

salesforce

The Magic of APIs

David Reed
Principal Member of Technical Staff,
Salesforce

@aorisdual | david@ktema.org
he/him



Agenda

What is an API?

Key concepts and patterns

Getting hands-on with Postman

APIs and Admin Workflows

Bonus: become a wizard with Python

Resources

Why It Matters to Me

Came to Salesforce as a nonprofit admin

Chose Salesforce specifically for API

APIs let me replace RE, save >\$10k/yr

Autonomy, open ecosystem



Assumptions

You are a Salesforce administrator, declarative developer, business analyst, or similar role.

You know Salesforce thoroughly.

You're curious about APIs: how they're used in applications, and how they can unlock more power to solve problems.

You don't write code (yet).

Let's start with user-interface metaphors



Here's Classic

Resource Owner (Endpoint)

Resource Identification

https://**david-ktema-dev-ed.my.salesforce.com**/**0011R00002jqGdB**

[Assigned Territories \(1\)](#) | [Users in Assigned Territories \(0\)](#) | [Contacts \(0\)](#) | [Opportunities \(0\)](#) | [Open Activities \(0\)](#) | [Activity History \(0\)](#) | [Cases \(15+\)](#) | [Files \(0\)](#) | [Notes & Attachments \(0\)](#) | [Notes \(0\)](#) | [Account Team \(0\)](#) | [Contacts \(0\)](#) | [Approval History \(0\)](#)

Account Detail

[Edit](#) [Delete](#) [Sharing](#) [Include Offline](#) [Submit for Approval](#) [Manage External Account](#) ▾

Account Name	Dewey, Cheatham, and Howe [View Hierarchy]	Phone	
Parent Account		Website	
Hourly Billing Rate		Industry	
External Id			
URL	conf		
T	0031		
Test			
Billing Address	Cartalk Plaza United States	Shipping Address	93324 Best Trail Apt. 275 Harryborough, Wyoming 88394 United States
Created By	David Reed , 9/29/2021 5:22 PM	Last Modified By	David Reed , 7/17/2022 12:14 PM
		Account Record Type	Client [Change]
Description			

Resource Content

[Edit](#) [Delete](#) [Sharing](#) [Include Offline](#) [Submit for Approval](#) [Manage External Account](#) ▾

Resource Actions

Assigned Territories

[Assign Territories](#)

Action	Territory	Territory Model	Model State	Assignment Date	Method
Other		Dave's Territory Model	Active	9/29/2021 5:22 PM	Territory Assignment Rule

Users in Assigned Territories

No records to display

Related Resources

Contacts

[New Contact](#) [Merge Contacts](#)

[Contacts Help](#) ?

No records to display

Let's start with user-interface metaphors



...and Lightning

Resource Owner (Endpoint)

Resource Identification

`https://david-ktema-dev-ed.lightning.force.com/lightning/r/Account/0011R00002jqGdBQAU/view`

Related Resources

Resource Actions

Resource Content

The screenshot shows the Salesforce Lightning interface for an Account record. The account name is 'Dewey, Cheatham, and Howe' and the account owner is 'David Reed'. The page is annotated with red boxes and labels:

- Resource Owner (Endpoint):** A red box highlights the domain part of the URL: `david-ktema-dev-ed.lightning.force.com`.
- Resource Identification:** A red box highlights the record ID part of the URL: `lightning/r/Account/0011R00002jqGdBQAU/view`.
- Resource Actions:** A red box highlights the action menu at the top right of the record header, containing '+ Follow', 'New Contact', 'New Case', and 'New Note'.
- Related Resources:** A red box highlights the 'Related' tab in the left-hand navigation pane.
- Resource Content:** A red box highlights the main content area of the 'Details' tab, which displays various fields for the account, such as Name, Parent Account, Billing Address, and Shipping Address.

