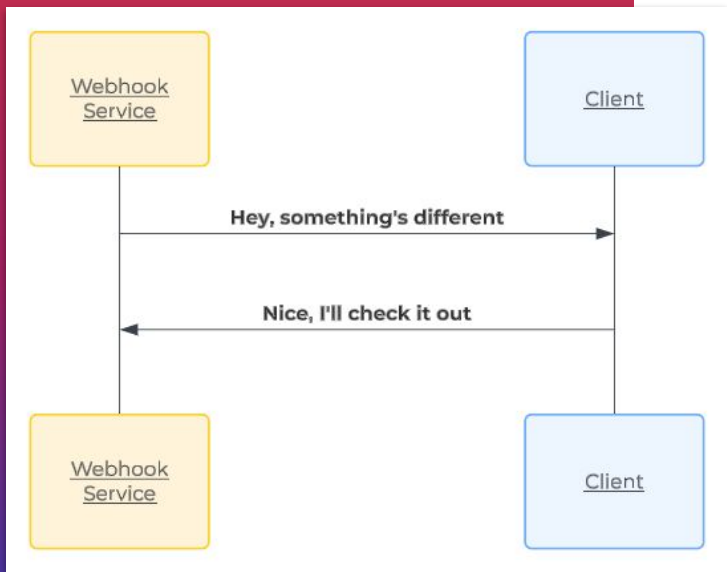
Things you can do:

- Supercharge your Smartsheet projects

4

Smartsheet Webhooks





Webhooks

Holler If You Hear Me

Using Webhooks



Registration



Verification



**Anatomy of a
Callback Event**



Processing Events



Duplicate Webhooks

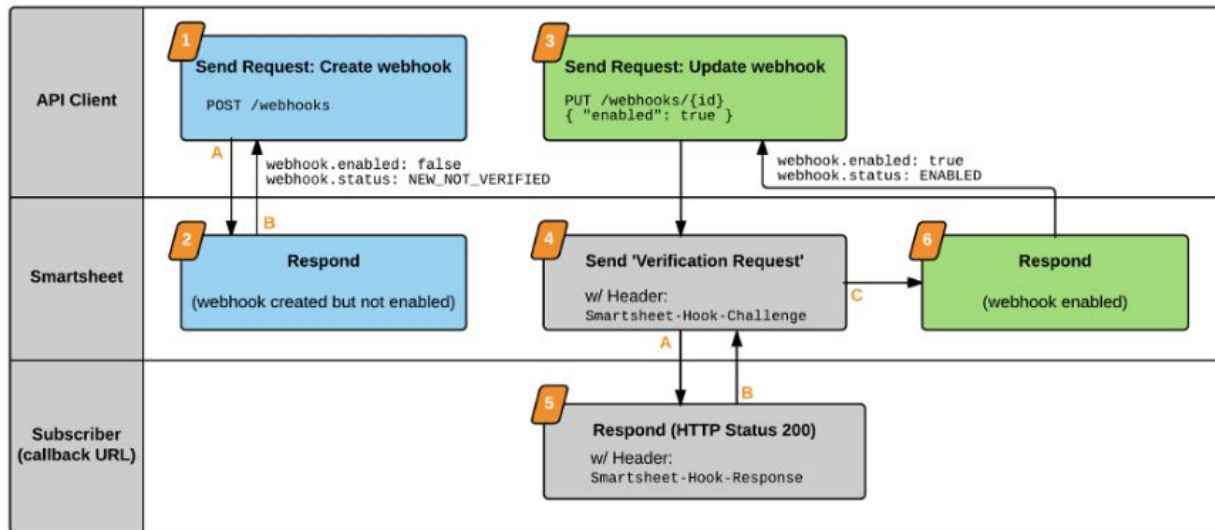


**Callback
Authentication**

Registration & Verification

Creating a Webhook

An API client can create a webhook by using the [Create Webhook](#) operation. A newly created webhook is disabled by default. After successfully creating a webhook, the API client must subsequently initiate and complete the [verification process](#) before the webhook is enabled. The following diagram illustrates the process of creating and enabling a webhook.

