API Principles

Learners will be able to...

- **explain the difference between a UI and an API**
- **classify the different types of API**
- **define the features of a Web API**
- **contrast the responsibilities of the client side and the server side**
- **identify the different components of a URI**
- **use a curl client to execute a simple request to an API that returns JSON**

info

Make Sure You Know

This assignment assumes a basic knowledge of JavaScript, HTML and CSS.

Limitations

We will not explain simple programming concepts, it is assumed that the user is familiar with these concepts.

UI versus API

What is a UI?

A **user interface** is the point of interaction between a person and a machine. For the purpose of this discussion we will focus on computers, but all machines have user interfaces. Your toaster oven is an example, it may have a dizzying array of buttons and knobs for its interface.

Common user interfaces are:

1. Menu driven
    - Click on the Help menu at the top of this window.
    - Select “Shortcuts”, this is a list of shortcuts you can use when you edit files later in this course.
    - Close the shortcuts window by clicking on the x in the right corner of the tab.

    shortcutstab with the x in the right corner circled
2. Command line interface
    - In the terminal on the left you can interact with the Linux terminal
    - Type `date` at the prompt and Linux will return the current date and time
3. Graphical User interface (GUI)
    - It’s likely you are using Codio inside a Window on a computer that has a GUI
    - You can use a mouse or trackpad to drag the window around and interact with the interface
4. Touch user interface
    - These are common on phones and notebook computers, you interact directly with the screen to manipulate objects.

What is an API?

An Application Programmer’s Interface (API) provides a way to communicate **programmatically** with a product or service. Through the API a program can access a different program’s data or request a service.

The program making the request through an API does not need to know the details of how the request will be handled, just how to make the request. This is similar to driving a car, driving does not require you to understand

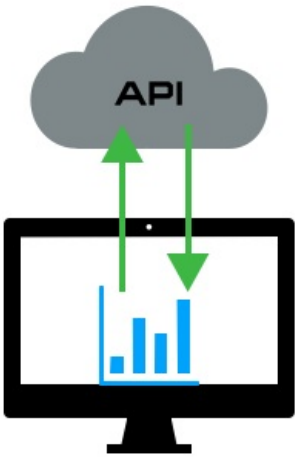
how the engine or the braking systems work, you just need to know which pedal to use.

API examples

- Application APIs, for example Excel, can provide access to the data the application manages. For example, Excel provides a JavaScript API that can be used to access worksheets, ranges, tables, and more.
- Device APIs allows you to interact with the hardware your browser is running on. For example the Geolocation API.
- Software libraries provide an API to access the utilities provided by the library. For example Python has an extensive library.
- Databases provide an API for data requests. For example, the ODBC interface provides a common way to access data in different types of databases.
- Web APIs are used to communicate between computers that are connected over the Internet. There are client-side APIs (for example fetching specific data from a server) and server-side APIs (returning customized search results).

Web APIs

As we mentioned on the previous page, an API is a way for one program to communicate with another. A web API provides a protocol for how a client may make a request for a service or for data from a server using HTTP. A protocol is required so that the requester knows how to formulate the request and the receiver knows how to interpret the request.



Why would a website want to provide an API?

If a website does not provide an API it makes it more difficult to access the information on the website. Another way to access information on a website is via a method called "web-scraping". Pages are downloaded and processed locally to extract information. If you see an email address with the `@` replaced by the word `at` this is often done to prevent the email address from being "scraped" and added to an undesired mailing list. This helps foil algorithms that search for an `@` sign to find email addresses.

- Many social media websites offer APIs. One example is YouTube, you can use their API to upload videos to the YouTube website.
- A credit card company would provide an API so that businesses can access customer info when the customer does not have their card.
- A review site such as Rotten Tomatoes provides an API so that other websites, for example movie theaters, can post review information.

Examples of Web API Protocols

SOAP - Simple Object Access Protocol

SOAP uses the Extensible Markup Language (XML) to allow processes to communicate between different operating systems. It can transport over any protocol such as HTTP, Simple Mail Transfer Protocol (SMTP) and Transmission Control Protocol (TCP). SOAP is often found on legacy systems because it supports multiple protocols.

REST - Representational State Transfer

Representational State Transfer REST. The REST API is often referred to as RESTful API. The REST architecture is stateless, this means each request is independent and not contingent on the completion of other requests. This reduces server load and supports scalability.

JSON-RPC - Remote Procedure Call -

JSON-RPC uses JSON for its data format. Similar to REST, it is stateless and similar to SOAP it is transport agnostic.

gRPC - Remote Procedure Call

gRPC uses HTTP/2 for transport.