Type	Phone	Website	Account Owner	Account Site	Industry
			David Reed		

Field	Value
Account Name	Dewey, Cheatham, and Howe
Parent Account	
Hourly Billing Rate	
External Id	
URL	conf
T	0031
Test	
Billing Address	Cartalk Plaza United States
Shipping Address	93324 Best Trail Apt. 275 Harryborough, Wyoming ... United States
Created By	David Reed, 9/29/2021 5:22 PM
Last Modified By	David Reed, 7/17/2022 12:14 PM
Account Record Type	Client
Description	

What is an API anyway?

“Application programming interface” isn’t very helpful!

“An API is equivalent to a user interface, except it’s designed for software instead of humans. This is why APIs are often described in the media as technology that allows applications to talk to one another.” (Trailhead: API Basics)



Salesforce has many APIs

.... even more than in this Trailhead module!



Get to Know the Salesforce Platform APIs ~15 mins	✓
Use REST API ~20 mins	✓
Use SOAP API ~30 mins	✓
Use Bulk API 2.0 ~25 mins	✓
Use Streaming API ~30 mins	✓





An API example

Structured, machine-readable messages instead of visual abstractions

PATCH

`https://ceres.my.salesforce.com`

`/services/data/v55.0/subjects/Account/OPA_Id__c/11999`

Authorization: Bearer <token>

Content-Type: application/json

```
{  
  "Name": "Ceres Docks Foundation",  
  "Type": "Legitimate Salvage"  
}
```

Breaking down the API call

Resource (Endpoint, URL)

- Identifies the data or capability with which you want to interact

Action (HTTP method)

- Identifies the genre of action you want to execute
- **GET** (retrieve data), **POST** (create data, run operation), **PUT** (create or replace data), **PATCH** (update data), **DELETE** (delete data)

Body content

- With **POST/PUT/PATCH**, supplies the data you want to modify.
- With **POST**, provides parameters for the operation you're requesting

• **Headers** include authentication and other metadata





Annotated API request example

Structured, machine-readable messages instead of visual abstractions

PATCH HTTP Method

`https://ceres.my.salesforce.com` Endpoint (Resource)
`/services/data/v55.0/subjects/Account/OPA_Id__c/11999`

Authorization: Bearer <token> Headers
Content-Type: application/json

```
{  
  "Name": "Ceres Docks Foundation",  
  "Type": "Legitimate Salvage"  
}
```

 Body Content (JSON)

Interpreting results



GET

`https://ceres.my.salesforce.com`

`/services/data/v55.0/subjects/Account/0011R00002jqGdB`

200 OK

```
{
  "attributes": {
    "type": "Account",
    "url": "/services/data/v55.0/subjects/Account/0011R00002jqGdB"
  },
  "Name": "Ceres Docks Foundation",
  "Type": "Legitimate Salvage"
}
```



Interpreting results



GET

`https://ceres.my.salesforce.com`

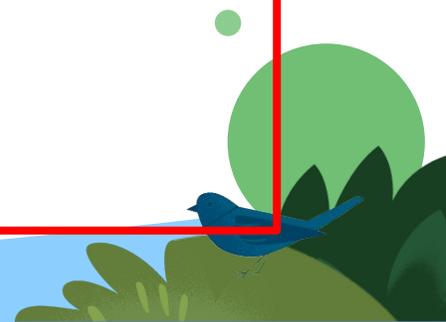
`/services/data/v55.0/subjects/Account/0011R00002jqGdB`

200 OK

HTTP status code (<300 is good)

```
{
  "attributes": {
    "type": "Account",
    "url": "/services/data/v55.0/subjects/Account/0011R00002jqGdB"
  },
  "Id": "0011R00002jqGdB",
  "Name": "Ceres Docks Foundation",
  "Type": "Legitimate Salvage"
}
```

Response body (JSON)