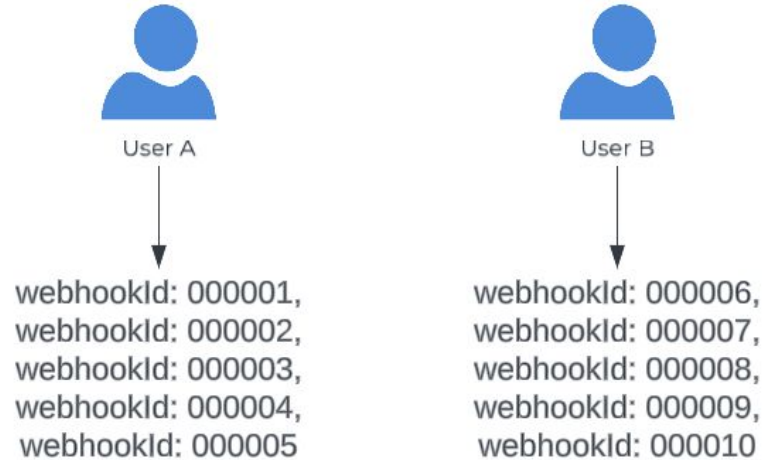


Webhook Management

Owned by Smartsheet user who created them

Can only be viewed/edited/deleted by owner

Not API token-dependent



Anatomy of a Callback Event

Callback Event

```
1 {
2   ... "nonce": "f0377980-ba92-43cd-8dcd-443414347967",
3   ... "timestamp": "2024-07-17T23:54:38.660+00:00",
4   ... "webhookId": 3163967318189956,
5   ... "scope": "sheet",
6   ... "scopeObjectId": 1636581748199300,
7   ... "events": [
8     {
9       ... "objectType": "cell",
10      ... "eventType": "updated",
11      ... "rowId": 7663406202064772,
12      ... "columnId": 607709735899012,
13      ... "userId": 4316535717160836,
14      ... "timestamp": "2024-07-17T23:54:33.000+00:00"
15    },
16  ],
17  ... "objectType": "cell",
18  ... "eventType": "updated",
19  ... "rowId": 7663406202064772,
20  ... "columnId": 8429653726285700,
21  ... "userId": 4316535717160836,
22  ... "timestamp": "2024-07-17T23:54:33.000+00:00"
23  ]
24 }
25 }
```

SheetId

A cell has changed

Processing Events

Check for challenge

```
headers = Object {host: "04b4-136-226-80-167.ngrok.io", user-agent: "Apache-HttpClient/4.5.1 (Java/11.0.18)"}
  host = "04b4-136-226-80-167.ngrok.io"
  user-agent = "Apache-HttpClient/4.5.1 (Java/11.0.18)"
  content-length = "81"
  accept-encoding = "gzip,deflate"
  content-type = "application/json; charset=UTF-8"
  smartsheet-hook-challenge = "52a70b68-6cdf-446d-aef9-ebba5b556410"
  x-datadog-parent-id = "504874062784070078"
  x-datadog-sampling-priority = "1"
  x-datadog-trace-id = "122166070470510356"
  x-forwarded-for = "54.209.130.254"
  x-forwarded-proto = "https"
  __proto__ = Object
```

Group events by rowId

```
Callback Event - Group
By SheetId/RowId

1 {
2   "nonce": "f0377980-ba92-43cd-8dcd-443414347967",
3   "timestamp": "2024-07-17T23:54:38.660+00:00",
4   "webhookId": "3163967318189956",
5   "scope": "sheet",
6   "scopeObjectId": "1636581748199300",
7   "events": [
8     {
9       "objectType": "cell",
10      "eventType": "updated",
11      "rowId": "7663406202064772",
12      "columnId": "607709735899012",
13      "userId": "4316535717160836",
14      "timestamp": "2024-07-17T23:54:33.000+00:00"
15    },
16    {
17      "objectType": "cell",
18      "eventType": "updated",
19      "rowId": "7663406202064772",
20      "columnId": "8429653726285700",
21      "userId": "4316535717160836",
22      "timestamp": "2024-07-17T23:54:33.000+00:00"
23    }
24  ]
25 }
```

Webhook retries with exponential backoff

Duplicate Webhooks

Duplicate webhooks are defined as two or more webhooks that meet the following conditions:

- on the same sheet
- created by the same user
- have identical: **callbackUrl** and **subscope** properties.

