INFO 2302 Web Technologies

Client-side Web API

Dr. Marini Othman
Department of Information Systems
Kulliyyah of Information and Communication Technology
International Islamic University Malaysia

Introduction to Web API

Application Programming Interfaces (APIs) are constructs made available in programming languages to allow developers to create complex functionality more easily.

API abstracts more complex code away from you, providing some easier syntax to use in its place.

Introduction to Web API

If you want to program some 3D graphics, it is a lot easier to do it using an API written in a higher-level language such as JavaScript or Python, rather than try to directly write low-level code (say C or C++) that directly controls the computer's GPU or other graphics functions.

Categories of client-side web API

Browser API

Built into your web browser and are able to expose data from the browser and surrounding computer environment and do useful complex things with it.

Third party API

Not built into the browser by default, and you generally have to retrieve their code and information from somewhere on the Web.

Other JavaScript tools

JavaScript libraries

Usually one or more JavaScript files containing custom functions that you can attach to your web page to speed up or enable writing common functionality. Examples: jQuery and React.

JavaScript framework

Packages of HTML, CSS, JavaScript, and other technologies you install and then use to write an entire web application from scratch. Eg: Angular and Ember.

The key difference between a library and a framework is "Inversion of Control".

When calling a method from a library, the developer is in control. With a framework, the control is inverted: the framework calls the developer's code.

How do APIs work?

1. They are based on objects

Your code interacts with APIs using one or more JavaScript objects, which serve as containers for the data the API uses (contained in object properties), and the functionality the API makes available (contained in object methods).

2. They have recognizable entry points

When using an API, you should make sure you know where the entry point is for the API.

```
//entry point – a reference to canvas element
const canvas = document.querySelector("canvas");
const ctx = canvas.getContext("2d");
```

How do APIs work?

3. They often use events to handle changes in states

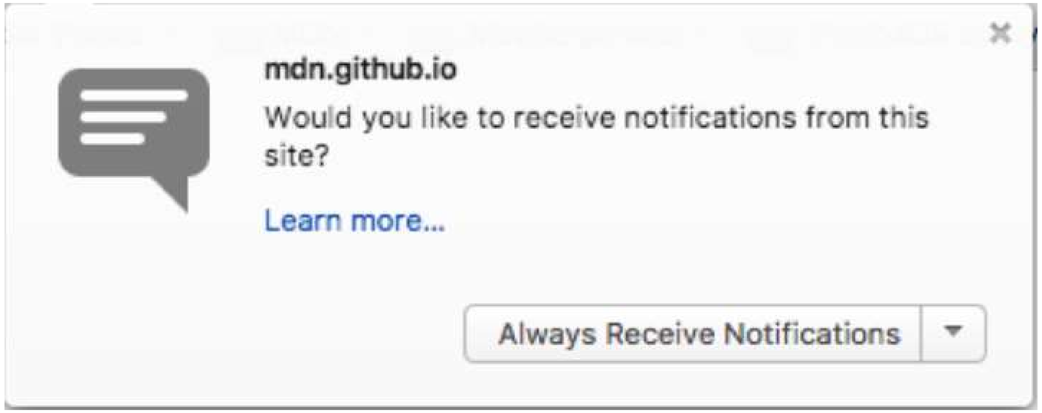
Some web APIs contain no events, but most contain at least a few. The handler properties that allow us to run functions when events fire

```
//Example: event handler in Web Audio API
// if track ends
audioElement.addEventListener("ended", () => {
        playBtn.setAttribute("class", "paused");    //set attribute for play Button as “Paused”
        playBtn.textContent = "Play";               //set text for play button as “Play”
});
```

How do APIs work?

4. They have additional security mechanisms where appropriate

Eg: The Notification API requests permission to be enabled from the user once calls to them are made in your code.



mdn.github.io

Would you like to receive notifications from this site?

Learn more...