Interpreting results

Different resources and methods have different inputs and outputs

POST

`https://ceres.my.salesforce.com`

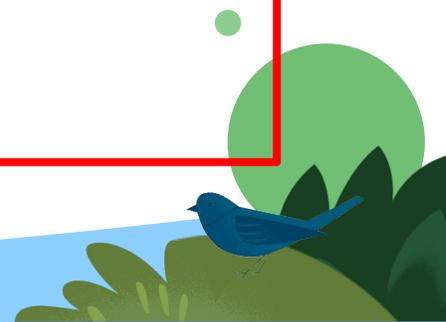
`/services/data/v55.0/subjects/Account/`

201 Created

HTTP status code (<300 is good)

```
{
  "id": "0011R00002viMiVQAU",
  "success": true,
  "errors": []
}
```

Response body (JSON)



Parsing some jargon

Not comprehensive! We'll leave SOAP aside.

HTTP (“Hypertext Transfer Protocol”)

The basic building blocks of API messages. Includes **headers** (metadata sent with request), a **method** (the type of action request), an **endpoint** (identifying the action target), and a **body** (details of the request)

REST (“Representational State Transfer”)

Conventions for designing and organizing *stateless* APIs: every request contains its complete context and content.

JSON (“JavaScript Object Notation”)

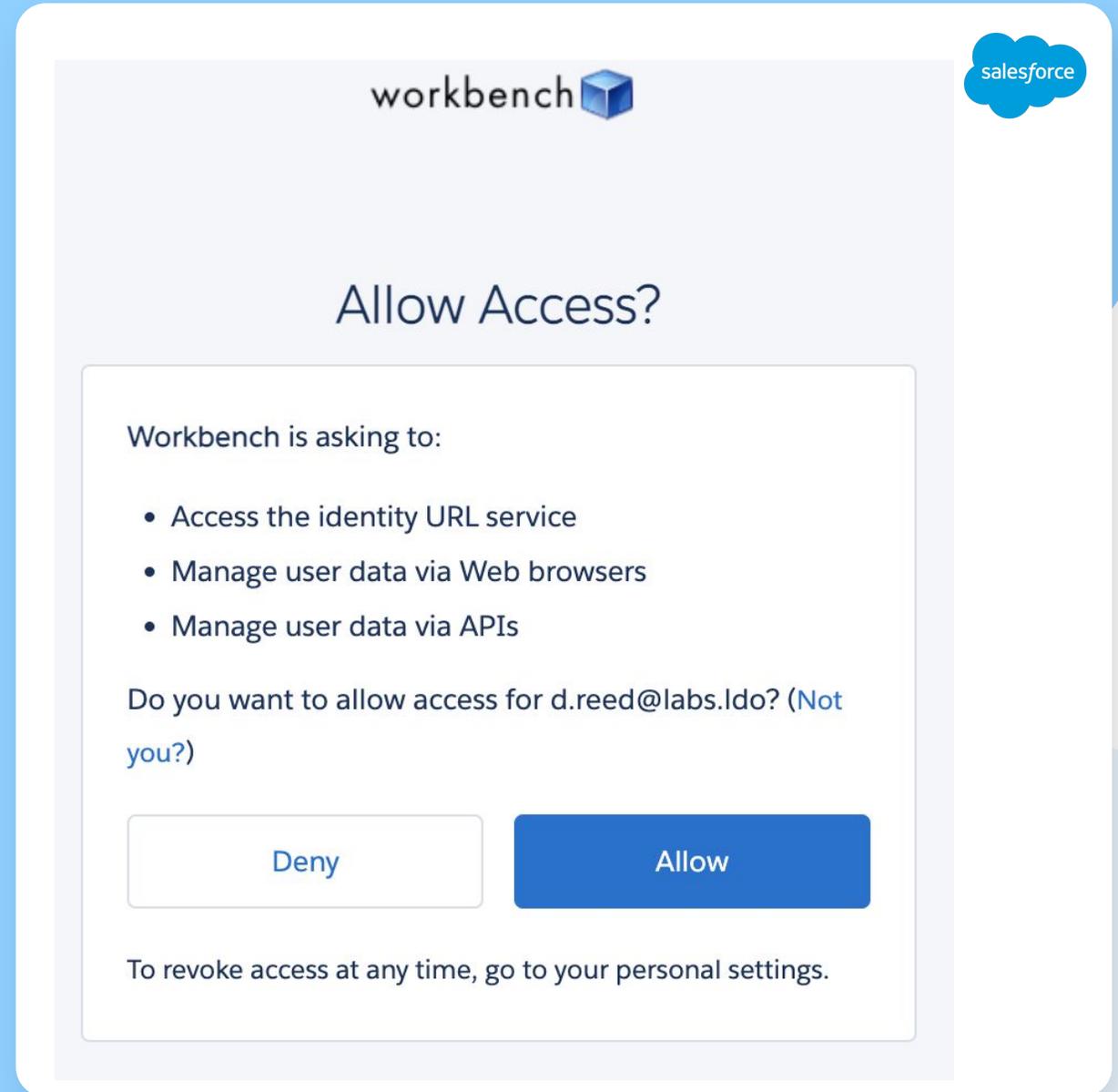
Lightweight, machine-readable data serialization format.