As of January 2023, Smartsheet will deduplicate webhook callbacks. This means for every set of duplicate webhooks on a sheet, Smartsheet will no longer issue one callback per webhook, and will instead only issue **one total callback**, for only the oldest enabled webhook.

Recommendation: When creating a webhook, first check to make sure that the sheet doesn't already contain an identical webhook. If it does, and it's disabled, re-enable it. If it's already enabled, do nothing.

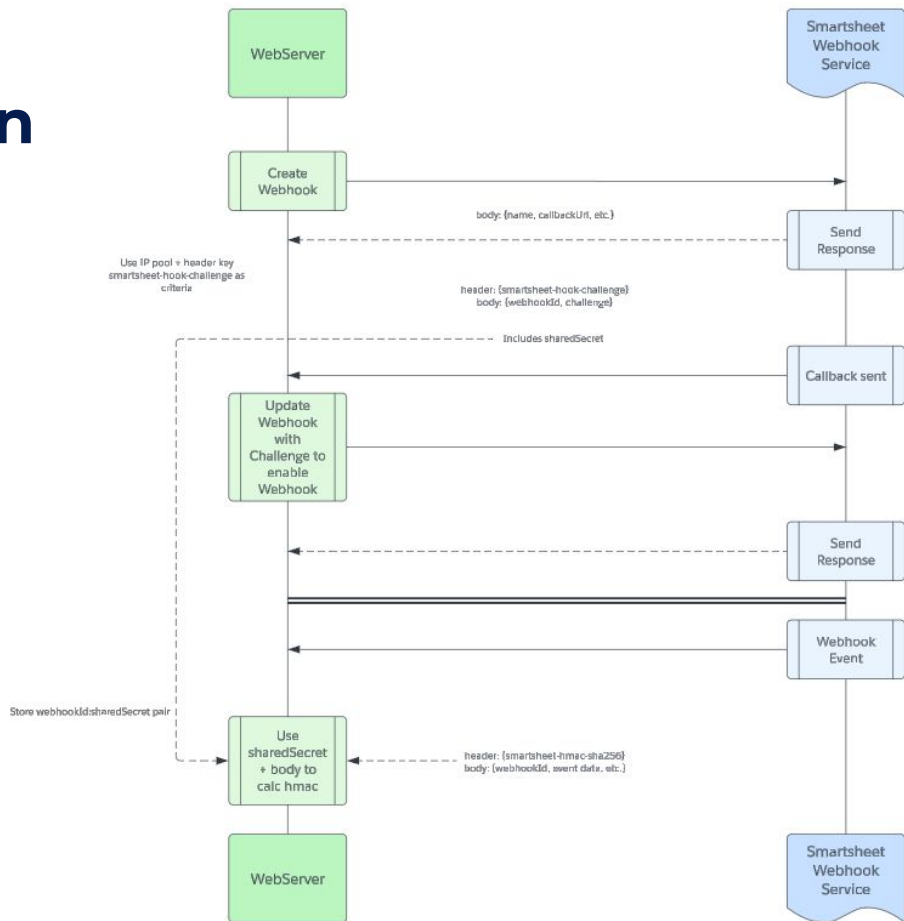
Callback Authentication

HMAC Validation

smartsheet-hmac-sha256: "xyz"

Callback URL parameter

<https://www.myserver.com/endpoint?param=XYZ>



5

Smartsheet's SDK offerings



Smartsheet Software Development Kits (SDKs)

Language	SDK	Sample application
C#	smartsheet-csharp-sdk	csharp-read-write-sheet
Java	smartsheet-java-sdk	java-read-write-sheet
Node.js	smartsheet-javascript-sdk	node-read-write-sheet
Python	smartsheet-python-sdk	python-read-write-sheet
Ruby*	smartsheet-ruby-sdk	ruby-read-write-sheet

Benefits include:

- Retry with backoff to automatically recover from network errors or rate limiting
- Request & response logging
- Native object models for request and responses (Java and C# only)

To use an SDK for development work, follow the instructions in the SDK readme to download and install the SDK. Then download the sample app and run it. Once you've run the sample application, you can clone it and use the structure it provides to start building your own applications.