Commonly used browser API

API	Purpose	Remarks
Document Object Model (DOM)	Document manipulation	Covered in Introduction to JS and this chapter
Fetch API	Retrieve data from server	Covered in JSON Older technique: XMLHttp Request object
Canvas	Drawing and manipulating graphics	Covered in MDN JS Loops and Events
Geolocation	Interaction with device hardware	This chapter

Commonly used third-party API

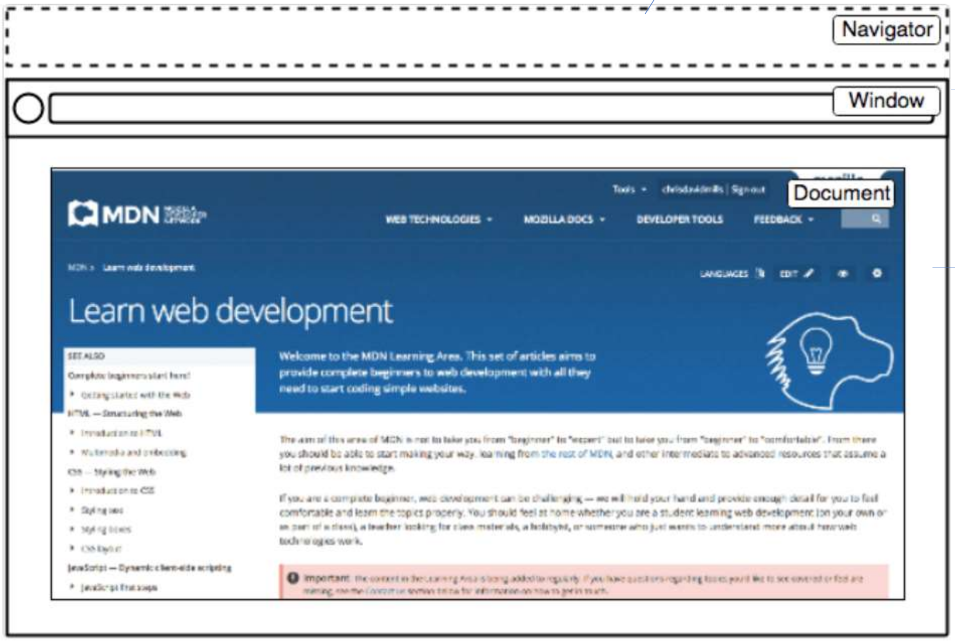
API	Purpose
Twitter API	To retrieve the latest tweets
Mapquest and Google Maps API	To do all sort of things with maps on your website
Pinterest API	Provide tools to manage Pinterest boards and pins to include them in your website
Facebook suite of API	Enables various parts of the Facebook ecosystem to benefit your app, such as login using Facebook login, accepting in-app payments and rolling out targeted ad-campaign

Document Object Model API

Document Object Model (DOM) API

Allows manipulation of HTML and CSS such as creating, removing and changing HTML, dynamically applying new styles to your page.

There are several main parts involved when you are viewing web pages: Navigator object, Windows object and Document object.



Learn web development

SEE ALSO

Complete beginners start here!

- Getting started with the Web

HTML — Structuring the Web

- Introduction to HTML
- Multimedia and embedding

CSS — Styling the Web

- Introduction to CSS
- Styling text
- Styling boxes
- CSS layout

JavaScript — Dynamic client-side scripting

- JavaScript first steps

Welcome to the MDN Learning Area. This set of articles aims to provide complete beginners to web development with all they need to start coding simple websites.

The aim of this area of MDN is not to take you from "beginner" to "expert" but to take you from "beginner" to "comfortable". From there you should be able to start making your way, learning from the rest of MDN, and other intermediate to advanced resources that assume a lot of previous knowledge.

If you are a complete beginner, web development can be challenging — we will hold your hand and provide enough detail for you to feel comfortable and learn the topics properly. You should feel at home whether you are a student learning web development (on your own or as part of a class), a teacher looking for class materials, a hobbyist, or someone who just wants to understand more about how web technologies work.

