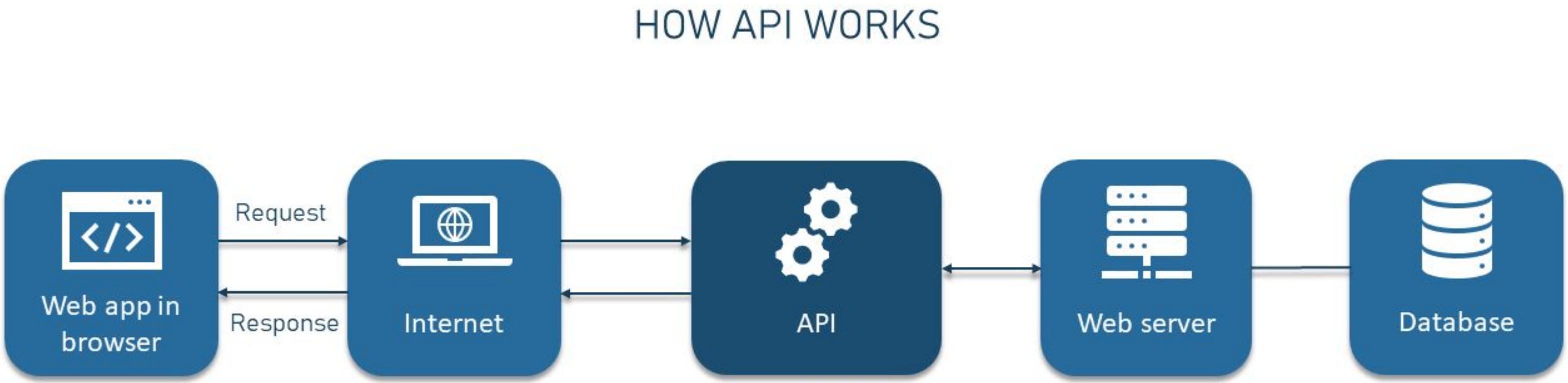
APIs

William Trang

What is an API?

APIs, or application programming interfaces, serve as a bridge that allows different software applications to communicate with each other. APIs enable the exchange of data and functionality between different systems



Why APIs?

- Resistant to front end change
- Ease of use
- More information available
- No ambiguity about whether or not you can access their data

Types of APIs

Private APIs

- Used to connect different software components within a single organization
- Not available for third-party use
- Some applications may include dozens or even hundreds of private APIs

Public APIs

- Provide public access to an organization's data, functionality, or services
- Can be integrated into third-party applications
- Some public APIs are available for free, while others are offered as billable products

Partner APIs

- Enable two or more companies to share data or functionality in order to collaborate
- Not available to the general public
- Leverage authentication mechanisms to restrict access

How do APIs work?

Similar to HTTP requests, APIs typically work on the request/response format

- Request: When you want to access data or services from an application, you send a request to its API
- Response: The API then sends back the requested data or performs the desired action, providing a response

REST APIs

REST, or Representational State Transfer, is a flexible and popular API architecture, and they support data transfer in different formats, including XML, HTML, plain text, JSON, and more. They perform operations using HTTP architecture

HTTP Methods

Common HTTP methods used in APIs are:

- GET: Retrieve data from a specified resource
- POST: Submit data to be processed to a specified resource
- PUT: Update a specified resource
- DELETE: Remove a specified resource

SOAP APIs

SOAP, or Simple Object Access Protocol, is another popular API architecture. It is a secure way to build APIs, and it works by encoding data in the XML format.

Format of SOAP APIs

Unlike REST APIs, which focus on only the request and the response, SOAP APIs have 4 components

- Envelope: Defining the structure of the message
- Encoding: Rules for expressing the type of data
- Requests: How each SOAP API request is structured
- Responses: How each SOAP API response is structured

REST vs SOAP APIs

- SOAP is language, transport, and even platform independent, whereas REST requires the use of HTTP
- SOAP is more secure, meaning it is better used for sensitive data
- SOAP is more complicated, worse performance, more runtime
- SOAP is harder to adapt for specific use cases

Challenges of APIs

- Like front end, APIs can have changing implementations/versions
- Rate limiting/pricing
- Reliability/uptime
- Quality of documentation