*The Ruby SDK is no longer maintained. It is still usable, but new functionality may be missing.

Add Rows (cURL)

Command Line HTTP Request

```
curl https://api.smartsheet.com/2.0/sheets/{sheetId}/rows \  
-H "Authorization: Bearer JKLMNOpQ12RStUVwxYZabcde3F5g6hijkLM789" \  
-H "Content-Type: application/json" \  
-X POST \  
-d '{"toTop":true, "cells": [ {"columnId": 7960873114331012, "value": true}, {"columnId": 642523719853956, "value": "New status", "strict": false} ] }, {"toTop":true, "cells": [ {"columnId": 7960873114331012, "value": true}, {"columnId": 642523719853956, "value": "New status", "strict": false} ] }'
```

```
{  
  "message": "SUCCESS",  
  "resultCode": 0,  
  - "result": [  
    - {  
      "id": 0,  
      "sheetId": 0,  
      "rowNumber": 1,  
      "expanded": true,  
      "createdAt": "2019-08-24T14:15:22Z",  
      "modifiedAt": "2019-08-24T14:15:22Z",  
      - "cells": [  
        - {  
          "columnId": 0,  
          "columnType": "string",  
          "value": "string",  
          "displayValue": "string"  
        }  
      ],  
      "version": 0  
    }  
  ]  
}
```

Add Rows - C#

```
// Specify cell values for first row
Cell[] cellsToInsert = new Cell[]
{
    new Cell
    {
        ColumnId = addedColumns[0].Id,
        ObjectValue = new MultiPicklistObjectValue(new string[] { "Bat", "Cat" })
    }
};
Row rowA = new Row
{
    ToTop = true,
    Cells = cellsToInsert
};

// Specify cell values of second row
Cell[] cellsB = new Cell[] {
    new Cell
    {
        ColumnId = 7960873114331012,
        Value = true
    },
    new Cell
    {
        ColumnId = 642523719853956,
        Value = "New status"
    }
};

// Specify contents of second row
Row rowB = new Row
{
    ToTop = true,
    Cells = cellsB
};

// Add rows to sheet
IList<Row> newRows = smartsheet.SheetResources.RowResources.AddRows(
    2331373580117892, // sheetId
    new Row[] { rowA, rowB } // IEnumerable<Row> rowsToAdd
);
```

Java

```
// Specify cell values for first row
Cell[] cellsToInsert = new Cell[]
{
    new Cell
    {
        ColumnId = addedColumns[0].Id,
        ObjectValue = new MultiPicklistObjectValue(new string[] { "Bat", "Cat" })
    }
};
Row rowA = new Row
{
    ToTop = true,
    Cells = cellsToInsert
};

// Specify cell values of second row
Cell[] cellsB = new Cell[] {
    new Cell
    {
        ColumnId = 7960873114331012,
        Value = true
    },
    new Cell
    {
        ColumnId = 642523719853956,
        Value = "New status"
    }
};

// Specify contents of second row
Row rowB = new Row
{
    ToTop = true,
    Cells = cellsB
};

// Add rows to sheet
IList<Row> newRows = smartsheet.SheetResources.RowResources.AddRows(
    2331373580117892, // sheetId
    new Row[] { rowA, rowB } // IEnumerable<Row> rowsToAdd
);
```