Important: The content in the Learning Area is being added to regularly. If you have questions regarding topics you'd like to see covered or feel are missing, see the Contribute section below for information on how to get in touch.

Navigator object – represents the browser's state and identity.
Usage: to retrieve language preference, media stream from the user's webcam, etc.

Window object – represents the tab that the browser loads into.
Usage: to manipulate the document in the browser, store data specific to the client-side browser data-specific

Document object – represents the actual document that loads into the browser.
Usage: to manipulate information on the HTML and CSSthe document in the browser, store data specific to the client-side browser data-specific

The document object model

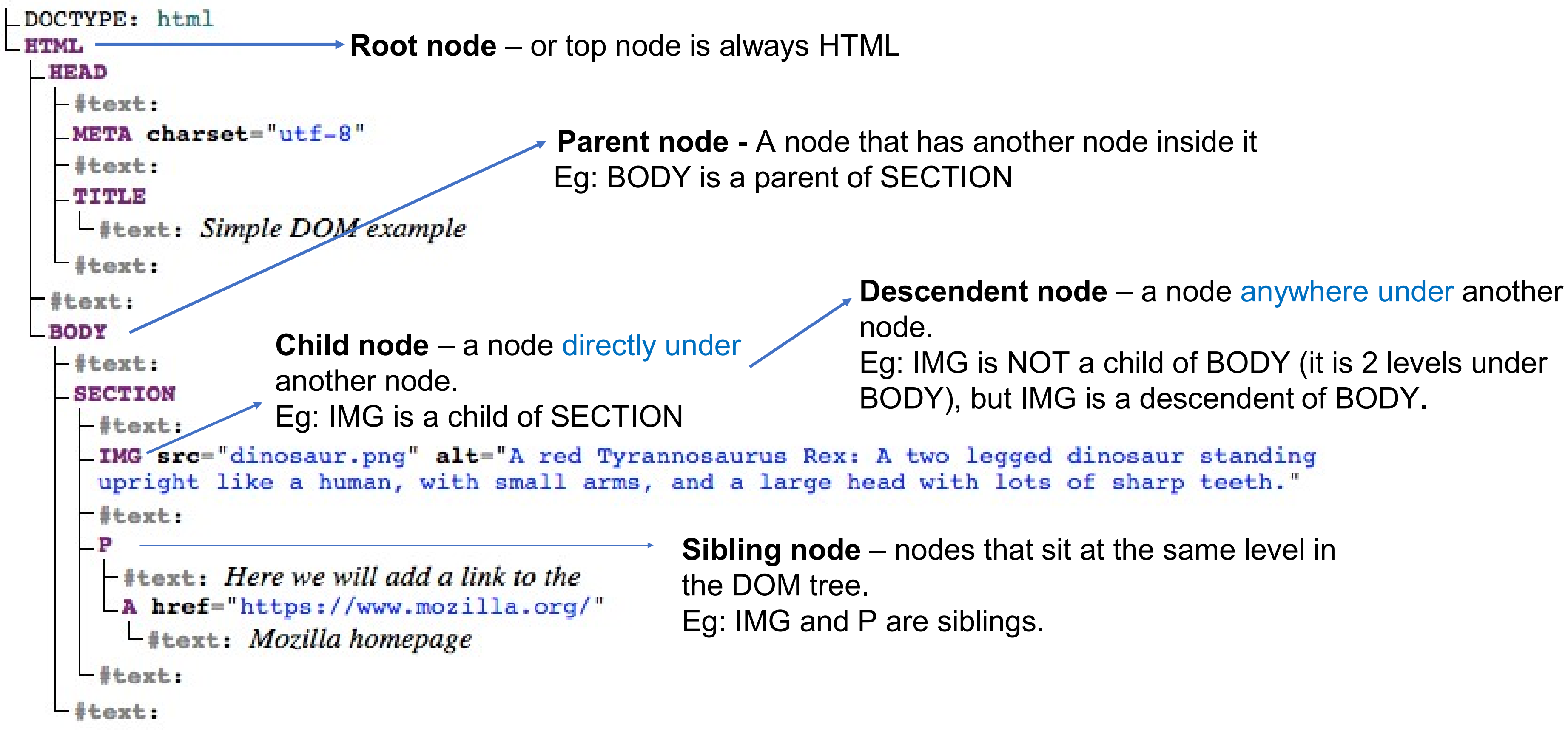
- The document loaded on your browser is represented by a “tree” structure that enables the HTML to be easily accessible by programming languages.

