

# URLs in JavaScript

---

Mohamed Maghraoui (@aliveupstairs)

What is a URL

The location API

Let's code a redirection browser extension

## What is a URL

---

## Examples of URLs

`http://www.example.com/index.html`

`https://github.com/`

`ftp://internet.address.edu/file/path/file.txt`

`whatsapp://send?text=Hello%20World!`

# The Anatomy of a URL

https:// en.film.com :443 /main/search.php ?text=manhattan&genre=comedy #cast

- Protocol/scheme\* Examples: http, https, ftp, mailto, tel, chrome.

# The Anatomy of a URL

https:// en.film.com :443 /main/search.php ?text=manhattan&genre=comedy #cast

- Protocol/scheme\* Examples: http, https, ftp, mailto, tel, chrome.
- Domain name\* An IP address can be used instead.

# The Anatomy of a URL

https:// en.film.com :443 /main/search.php ?text=manhattan&genre=comedy #cast

- Protocol/scheme\* Examples: http, https, ftp, mailto, tel, chrome.
- Domain name\* An IP address can be used instead.
- Port Default :80 for http. default :443 for https.

# The Anatomy of a URL

https:// en.film.com :443 /main/search.php ?text=manhattan&genre=comedy #cast

- Protocol/scheme\* Examples: http, https, ftp, mailto, tel, chrome.
- Domain name\* An IP address can be used instead.
- Port Default :80 for http. default :443 for https.
- Path Traditionally physical; now, mostly an abstraction by web servers.



# The Anatomy of a URL

`https:// en.film.com :443 /main/search.php ?text=manhattan&genre=comedy #cast`

- **Protocol/scheme\*** Examples: `http`, `https`, `ftp`, `mailto`, `tel`, `chrome`.
- **Domain name\*** An IP address can be used instead.
- **Port** Default `:80` for `http`. default `:443` for `https`.
- **Path** Traditionally physical; now, mostly an abstraction by web servers.
- **Parameters/query/search** & separated key/value pairs.

# The Anatomy of a URL

https:// en.film.com :443 /main/search.php ?text=manhattan&genre=comedy #cast

- Protocol/scheme\* Examples: http, https, ftp, mailto, tel, chrome.
- Domain name\* An IP address can be used instead.
- Port Default :80 for http. default :443 for https.
- Path Traditionally physical; now, mostly an abstraction by web servers.
- Parameters/query/search & separated key/value pairs.
- Fragment/hash Client side; does not get sent with the request; Browsers scroll to position.

# URL Encoding

space	!	#	\$	%	&	'	(	)	*
%20	%21	%23	%24	%25	%26	%27	%28	%29	%2A

+	,	/	:	;	=	?	@	[	]
%2B	%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D

Example:

`https://wa.me/1555555555?text=I%27d%20like%20to%20chat%20with%20you`

## The location API

---

## window.location Properties

https:// en.film.com :80 /1979 ?text=manhattan&genre=comedy #cast

- location.protocol: "https:".

## window.location Properties

https:// en.film.com :80 /1979 ?text=manhattan&genre=comedy #cast

- location.protocol: "https:".
- location.hostname: "en.film.com".

## window.location Properties

https:// en.film.com :80 /1979 ?text=manhattan&genre=comedy #cast

- location.protocol: "https:".
- location.hostname: "en.film.com".
- location.port: "80".

## window.location Properties

https:// en.film.com :80 /1979 ?text=manhattan&genre=comedy #cast

- location.protocol: "https:".
- location.hostname: "en.film.com".
- location.port: "80".
- location.path\*: "/1979".



## window.location Properties

https:// en.film.com :80 /1979 ?text=manhattan&genre=comedy #cast

- `location.protocol`: "https:".
- `location.hostname`: "en.film.com".
- `location.port`: "80".
- `location.path*`: "/1979".
- `location.search`: "?text=manhattan&genre=comedy". Can be used in conjunction with `URLSearchParams`.

## window.location Properties

https:// en.film.com :80 /1979 ?text=manhattan&genre=comedy #cast

- `location.protocol`: "https:".
- `location.hostname`: "en.film.com".
- `location.port`: "80".
- `location.path*`: "/1979".
- `location.search`: "?text=manhattan&genre=comedy". Can be used in conjunction with `URLSearchParams`.
- `location.hash`: "#cast".

- `location.assign()`: load a URL.

- `location.assign()`: load a URL.
- `location.reload(Boolean)`: reload from server or cache based on boolean passed.

- `location.assign()`: load a URL.
- `location.reload(Boolean)`: reload from server or cache based on boolean passed.
- `location.replace()`: similar to `.assign` but without history.

Let's code a redirection browser  
extension

---

- Tampermonkey extension: chromium, firefox.
- JavaScript.

## Script code

```
// ==UserScript==  
// @name      Wikipedia mobile  
// @description Redirect to mobile version of wikipedia  
// @match     https://en.wikipedia.org/*  
// @run-at    document-start  
// ==/UserScript==  
  
(function() {  
    'use strict';  
    if (location.hostname === 'en.wikipedia.org') {  
        location.hostname = 'en.m.wikipedia.org';  
    }  
})();
```



Do you wish to add anything?

Thank You