Authentication

With the REST API, authentication is by **access token**.

We get an access token by performing an **OAuth flow** to gain access to the org.

OAuth gives you control over access, and makes sure you never store or disclose your credentials.



The screenshot shows a 'workbench' interface with the Salesforce logo in the top right corner. The main heading is 'Allow Access?'. Below this, a white box contains the text 'Workbench is asking to:' followed by a bulleted list of permissions: 'Access the identity URL service', 'Manage user data via Web browsers', and 'Manage user data via APIs'. Below the list is the question 'Do you want to allow access for d.reed@labs.ldo? (Not you?)'. At the bottom of the white box are two buttons: 'Deny' (light blue) and 'Allow' (dark blue). Below the white box, there is a note: 'To revoke access at any time, go to your personal settings.'



API Versioning

Stability in a changing platform

```
/services/data/v55.0/subjects/Account/
```

The Salesforce APIs are versioned – here, v55.0 is the latest API version

When you use an API version, you get consistent behavior even as the platform evolves.

Opt in to new capabilities by updating the API version.

Not all APIs are versioned! This is a Salesforce architectural decision.





Tangent: API Retirement for Admins

API versioning doesn't last forever!

Components at Risk

- External API integrations
 - Managed, maintained solutions are unlikely to be affected; ask your vendor.
- Code components *that make API calls to your org*
 - Apex with callouts
 - Visualforce with JavaScript API calls
 - Lightning comps w/JavaScript API calls

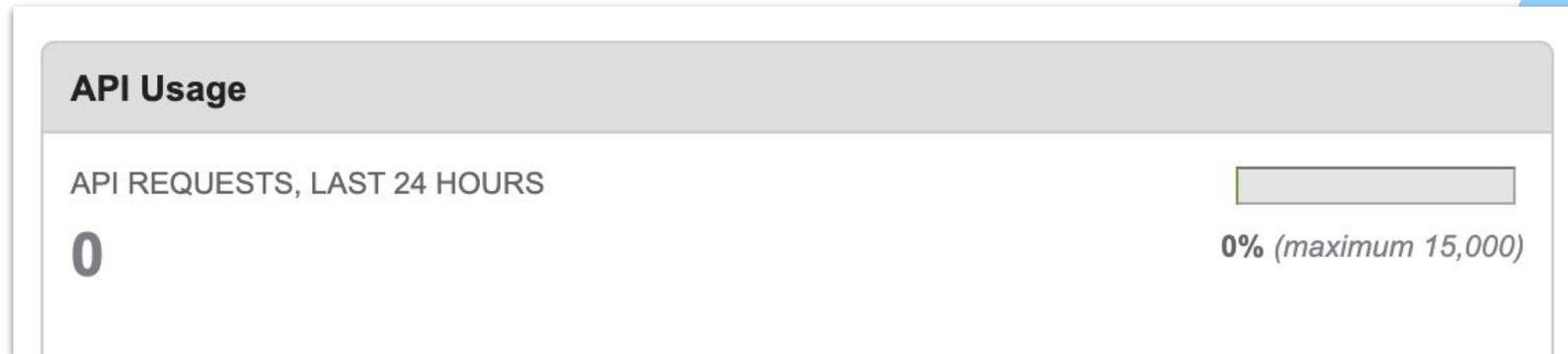
Components Not at Risk

- Declarative automation
- Apex classes or Visualforce pages assigned to old API versions
- Apex, Visualforce, and Lightning components that do not make API calls to the org