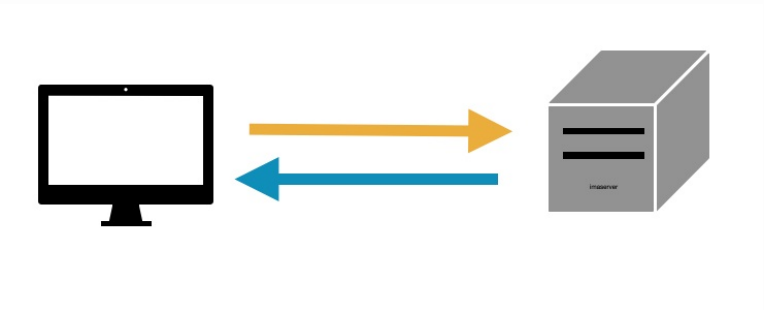
GraphQL

GraphQL is a query language used to describe data requirements and a server-side for interpreting and filling requests.

Client-side versus server-side

The terms **client-side** versus **server-side** refers to where code execution occurs. Input is gathered from users (often using forms) on the client-side and processed on the server-side. The client-side code may do some data verification, such as making sure numbers are entered in numeric fields, and that all required fields have been filled. On the server-side it may store the information or return the information requested in the form.

Client-side programming is more focused on the user experience and the interface and server-side development is more focused on processing data and returning information.



.guides/img/clientserver

important

What is HTTP?

Hypertext Transfer Protocol (HTTP) enables communication between clients and servers. Hypertext refers to the fact that web pages have a special format (HTML), and can contain links, which when clicked will fetch a new web page.

A client makes an **HTTP request** to access a resource on a server. A request contains the following components:
1. A request method that specifies the action requested. For example `GET` or `POST`.
1. The path of the resource to fetch or a destination for a post.
1. The version of the **HTTP protocol** being used - `HTTP/1.0` or `HTTP/1.1`.
1. Optional headers that convey more information.
1. In the case of a `POST` there will be a body that contains the data sent.

The server provides an **HTTP response** that does one of these things:
- provides the client the requested resource
- informs the client that the requested action has been completed
- informs the client that an error occurred when executing the request

The components of a response are:
1. The version of the **HTTP protocol** being used.
1. A status code and explanation pertaining to the success of the request.
1. Optional headers that convey more information.
1. An optional body if data was requested.

Components of a URI

A **Uniform Resource Identifier (URI)** is a sequence of characters that identifies a web resource by name or location or both. A **Uniform Resource Locator (URL)** is a form of URI that identifies a particular web address and how to access that address. A **Universal Resource Name (URN)** is also a form of URI, it identifies a resource by name within a namespace. A URN always has the prefix `urn:`.

Scheme

The first element of a URI is the scheme. It is separated from the rest of the URI by a colon. The scheme defines the format of rest of the URI. The Internet Assigned Number Authority (IANA) maintains a list of official URI schemes. The IANA is responsible for the parts of Internet that need to coordinated in order for it to run smoothly for example, domain names.

Some examples of schemes are:
Name	Example	Purpose
————	————————-	———-
HTTPS	`https://www.irs.gov/`	Secure Hypertext Transfer Protocol
spotify	`spotify:search:<Guided by Voices>`	Access Spotify Directly
view-source	`view-source:https://www.codio.com/`	Shows the source code of any website

Authority

The authority component begins with `//` then there is an optional **user info** component followed by an `@`, a **host** subcomponent, followed by `:` and finally an optional **port** component.

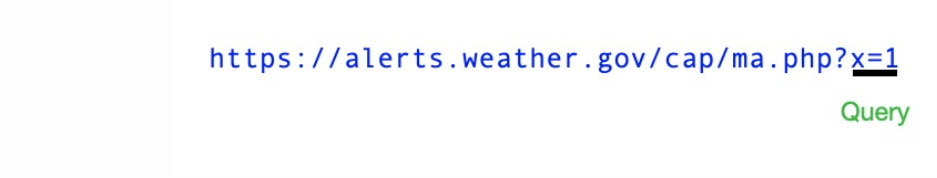
URI Path

The path component can include any number of path segments separated by a slash (/). A path is always defined for a URI but it can be empty. In the weather example below, the URI path is `/cap/ma.php`.

Query (optional)

A query consists of the object to be queried and the query to made on the object separated by a `?`.

In the example below we are querying the National Weather Service public alerts.



A query to the NWS website

Paste this URL into a browser tab - `https://alerts.weather.gov/cap/ma.php?x=1`.

You can change the `ma` to be the initials of the state you are interested in. You can also change the query as listed below.