- Refer live example: Simple DOM example (mdn.github.io)

- The HTML document is represented by a “tree” structure.

- Each entry is known as a node.

- Nodes are referred based on their positions on the tree such as root, child, descendent, parent and sibling.



```
├DOCTYPE: html
└HTML
  ├HEAD
  │ ├#text:
  │ ├META charset="utf-8"
  │ ├#text:
  │ ├TITLE
  │ │ └#text: Simple DOM example
  │ └#text:
  ├#text:
  └BODY
    ├#text:
    ├SECTION
    │ ├#text:
    │ ├IMG src="dinosaur.png" alt="A red Tyrannosaurus Rex: A two legged dinosaur standing
    │ │ upright like a human, with small arms, and a large head with lots of sharp teeth."
    │ ├#text:
    │ ├P
    │ │ ├#text: Here we will add a link to the
    │ │ └A href="https://www.mozilla.org/"
    │ │   └#text: Mozilla homepage
    │ └#text:
    └#text:
```

Root node – or top node is always HTML

Parent node - A node that has another node inside it
Eg: BODY is a parent of SECTION

Child node – a node directly under another node.
Eg: IMG is a child of SECTION

Descendent node – a node anywhere under another node.
Eg: IMG is NOT a child of BODY (it is 2 levels under BODY), but IMG is a descendent of BODY.

Sibling node – nodes that sit at the same level in the DOM tree.
Eg: IMG and P are siblings.

Basic DOM manipulation

Method	Usage	Remarks
Document.querySelector()	Select the first matched element that appears in the document.	Recommended modern approach
Document.querySelectorAll()	Select every elements that matches the selector.	Recommended modern approach
Document.getElementById()	Selects an element with a given id attribute value	Older methods
Document.getElementsByTagName()	Selects all the elements on the page of a given type.	Older methods

Basic style manipulation

1) Adding inline styles

//Use HTMLElement.style property

para.style.color = “white”;