NodeJS

```
// Specify rows
var row = [
    {
        "toTop": true,
        "cells": [
            {
                "columnId": 7960873114331012,
                "value": true
            },
            {
                "columnId": 642523719853956,
                "value": "New status",
                "strict": false
            }
        ]
    },
    {
        "toTop": true,
        "cells": [
            {
                "columnId": 7960873114331012,
                "value": true
            },
            {
                "columnId": 642523719853956,
                "value": "New status",
                "strict": false
            }
        ]
    }
];

// Set options
var options = {
    sheetId: 2252168947361668,
    body: row
};

// Add rows to sheet
smartsheet.sheets.addRow(options)
    .then(function(newRows) {
        console.log(newRows);
    })
    .catch(function(error) {
        console.log(error);
    });
```

Add Rows - Python

```
# Specify cell values for one row
row_a = smartsheet.models.Row()
row_a.to_top = True
row_a.cells.append({
    'column_id': 7960873114331012,
    'value': True
})
row_a.cells.append({
    'column_id': 642523719853956
    'value': 'New Status',
    'strict': False
})

# Specify cell values for another row
row_b = smartsheet.models.Row()
row_b.to_top = True
row_b.cells.append({
    'column_id': 7960873114331012,
    'value': True
})
row_b.cells.append({
    'column_id': 642523719853956
    'value': 'New Status',
    'strict': False
})

# Add rows to sheet
response = smartsheet_client.Sheets.add_rows(
    2331373580117892,      # sheet_id
    [row_a, row_b])
```

Ruby

```
# Specify values
body = {
  to_bottom: true,
  cells: [
    {
      column_id: 495660408563588,
      value: 'New Status'
    }
  ],
  locked: false
}

# Add rows to a sheet
response = smartsheet.sheets.rows.add(
  sheet_id: 2331373580117892,
  body: body
)
```

Limits and error handling

6

API limits



Rate Limits

There is a limit of 300 calls per access token per minute.



Concurrent updates

No single access token can perform parallel updates on the same sheet at the same time.



Attachment size (30MB)

The max file size for uploads through the API is limited to 30MB.

Error handling

Error handling

Error codes

Error Code List

For an explanation of the logic behind Smartsheet error codes and error handling, see the [HTTP and REST](#) portion of the Introduction.

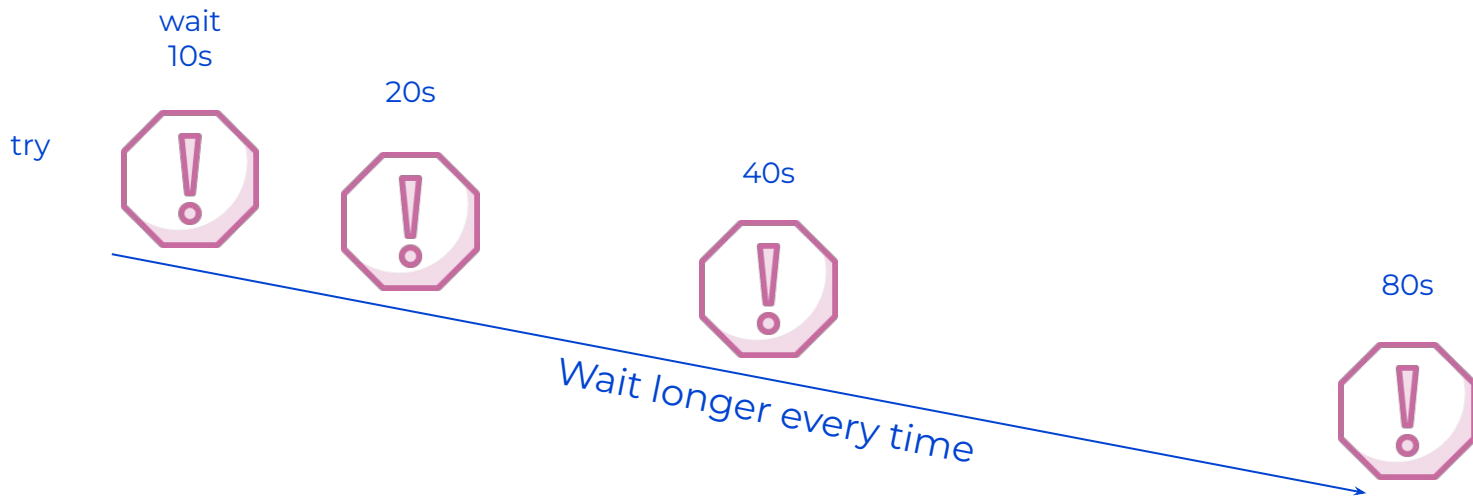
400-Level Error Codes

400-level error codes generally indicate that there is something you should fix or add to your request before you try the request again.

