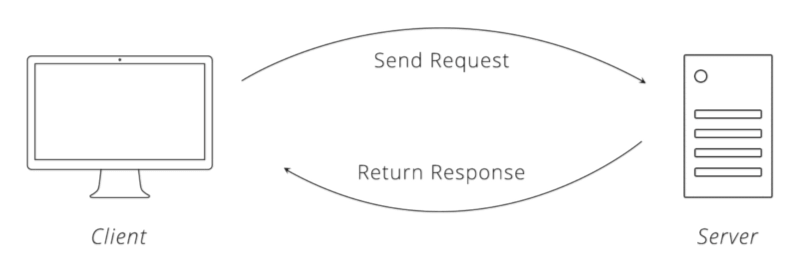
**What Is an API?**

**How Do APIs Work?**

So...what does an API have to do with scrolling my Facebook feed?

Well, to answer that question, you have to know a little about how APIs work. Somewhere in the world, Facebook servers have access to all the data you want to see, and for you to see it, your **app** (called a **client**) needs to have a conversation with the server. An **API**allows these different application components to communicate messages to each other using **requests and responses.**

A typical conversation between client and server

This looks something like:

* Client: “Hey server, can I get the most recent posts for all of Rebecca’s friends?" 🤔
* Server: “Here you go, all the most recent posts from Rebecca’s friends!" 🎉

Or the server could reply something like:

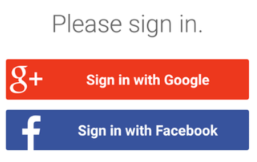
* Server: “Sorry, I actually can’t find Rebecca’s friends because her Wi-Fi isn’t working right now." 🙁

This is what’s called **communication between client and server:** The client makes a request for data, and the server responds with data. To put this into a real-life context, you can think of an API as a waiter in a restaurant.

You want dinner, so you ask the waiter for a slice of pizza. 🍕The waiter then sends that message to the kitchen, which is a little like the server. The kitchen takes your order and cooks your delicious meal, 🍳 and then the waiter returns it back to you (the client), so you can enjoy your food. 🙌 An API works the same way, except instead of a client requesting a food item, it requests some type of data, which is then returned by the server.

Beyond just working between clients and servers, APIs can communicate between many different **components** of your application as well as components of your application and other **developers.** They do this by acting as a go-between that delivers messages through requests and responses. APIs can communicate between server to server, server to developers, or client to server.

You've probably already seen one example of how an API is used to communicate between servers and developers. You know how you can use your Google or Facebook accounts to log into a website? It's because they have built APIs that other developers can use in their own websites to take care of the users for them!

APIs make these logins possible!

APIs also create **standardized** and **reusable** methods that allow developers to access specific data when building applications. To relate this back to the waiter example, you can also think of an API as a type of menu for all the available options of accessible data. When you eat out, the restaurant menu shows a specific amount of predetermined options. This makes it easier for you to get food quicker and the chef to understand what you want better.

The same is true of APIs. You can create or consume a set functionality within your API - like that log in through Google/Facebook feature!  This makes it easier to request the same type of data multiple times and to know what kind of data is available to use.

**How Can I Use APIs as a Developer?**

There are two main ways developers use APIs in their applications.

**Private**

**Private** APIs make sure that people outside your organization or application don’t have access to the data available from the server. For example, if Facebook developers wanted to build an **internal** application for HR employees to manage and analyze hiring information, there would be a lot of data the employees would want to view, create, and edit. In order for the users to interact with the data in this way, Facebook developers could build an API so the HR employees could access the hiring data through their application.

If this application is being used internally, why can't developers just directly query the database to get this data?

An API can be used as a **layer** between the database and the person who wants to access or manipulate the data. Even with developers as intelligent as the ones at Facebook, a direct and uncontrolled database query could cause chaos! 🤯 What if someone accidentally deletes an important table or edits something they shouldn't have? A database is a bedrock for all the data in an application, so you wouldn't want it to be easily accessible or manipulated by just anyone. An API allows for an extra level of abstraction to better manage the data being accessed or edited.

A **private** API enables *only* the users within a certain organization, in this case, Facebook, to be able to use the API that can access the database.

**Public**

These are APIs that others can use. They allow developers to get data from another application to enhance their own projects. There are many ways you can use data from third-party APIs, but here are a few:

1. Let’s say you want to build a travel website that presents the weather of the location you want to travel to. Instead of collecting your own weather data, you can use a [Weather API](https://openweathermap.org/api) and get the data from there! 🌤
2. If you’re a musician looking to build a website so your fans can listen to your music, instead of building your own music streaming player, you can use the [Spotify API](https://developer.spotify.com/documentation/web-api/" \t "_blank)and play your music directly on your website! 🎵
3. You want to build a fan page for your favorite TV show (Game of Thrones of course), by aggregating all the different cast’s Instagram accounts into one website - guess what, there’s an [Instagram API](https://developers.facebook.com/docs/instagram-api/" \t "_blank)that can help you do that! 🌅

There are also some APIs that are somewhere in the middle of public and private. This can happen when different API **requests** are only possible, depending on the level of **access**you have. 🔐We'll talk about this later when we cover authentication.

There are thousands of public APIs developers can use in different ways to enhance their projects. Here’s a list of those [**publicly available APIs**](https://github.com/toddmotto/public-apis) you can use!

**Let's Recap!**

* APIs are ways to communicate data between different components of your application **and** between your application and other developers using **requests**and **responses.**
* APIs provide a means to access data in a reusable and standardized way.
* Developers can use public and private APIs.