

NATIONAL ELECTRICAL MANUFACTURERS REPRESENTATIVES ASSOCIATION

Exchanging Data Using APIs

Brief Technology Introduction



What is an API and how is it used?

An API, in its simplest form, is a way for applications on your computer, phone, even light fixtures, to quickly and easily exchange information. They are everywhere, used by almost every modern service or application to communicate with others. Most users probably are making hundreds or thousands of interactions with an API every day using their iPhone or Android device. For instance, you receive a notification on your phone that the Houston Texans have once again defeated the Tennessee Titans. That notification came from the ESPN app on your phone which made a quick connection to their services to see if there was any news. At the same time, it checked with a 3rd party service to provide you with an ad relevant to your interests.

You decide to let your friends know about this so you click the share button and post the link and some trash talk to Twitter.

All of this happened with API interactions, without the user having to know anything about the technology. All the user had to do was install the app and enter the login info. It was seamless to them and so simple that their 2 year old can do it. It is simple, pervasive and it just works. As a result, the user expects any system to be able to communicate with other systems. When they run into something that can't communicate like this it seems broken.

In the developer world, eCommerce sites lose sales every day because people close the web page if they have to wait 3 seconds for an order to process, or a page to load. What happens if the counter salesperson tells a customer it will take a couple of hours to let them know if something is in stock or not?

Why can't I just fax you our inventory?

It wasn't that long ago that manufacturers faxed their current inventory to salespeople every day. Someone would enter that info into an old POS terminal that connected to all of their stores so everyone knew what was in stock. The fax was replaced by email, then downloading Excel spreadsheets. It was tedious, prone to error and forgetfulness, and usually only updated once a day or week so it wasn't very accurate. Most businesses had systems delays throughout the day so they could process these updates. Users grew to accept these delays and might know the "server is down for processing", or something similar but for the most part they were oblivious to the struggle on the backend.

With the advent of the Internet, IT Departments and developers started working on ways to make the backend process easier. They came up with multiple ways to do it, but everyone had a different way of doing things. So the backend processes we used to gather all the information, convert it into a format we could use, and get it into systems in an automated fashion seemed to be held together with duct tape and baling wire. Any small change or error could make the whole thing fall apart, or result in a cascade of errors.

We needed something simple, reliable, easy to implement, and secure. It needed to work on everything and not take a ton of processing power or bandwidth. It needed to be something we could easily automate. Currently, the best solution is the API.



So, what is sent by an API?

When your Twitter app sent the tweet it sent something like the following :



The above image is decrypted and formatted in a way that is easy for most people to read. The actual data that is sent between the two systems is encrypted and in a compressed format. The communication includes what is called a Token, or API Key, which is an encrypted version of your username and password. The data sent is encrypted before it is sent. The data itself is in a simple format called JSON, something most databases and applications can already understand and is fairly easy for the average person to understand.

There is a lot more to the process than this, but most of that happens automatically. The user will never interact with the API directly and your IT team will most likely not have to deal with that communication directly.

Many of you may already have your data available in an API?

At SLG Lighting we get some of the components we use to manufacture our lights from a small local warehouse for a manufacturer in China. Because they are so close, we simply pick up those components as needed instead of stocking thousands of them. But, we don't want to lose sales because our customers think we don't have those items in stock. So, we needed an easy way to add their inventory to our online inventory.

When we checked with the manufacturer we were told there was no way. They didn't have an IT team, didn't want to spend a lot of money getting that process setup, and preferred to keep track of their US inventory using Smartsheet. Smartsheet is an online spreadsheet, similar to Excel. When we looked into it we found that Smartsheet already had an API. Any spreadsheet could be made available via their API without any additional setup or cost. All we had to do was create a free account and ask them to share a spreadsheet with us.

They created a new spreadsheet, used the Smartsheet utility to automatically copy the info they wanted us to access from their main sheet to that new sheet. Then they shared it with us. Now we pull their current inventory every 15 minutes and add those components to our inventory that we present to the Reps. Since

we started doing this we've added two other manufacturers using a similar process. Because Smartsheet, Google Sheets, and Excel Online all offer similar APIs anyone could do the same with any manufacturer using those services.

Most of those are going to be simple API, not much different from sending the fax of all your inventory. Services like TradeTechSolutions can still use the data to provide inventory updates and if this is all the manufacturer can provide they everyone will probably be happy to use it. But, a complex API where you can request specific information, upload orders, and similar would be more useful.

Most ERPs have a complex API built-in. If you are using Netsuite, Microsoft Business Central, SAP, Zoho Inventory, or Odoo it already has an API. Customers can submit orders, request order updates, tracking numbers, part pricing, part availability, and similar using the API from these services. If you use one of these ERPs your users can, most likely, begin accessing the data without you having to do more than grant them access to the info they need.

If you have a simple API provided by Smartsheet, Excel, Google Sheets, or similar and you want to make that a complex API, you have a few options.

- 1. Create your own API
- 2. Use a 3rd party API Service that links to your spreadsheet

Neither of these options is relatively difficult or expensive. A sysadmin or developer with experience should be able to get an API up and running in a day using the common Node-Express-Sequelize build, the Python-based FastAPI, or a similar open source solution. You would need to pay a small amount per month for hosting the API, around \$20 in most cases, but that is the only charge you would incur to run your own API.

If you don't have an IT Department with a developer on staff you can always use a3rd party service like <u>NoCodeAPI</u>. For about \$6 a month you can link to the data in your online spreadsheet, or dozens of other online data storage mediums, and present it as a complex API without ever writing a single line of code. Similar services are available from <u>Google's AppSheet, Zabbix</u>, <u>Appy Pie</u>, and similar services.

Login



If your data is already online you can easily make it available to your customers as an API quickly, cheaply, and securely.

How secure is my data?

If you are using one of the ERP or spreadsheets mentioned your data is as secure as they can make it. Typically the API is not a direct entry into your database. It is a process running on another system. It has various checks and validations that have to be completed to verify who you are, what information you are asking for, and making the request but it does not have the data itself. Usually the data it accesses is "gapped" from the main source. Your main system sends data to the API database on a schedule, and sometimes picks up info as well - a one way communication initiated by the main system. The API database has no way to communicate back to the main database, or make changes, it can only hand it info when it asks for it. Since there is no direct connection between the API and your data, even if your API was hacked it can only access that small subset of your data you stored on the API database.



For More Information

ERP API

<u>Netsuite</u>

https://www.netsuite.com/portal/resource/articles/erp/netsuite-expands-integrationcapabilities-with-new-rest-web-services.shtml

Microsoft Business Central -

https://docs.microsoft.com/en-us/graph/dynamics-business-central-concept-overview

SAP One

https://blogs.sap.com/2020/05/04/rest-api-for-sap-business-one-using-sap-b1integration-framework-b1if/

Zoho Inventory https://www.zoho.com/inventory/api/v1/

<u>Odoo</u>

https://www.odoo.com/documentation/15.0/developer/misc/api/odoo.html

Online Spreadsheet API

Excel Online https://docs.microsoft.com/en-us/sharepoint/dev/general-development/excelservices-rest-api Google Sheets https://developers.google.com/sheets/api/reference/rest

Smartsheet

https://www.smartsheet.com/content-center/best-practices/tips-tricks/api-getting-

started

No Code API Solutions

NoCodeAPI https://nocodeapi.com/

Google Appsheet https://www.appsheet.com/

<u>Appsmith</u> <u>https://www.appsmith.com/</u>

<u>Xano</u>

https://www.xano.com/



NATIONAL ELECTRICAL MANUFACTURERS REPRESENTATIVES ASSOCIATION