HTTP status code	Smartsheet errorCode	Smartsheet message
401	1001	An Access Token is required.
401	1002	Your Access Token is invalid.
401	1003	Your Access Token has expired.
403	1004	You are not authorized to perform this action.
401	1005	Single Sign-On is required for this account.
404	1006	Not Found.
404	1007	Version not supported.
400	1008	Unable to parse request. The following error occurred: {0}
400	1009	A required parameter is missing from your request: {0}.
405	1010	HTTP Method not supported.
400	1011	A required header was missing or invalid: {0}
400	1012	A required object attribute is missing from your request: {0}.
403	1013	The operation you are attempting to perform is not supported by your plan.
403	1014	There are no licenses available on your account.
403	1015	The user exists in another account. The user must be removed from your account.

Error handling

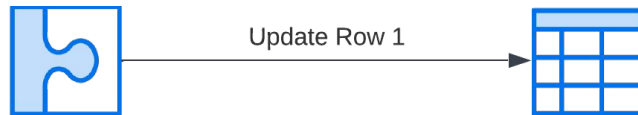
Exponential backoff



Best practices

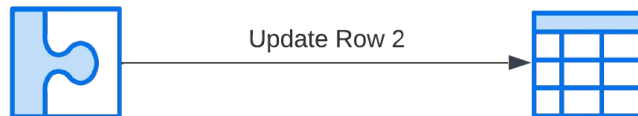
7

Optimize your actions



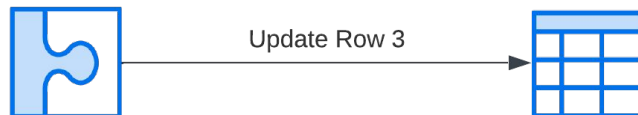
Custom Code

Sheet



Custom Code

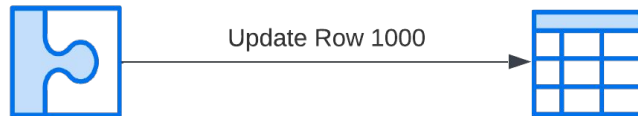
Sheet



Custom Code

Sheet

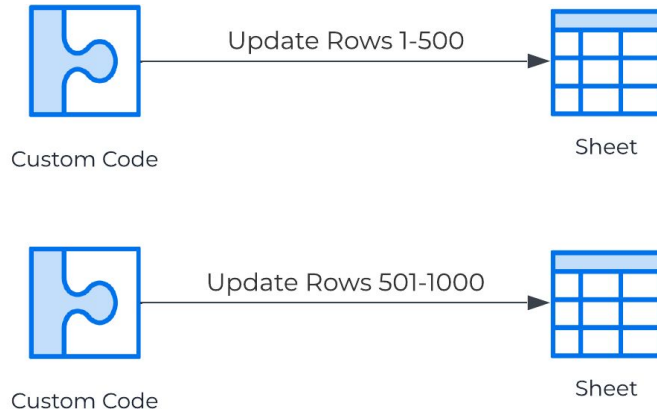
...