Review with your developers or consultants!

APIs have limits

(This org is doing well!)



API call limit is on inbound API calls.

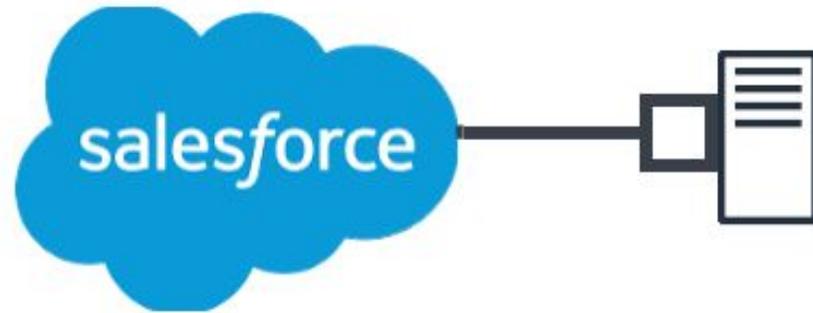
Rolling 24-hour window

Outbound API calls *do not* count

Poorly-designed integrations can be expensive

API use cases

Your org makes API calls to another system



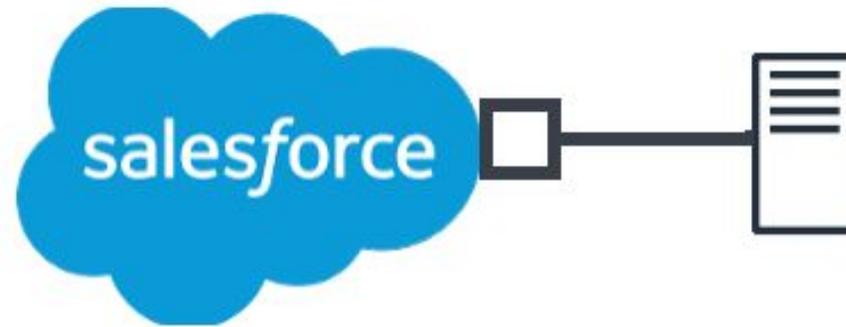
“Some data changed!” (**POST**: updating another system)

“What’s the latest exchange rate?” (**GET**: retrieve information from elsewhere to support a business process in Salesforce)



API use cases

Another system makes API calls to your org



“Some data changed!” (**POST**: updating Salesforce from another system)