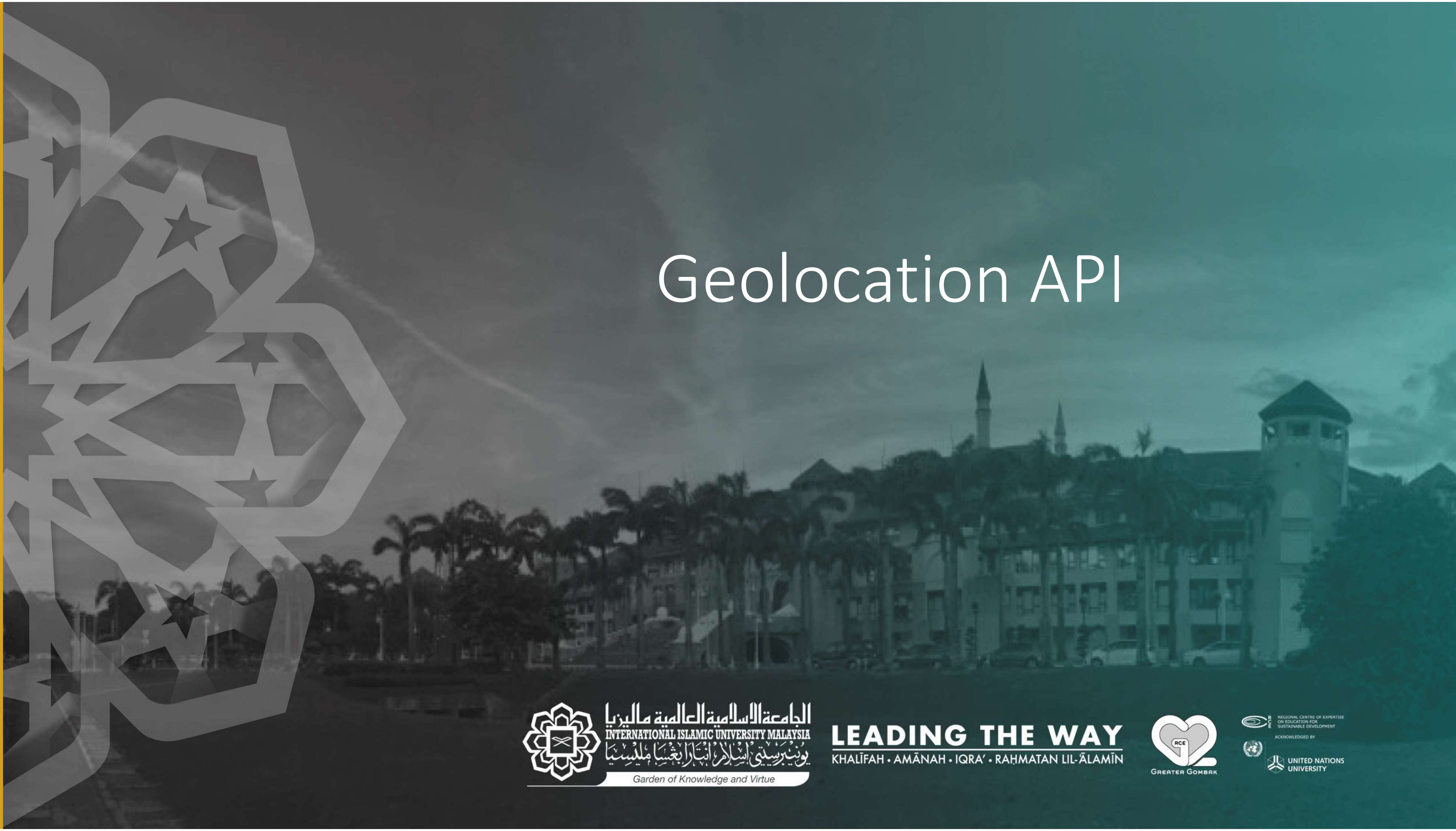
para.style.textalign = “center”;

2) Adding a style in the CSS rule

//Use Element.setAttribute() method

para.setAttribute("class", "highlight");

Geolocation API



Geolocation API

Allows the user to provide their location to web applications if they so desire. For privacy reasons, the user is asked for permission to report location information.

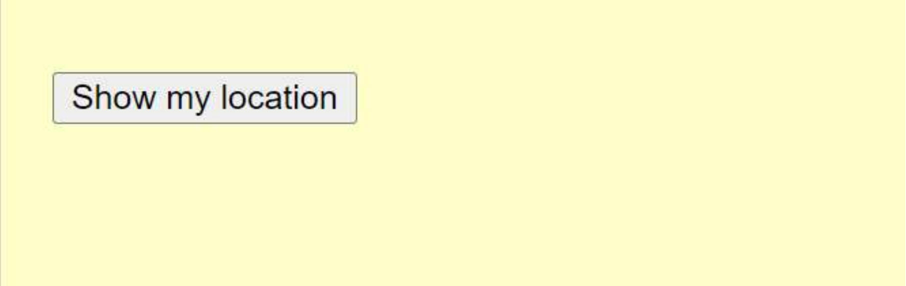
Usage example: To retrieve a user's location information in your web app, for example to plot their location on a map, or display personalized information relevant to their location.