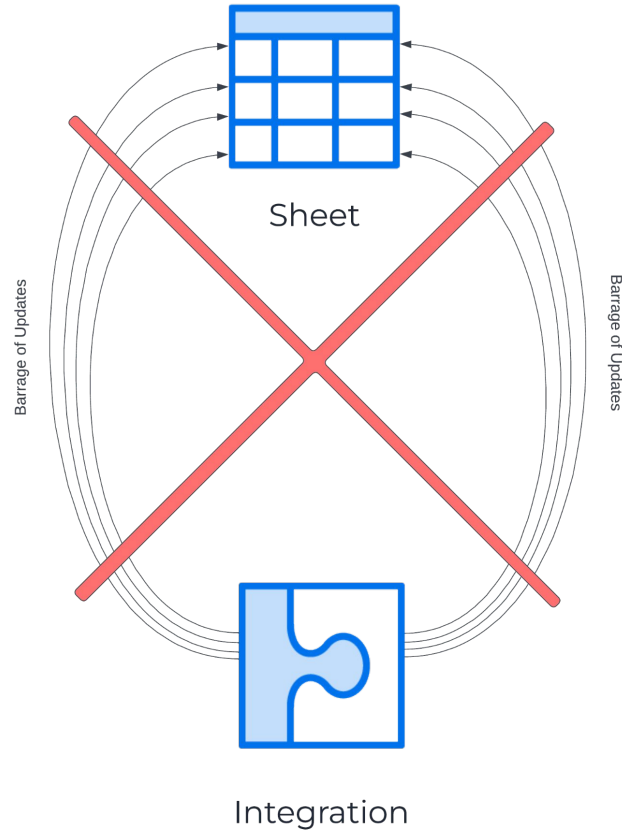
Custom Code

Sheet

Optimize your actions



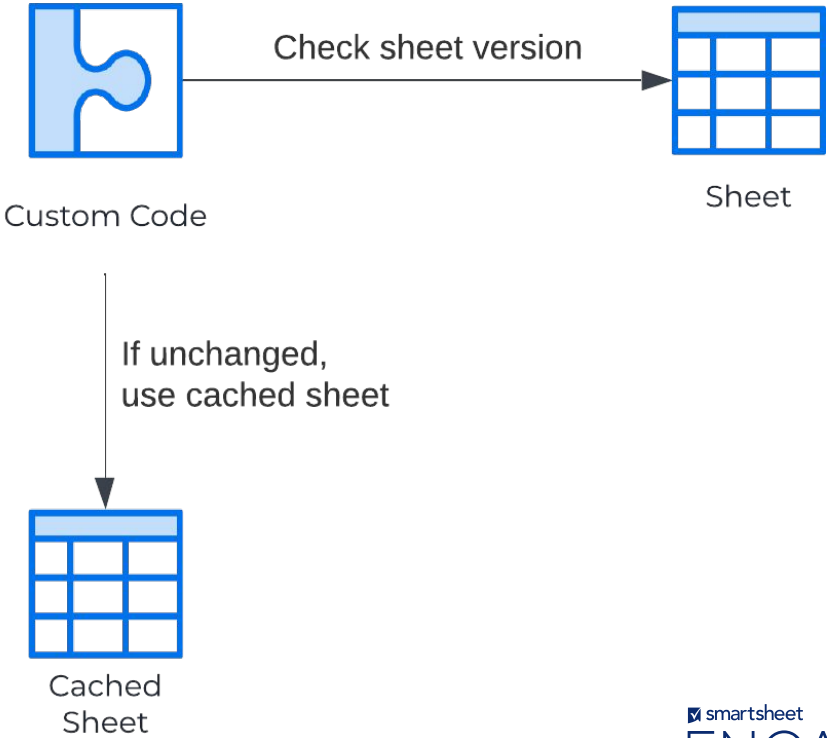
Rapid-fire updates



Be mindful of your impact

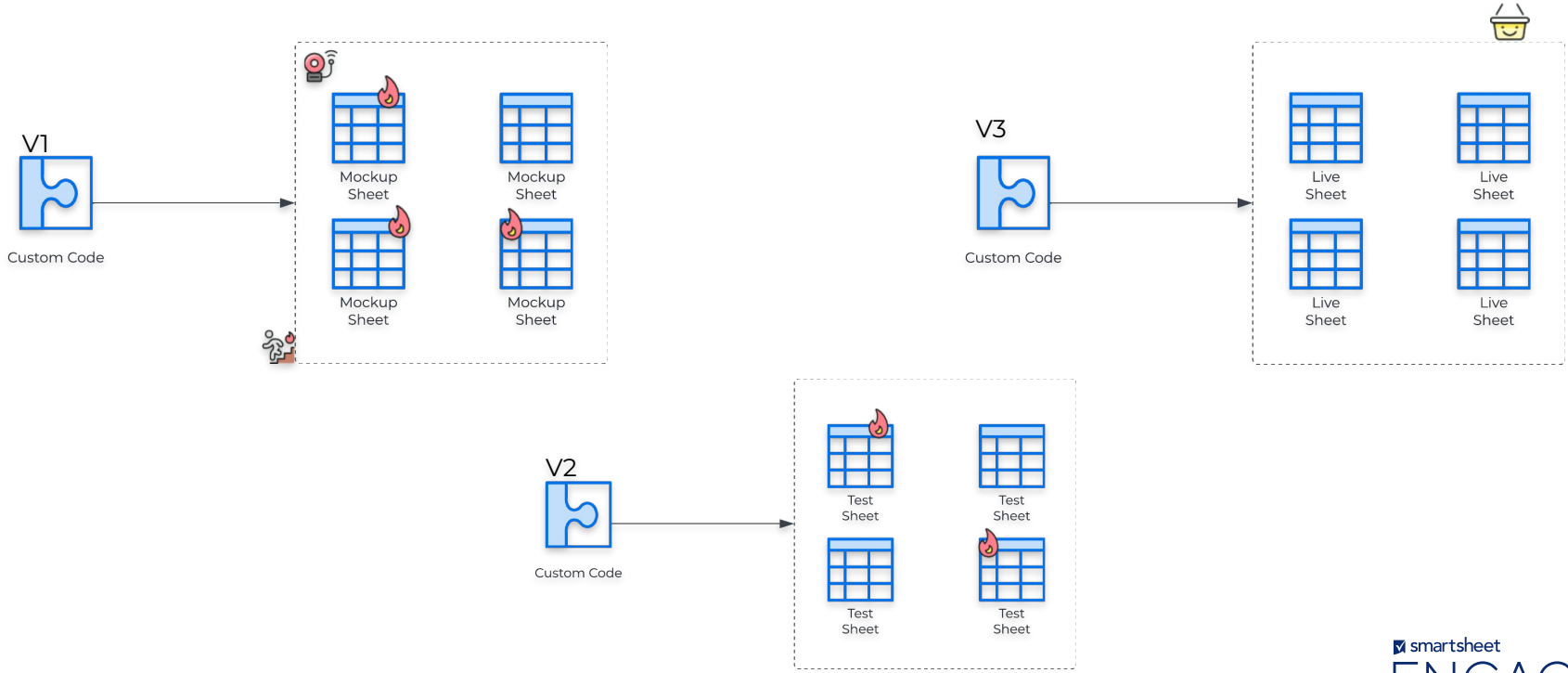


Cache me if you can



Test, test, test

And then test some more!



Learning Objectives

1

Understand Smartsheet's API offerings

2

Learn how to get started with the Smartsheet API

3

Gain insight into Integrations and Advanced Automations

4

Understand Smartsheet Webhooks

5

Understand Smartsheet's SDK offerings

8

Thank you

smartsheet
ENGAGE

Register for upcoming User Groups



Continue to expand your Smartsheet skills and connections by attending a **User Group**.

Smartsheet User Groups

- **Discover** how others are using Smartsheet
- **Network** with the Smartsheet team and your peers
- **Virtual** and **in-person** events around the globe

Next steps...

- Think of all the ways that your team can benefit from APIs.
- Review the documentation.
- Take a look for yourself.

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



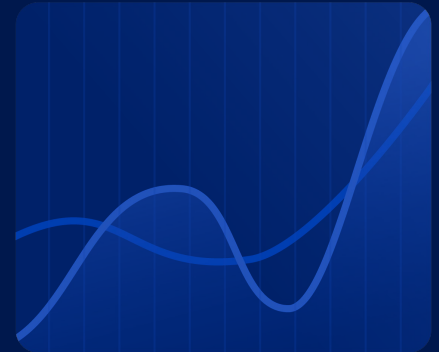
@smartsheet



@smartsheet



@smartsheet



smartsheet

ENGAGE