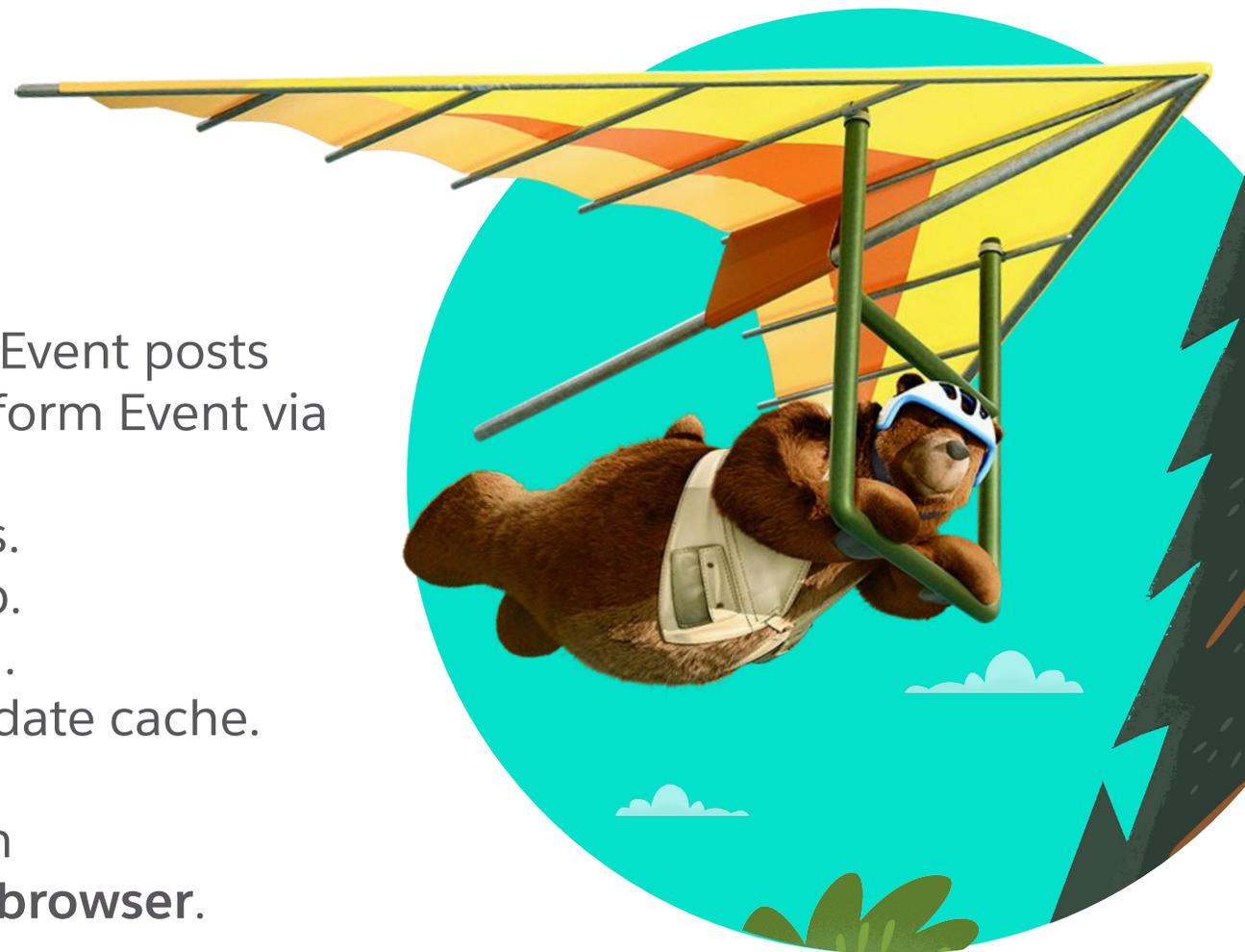
“What are the gifts received today?” (**GET**: retrieve information from Salesforce to support a business process elsewhere)



Flow is awesome, right?