is accessed via a call to navigator.geolocation; this will cause the user's browser to as

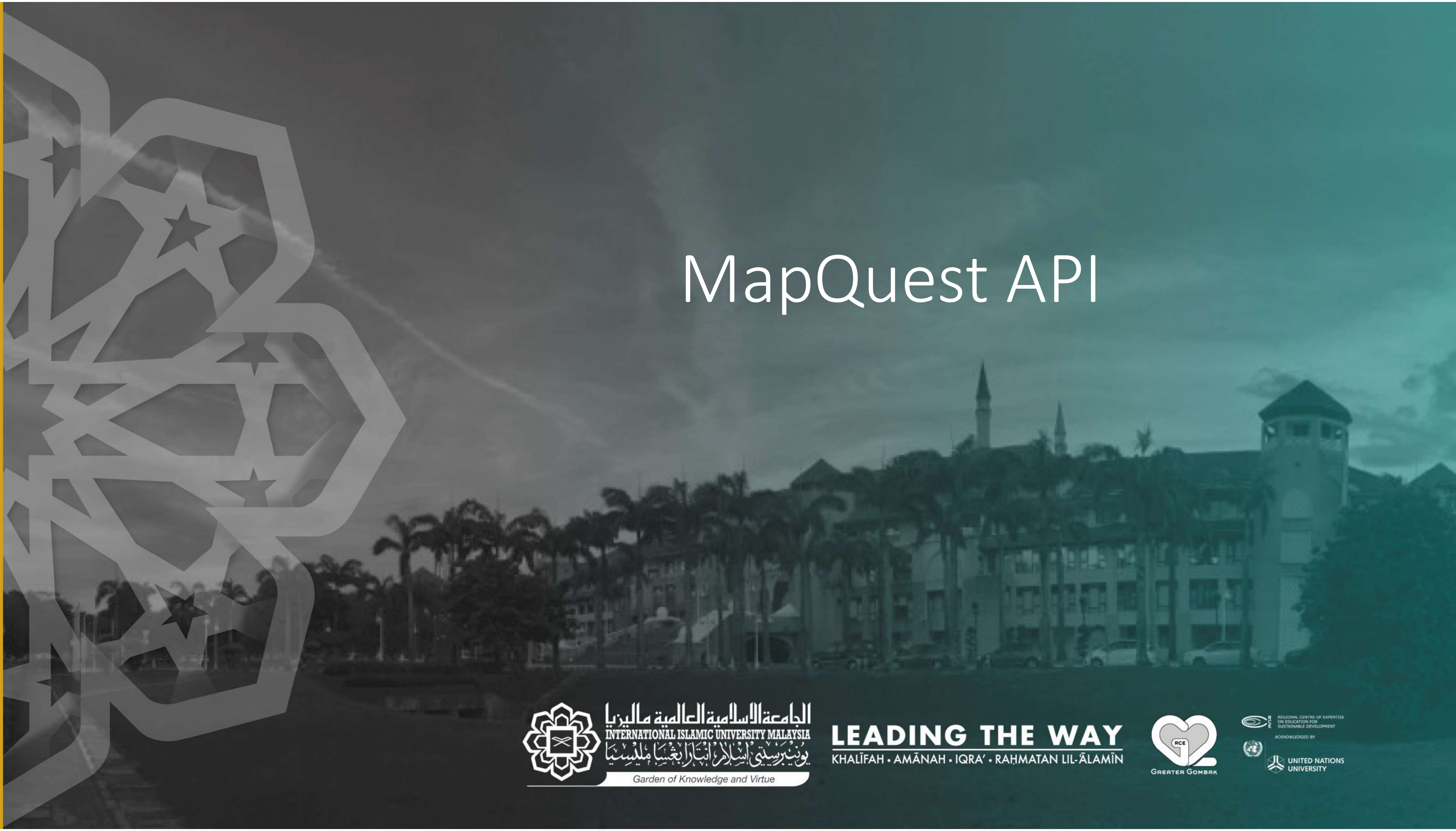
Geolocation API – usage

Interfaces	Purpose	Remarks
navigator.geolocation	**Entry point** to Geolocation API	A call to navigator.geolocation returns a Geolocation object instance, from where all functionalities can be accessed.
Geolocation	Main object of this API	Contains methods to retrieve the user's current position watch for change of position, and clear a set if previously set watch.
GeolocationPosition	Represents the position of the user – mandatory successful callback	A GeolocationPosition instance is returned by a successful call to one of the methods inside of Geolocation.
GeolocationPositionError	Optional error callback	If location retrieval is unsuccessful, the callback executes GeolocationPositionError object as its only parameter, contains error code and message

```
<!--html-->
<button id="find-me">Show my location</button><br />
<p id="status"></p>
<a id="map-link" target="_blank"></a>

//css
body {
      padding: 20px;
      background-color: #ffffc9;
}
button {
      margin: 0.5rem 0;
}
```



```
function geoFindMe() {
        const status = document.querySelector("#status");
        const mapLink = document.querySelector("#map-link");
        mapLink.href = "";
        mapLink.textContent = "";

        function success(position) {
                const latitude = position.coords.latitude;
                const longitude = position.coords.longitude;
                status.textContent = "";
