There are multiple API method and styles I have setup and used in the past.

Webhooks – is a trigger event response usually server based

We use this at work as a trigger when an invoice or purchase order was generated to automatically send it to the customer or vendor. We also did this for bank reconciliation where one would compare the checks being paid against the accounting systems records. One last place we used this was in paying out taxes to the state as we did not pay taxes on equipment in our processing plant an if we paid our own tax we received a reduction from the state by 2% which ends up being hundreds of thousands of dollars in a year. And millions for the three years we were allowed to go back and claim it.

WebSocket is a protocol that allows for full-duplex communication over a persistent connection, making it suitable for real-time applications. Unlike the request-response model of HTTP, WebSocket keeps the connection open, allowing for continuous bidirectional data exchange. It can be used with various programming languages, including C, Python, JavaScript, and more, to facilitate real-time interactions in applications such as live chats, gaming, and live data updates. We uses a completely isolated version of this to control pumps and valves at one of our plants going to a dock. I also used this in an client and server application I created which could send pertinent information back and forth. You might be familiar with many of the messaging services or stock tickers which may use this service format.

There are many others which I have not used lately.

**GraphQL:**

An alternative to REST, GraphQL allows clients to request only the data they need, potentially reducing the amount of data transferred over the network.

**gRPC:**

A high-performance, open-source framework developed by Google. It uses HTTP/2 for transport, Protocol Buffers as the serialization format, and it allows for bidirectional streaming of requests and responses.

**SOAP (Simple Object Access Protocol):**

An older standard for designing APIs. SOAP APIs are characterized by XML-based messages, strict schemas, and operations.

**OData (Open Data Protocol):**

A protocol that allows the creation and consumption of queryable and interoperable RESTful APIs. We used OData for our Inventory cycle counting and work order processing at our plant.

One of the most widely used Models is what we will concentrate on.

**<CHANGE SLIDE**

**HTTP RESTful API’s**

Here are some of the benefits of this format:  
 Communication standard – the connectivity between the server and the clients are regulated through the HTTP process.

Scalability – A large number of clients can use this process because it is HTTP socket based.

Platform Agnostic: Used across different devices and platforms, including some programming languages.

Decoupled Development: since it is single side development being the client side is HTTP standard connectivity, then the developer of the API can easily develop or modify content.

**There are several data formats available:**

JSON or (JavaScript Object Notation) seems to be the most prevalent of all that I have used.

XML or (eXtensible Mark up Language): is a cross between human and machine readable

The two less common formats are HTML which builds an entire functional webpage out of the data, and YAML which is usually for Human Readable outputs

**<CHANGE SLIDE**

Here are examples of each format

JSON looks like its an array format and XML format, very similar to HTML but is not in HTML sections

**<CHANGE SLIDE**

HTML creates an HTML viewable page out of the data while YAML is used mainly for human display and readability.

**<CHANGE SLIDE**

**Authentication Process:**

API Key: A simple alphanumeric string that the client should send to authenticate the request.

Basic Auth: Uses a username and password combination to authenticate.

OAuth: A common authorization protocol. There are two versions: OAuth 1.0a and OAuth 2.0, with OAuth 2.0 being more common. It allows third-party services to exchange user's access credentials for an access token.

Bearer Token: A type of token authentication where the client sends a token with every request to authenticate.

Different API’s have different usage capabilities and restrictions

Some are free, many are not.

Almost all have a limited number of allowed calls or transactions within a set timeframe.

Where are Restfull API’s used?

**<CHANGE SLIDE**

At the plant I use to work at we used this to update plant status to a display,

we also use this on a schedule to update our accounting system based on consumption and creation matrix.

**<CHANGE SLIDE**

I have used a couple of API in my personal coding one is the PokeAPI I built a game using the images from the site loading them into an array so that I would only have to make a single call for each of the sub entity Pokémon. You can see in the array that there are multiple different queries,

**<CHANGE SLIDE**

and here is a snippit of code I used for the calls.

**<CHANGE TO GAME**

This is the result of the HTML and JavaScript using API calls and DOM manipulation.

**<CHANGE SLIDE**

Another is APINinja making use of their weather API specifically, but they have a verity of api’s to choose from.

One can see the JSON which is returned with information for the area, along with the Information being relayed to the DOM.

**<CHANGE SLIDE**

Here is the Java Script I used for this with my redacted API KEY

Another API call is the local cache stprage which we have recently used to retain information in reference to last visit and number of visits.