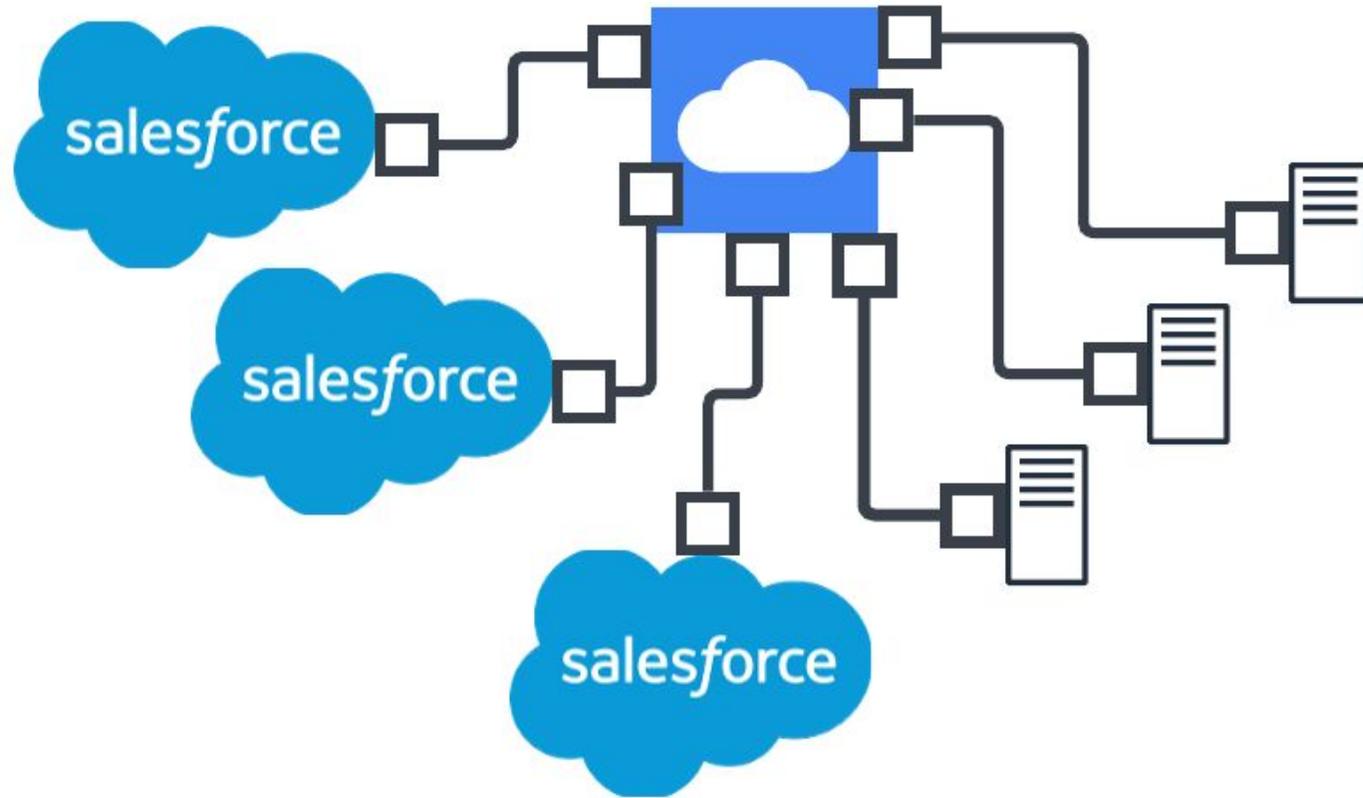
What if you could Flow-orchestrate business processes *outside* your org?

1. Major Gift created in **Salesforce**; Platform Event posts
2. **Enterprise Service Bus (ESB)** receives Platform Event via PubSub API from **Salesforce**.
3. **ESB** calls **AccountingSys** to update ledgers.
4. **ESB** calls **DataWarehouse** to update rollup.
5. **ESB** calls **Salesforce** to sync the latest data.
6. **DataWarehouse** sends **ESB** event to invalidate cache.
7. **ESB** asks **WebServer** to clear cache.
8. **WebServer** pulls latest donation total from **DataWarehouse** and serves to **end user's browser**.



API use cases

Many complex patterns as systems grow



If APIs are so powerful, why isn't everything easy?

It is and it isn't.

APIs make it possible to compose *loosely-coupled* systems.

In a universe of N systems, every system can't know about every other system's API.

Patterns like ESBs, connectors (Zapier), and point-to-point integrations mediate these connections.

Getting Hands-On with APIs

Script your org to solve business
problems





Exploring with Postman

Let's get hands-on and do some work

Create an Account

Retrieve the Account

Create a Contact associated with the Account

Update the Account's Name

Delete the Contact

More: Bulkification, composite calls



What if we did all of that in one request?