`x=1` Watches, Warnings or Advisories
`x=2` Zone listing
`x=3` County listing

Fragment (optional)

The fragment portion begins with a # and it can refer to a portion of a web page. If the URI points to an HTML document the fragment might refer to an element on the page and that element will appear in view in the browser.

Here is an example from our docs. If you click on the link below you will be taken to the specific section on the page. Paste the link in a browser tab to try it out, the text to the right of the # refers to the section on the page.

```
https://docs.codio.com/students/accessing-
        codio/faq.html#deleted-a-file-by-mistake
```

Access Web APIs

cURL (Client URL)

cURL is a command line tool, available on many operating systems, that provides a way to transfer data between a client and a server. The command includes an endpoint, which uses URL (Uniform Resource Locator) syntax, to specify where to send the request. The data is returned in JSON format. You can learn more about the cURL command by typing `man curl` at the command prompt.

Many interesting websites can be accessed using the cURL command. Some web pages pull information from various sources and combine it in a new way. We will explore this in greater detail in assignment 1.3 of this module. In the examples below we demonstrate how you can use a map website to search for places by name. Then you can use the returned information to find the coordinates of the location you are interested in and its street address.

Using the coordinates you can search for interesting places nearby. You can also get information about the weather, sunrise and sunset times, local time and timezone using the coordinates.

Try some of the examples below in the terminal window on the left. There are two versions of some of the commands, the first version uses the -i parameter and outputs the HTTP response headers. In the second version we are piping the output to the Linux utility `jq` for readability.

API of OpenStreetMap names

Find the places containing “central park” in their name in the US:

```
curl -i "https://nominatim.openstreetmap.org/search?
q=central+park&countrycodes=us&format=json"
```

```
curl "https://nominatim.openstreetmap.org/search?
q=central+park&countrycodes=us&format=json" | jq
```

The decimal coordinates of Central Park in New York are:
Latitude: `40.7827725`
Longitude: `-73.965355`

The view from the coordinates 40.7827725, -73.965355

Find the address by the coordinates:

```
curl -i "https://nominatim.openstreetmap.org/reverse?
lat=40.7827725&lon=-73.965355&format=jsonv2&namedetails=1"
```

```
curl "https://nominatim.openstreetmap.org/reverse?
lat=40.7827725&lon=-73.965355&format=jsonv2&namedetails=1" | jq
```

API of Wikipedia

Find points of interest near the coordinates:

```
curl "https://en.wikipedia.org/w/api.php?
action=query&list=geosearch&gsradius=1000&gscoord=40.7827725|-73
.965355&format=json&gssort=relevance&gslimit=20" | jq
```

Information from the Spanish wikipedia:

```
curl "https://es.wikipedia.org/w/api.php?
action=query&list=geosearch&gsradius=1000&gscoord=40.7827725|-73
.965355&format=json&gssort=relevance&gslimit=20" | jq
```

API of the US National Weather Service

Weather alerts for a state:

```
curl "https://api.weather.gov/alerts/active?area=NY" | jq
```

challenge

Try this variation:

- Type the cURL command to get the weather for your state.

Request weather data using coordinates:

```
curl "https://api.weather.gov/points/40.7828,-73.9653" | jq
```

Sunset/sunrise API

```
curl "https://api.sunrise-sunset.org/json?
lat=40.782864&lng=-73.965355&date=tomorrow&formatted=0" | jq
```

Time API

Get the current time for the coordinates:

```
curl "https://timeapi.io/api/Time/current/coordinate?
latitude=40.7827725&longitude=-73.965355" | jq
```

Get the current time for a time zone:

```
curl "https://timeapi.io/api/Time/current/zone?
timeZone=Europe/Amsterdam" | jq
```

Get the timezone using coordinates:

```
curl "https://timeapi.io/api/TimeZone/coordinate?
latitude=40.7827725&longitude=-73.965355" | jq
```

Get the name of the day by the date:

```
curl "https://timeapi.io/api/Conversion/DayOfTheWeek/2022-11-22"
| jq
```

challenge

Try this variation:

- Type the cURL command to get the day of the week for your birthday next year.

Convert between two time zones:

```
curl -X 'POST' \
  'https://timeapi.io/api/Conversion/ConvertTimeZone' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "fromTimeZone": "Europe/Amsterdam",
  "dateTime": "2021-03-14 17:45:00",
  "toTimeZone": "America/Los_Angeles",
  "dstAmbiguity": ""
}' | jq
```

challenge

Try this variation:

- Type the cURL command to convert between your time zone and a time zone in Asia, or vice versa.