3.2.1 What are APIs and why do they matter?

Many people in the tech world have a vague idea about what an application programming interface (API) is. Put simply, an API is a software go-between that allows two applications to communicate with each other. Think about it: nearly all of the daily tasks you perform online, whether on your phone, tablet or laptop, rely on sending and receiving information between different sites or services. These requests and responses are all possible thanks to APIs.

The following video briefly explains how APIs work:

00:35

COS60016: How do APIs work? (2021) created by Swinburne Online.

Transcript (https://swinburneonline.instructure.com/courses/5055/pages/transcript-how-do-apis-work)

Every time you open Facebook or scroll through Instagram, you are calling the app’s API. In other words, you are sending a request to the app’s API and getting a response back in the form of your newsfeed. This is facilitated through an interface (the I in API), which is something that an entity uses to interact with another system.

APIs act as gateways between service providers and consumers. Consumers gain access through various interfaces depending on the consumers' credentials. In turn, consumers can use the services they receive as they choose.

> ### API waiter at your service
>
> A common analogy used to describe APIs is of a waiter at a restaurant. You, the customer or user, ultimately want food from the kitchen, or web server, but you're sitting at a table with a menu. To communicate your order to the kitchen and to bring your order, you need a link. It can't be the chef because the chef is cooking in the kitchen. So, you use a waiter, or API, as the go-between—the waiter takes the order, tells the kitchen what to prepare, and delivers the food, or response, back to you.

When should you create an API?

While APIs are powerful tools, they are not suitable in every instance. So how do you decide whether to create one?

Use the following questions to guide your decision: if you answer 'yes' to most of them, then an API is a good idea (Smyth 2018, Jones 2017):

- Will several applications or sources consume your API?
- Do you have too much data to use a data dump or other transfer method?
- Does the data change regularly, and is data recency important for your users' purpose?
- Will your users only need access to segments of your data set at a single time?
- Will your users need to do more than retrieving data? That is, do they need to update, contribute to, or delete data?

APIs in practice

Some organisations provide ‘open’ access to their APIs, allowing anyone to connect and request information, including Uber and Slack. An example of this is the Twitter API that is available to anyone who has an account with them. Using this API, it is possible to write code in languages like JavaScript or Python to collect Tweets and a whole lot of other operations. This is incredibly valuable because this data can then be analysed to inform things like sentiment analysis to understand how a population of users feels about a particular topic.

Sentiment analysis is remarkably valuable for large companies, who can monitor how customers respond to new products or marketing campaigns and mitigate the reputational damage that poor sentiment can cause. It can also be used to monitor public sentiment about political figures. Academics demonstrated a relationship between Twitter sentiment analysis and voting trends (O’Connor et al. 2010).

Further reading

This resource provides a comparison of various politician's Twitter behaviour using sentiment analysis.

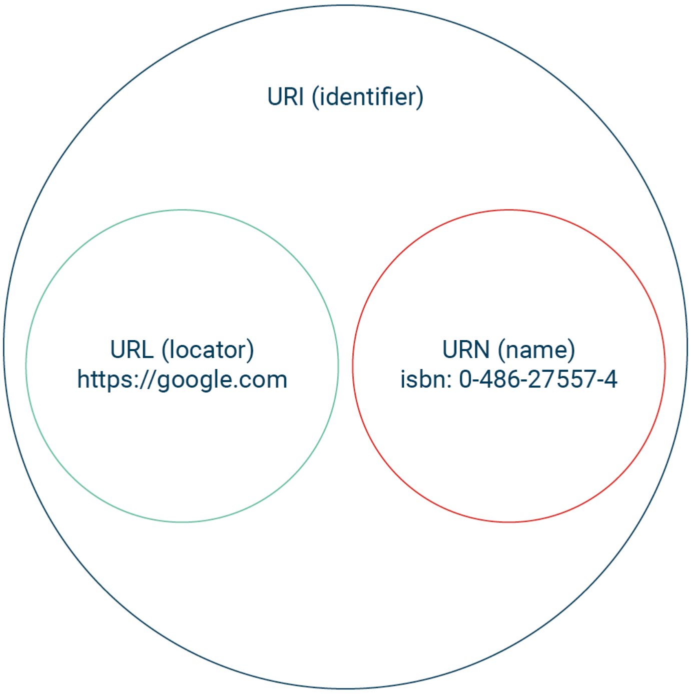
- **Understanding Political Twitter (https://towardsdatascience.com/understanding-political-twitter-ce3476a38377)** (Grubbs & Mandi 2020)

API endpoints and URLs

Essentially, an API endpoint is the point at which it connects with another service or server, such as via a Uniform Resource Locator (URL). When two systems—in this instance, an API and a server—interact, the communication entry point of that communication channel is called the API endpoint: it functions as a doorway through which the server is able to interact with the API. If the endpoint URL changes, linked applications will no longer function, as they will no longer be able to access it.

You are almost certainly familiar with the term URL, even if you don't know what the acronym stands for. You use URLs every time you visit a webpage to check your email or access this unit. URL stands for 'Uniform Resource Locator', and it functions as the network location of a resource. It is a subtype of a larger category called Uniform Resource Identifiers (URIs). A URI is an array of characters used to identify a web resource.

APIs' endpoints are usually URLs in public APIs, but they have to be URIs. Another type of URI is a Unique Resource Name (URN), an example of which is the ISBN numbers that identify books.



COS60016: URLs, URNs and URIs (2021) created by Swinburne Online.

Read: You can read more about the distinction between URIs and URLs in **What is the Difference between a URI and a URL?** (https://rapidapi.com/blog/url-vs-url/) (RapidAPI 2020).

Reflection

Can you think of other instances where you might use an API every day? Why do you think a vendor, like Twitter, might provide 'open' access to their APIs?