                mapLink.href = `https://www.openstreetmap.org/#map=18/${latitude}/${longitude}`;
                mapLink.textContent = `Latitude: ${latitude} °, Longitude: ${longitude} °`;
        }

        function error() {
                status.textContent = "Unable to retrieve your location";
                }  if (!navigator.geolocation) {
                        status.textContent = "Geolocation is not supported by your browser";
                } else {    status.textContent = "Locating…";
                navigator.geolocation.getCurrentPosition(success, error);      }
}
document.querySelector("#find-me").addEventListener("click", geoFindMe);
```



MapQuest API

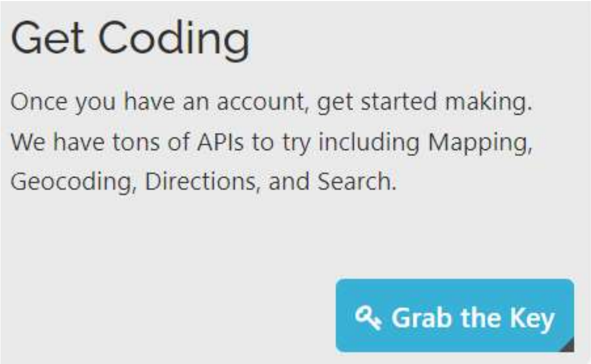
الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
يونيۏرسيتي اسلام انتارابڠسا مليسيا
Garden of Knowledge and Virtue

LEADING THE WAY
KHALĪFAH • AMĀNAH • IQRA' • RAḤMATAN LIL-ĀLAMĪN

RCE
GREATER GOMBAK

REGIONAL CENTRE OF EXPERTISE ON EDUCATION FOR SUSTAINABLE DEVELOPMENT
ACKNOWLEDGED BY
UNITED NATIONS UNIVERSITY

MapQuest API



Get Coding

Once you have an account, get started making. We have tons of APIs to try including Mapping, Geocoding, Directions, and Search.

1. Get API Key

Go to https://developer.mapquest.com/documentation/ and register for an account to get the API key.

2. Use Mapquest API to get Geo data

Eg: Use Mapquest's Geocoding API to take an address an associate it with latitude and longitude.

References

MDN Web Docs. Client-side Web APIs. Client-side web APIs - Learn web development | MDN (mozilla.org)

MapQuest Developer Documentation
https://developer.mapquest.com/documentation