Composite API unlocks another layer of orchestration capability

Create an Account

Query the Account

Create a Contact associated with the Account

Update the Account's Name

Delete the Contact



One request with all-or-none option,
data **flows** from result to request,
one unified return value,
no code!

“This doesn’t seem that useful”

You’re not wrong

APIs are *generally* not day-to-day tools for admins. Admins use tools built with APIs, like Data Loader, Zapier, Workbench, and more.

APIs can provide extra capabilities to solve edge cases

Where things get really exciting is when you *orchestrate* APIs to solve business problems.





Consume APIs with clicks or code

Flows, External Services, Zapier, Workato.... or Python!

Many tools, on and off platform, let you knit together your Salesforce org with other systems using APIs. These can be great options for admins to solve business problems!

I got into Salesforce in 2015 as a nonprofit admin. I had no budget for commercial tools. I wrote Python scripts to process, clean, import, and enrich my org's data, via API.

Here's a taste – sharing my passion for this approach.



Under the hood with Python and APIs

Let's do this live!

I've got an NPSP org with some data in it.

Let's think through a business process
together and solve it with APIs and Python.

A little knowledge is a dangerous thing

How not to break your org

You can very quickly cause damage that is hard to cause in the UI.

Testing is even more critical.

Backup, validate, dry run.

Limits still apply - although some are *different*.

- 50k SOQL rows limit *does not apply!*
- There *is* a concurrent long-running request limit.
- Queries can time out.





Remember the user-interface metaphor?

It's API-based too.

Resource Owner (Endpoint)

Resource Identification

`https://david-ktema-dev-ed.lightning.force.com/lightning/r/Account/0011R00002jqGdBQAU/view`

The screenshot shows the Salesforce user interface for an Account record. The account name is 'Dewey, Cheatham, and Howe' and the account owner is 'David Reed'. The page is annotated with red boxes and labels:

- Resource Owner (Endpoint):** A red box highlights the URL path `/david-ktema-dev-ed.lightning.force.com`.
- Resource Identification:** A red box highlights the URL path `/lightning/r/Account/0011R00002jqGdBQAU/view`.
- Resource Actions:** A red box highlights the action buttons: '+ Follow', 'New Contact', 'New Case', and 'New Note'.
- Related Resources:** A red box highlights the 'Related' tab in the 'Details' section.
- Resource Content:** A red box highlights the main details section of the account record.

Related Resources

Resource Content

Resource Actions

UI API, REST API, and more

Your browser is an integrated system too!

User Interface API Resources

This chapter contains User Interface API resources (also called *endpoints*).

- **Records**

Use these resources to get metadata and data about objects and records. These resources make it easy to build Salesforce UI.

- **Actions**

Use these resources to get data and metadata about actions displayed in the Salesforce UI. Examples include standard and custom buttons, quick actions, and productivity actions.

Takeaways



APIs are everywhere!

... but not everything is available as an API

APIs and API-based products can help solve complex business problems – in addition to, or as part of, declarative automation.

APIs have their own limits, risks, and best practices.

API awareness can help admins understand their systems, master new tools, and get under the hood!



Resources



Trailhead

- [API Basics](#)
- [Platform API Basics](#)
- [Postman API Client](#)
- [Quick Start: Connect Postman to Salesforce](#)

API Documentation

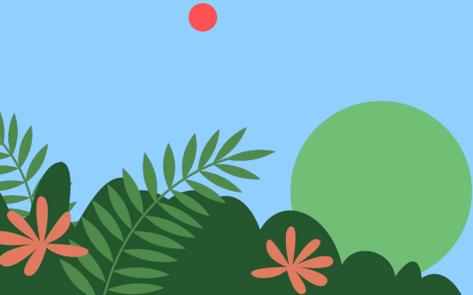
- [REST API Developer Guide](#)

API Resources

- [Postman Request Collection](#)

Plug: Automate Your Org with Cumulus Suite and No Code

“Build Applications with CumulusCI” on Trailhead





Thank You