# DESARROLLO DE APLICACIONES WEB FRONT END

Módulo 4

OTEC PLATAFORMA 5 CHILE

MÓDULO 4


# CLASE 2

Introducción

Sintaxis de la URL

Componentes

Parámetros de búsqueda

Codificación

# Introducción

**Hoy trabajaremos con el Objeto URL, el cual nos brinda una interfaz que sirve para crear y analizar URLs.**

**En sí no es un objeto que tendrá que utilizar muy comúnmente, pero es importante que sepa cómo está compuesta, para poder utilizar todos los datos que nos aporta.**

# *Sintaxis de una URL*

**La sintaxis para crear un nuevo objeto URL es:**

```
new URL(url, [base])
```

- **url: La URL completa o ruta única (si se establece base, mira a continuación),**
- **base: una URL base opcional: si se establece y el argumento url solo tiene una ruta, entonces la URL se genera relativa a base.**

**Por ejemplo:**

```
let url = new URL('http://plataforma5.iicap.cl/user/profile');
```

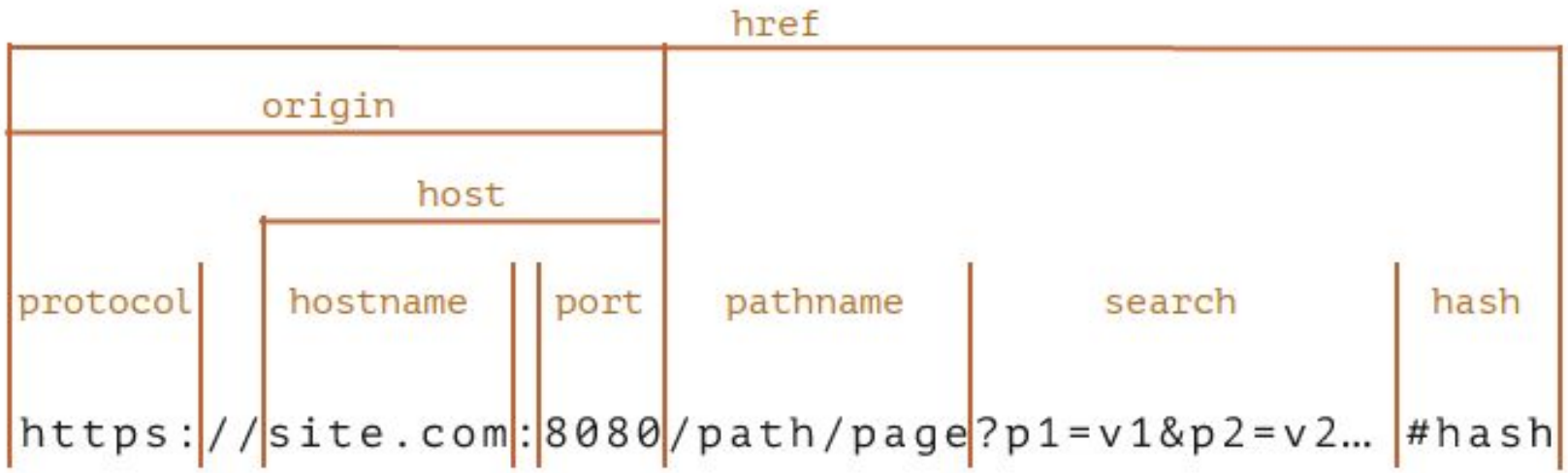
**Estas dos URLs son las mismas:**

```
let url1 = new URL('http://plataforma5.iicap.cl/user/profile');
let url2 = new URL('/user/profile','https://javascript.infohttp://plataforma5.iicap.cl');

alert(url1); // http://plataforma5.iicap.cl/user/profile
alert(url2); // http://plataforma5.iicap.cl/user/profile
```

# *Componentes de una URL*

- **href: es la url completa, igual que url.toString()**
- **protocol: acaba con el carácter dos puntos :**
- **search: un string de parámetros, comienza con el signo de interrogación ?**
- **hash: comienza con el carácter de hash #**



Podemos usar un objeto URL en fetch o XMLHttpRequest, casi en todas partes donde se espera un URL-string.

Generalmente, un objeto URL puede pasarse a cualquier método en lugar de un string, ya que la mayoría de métodos llevarán a cabo la conversión del string, eso convierte un objeto URL en un string con URL completa.

# Parámetros de búsqueda “?...”

Digamos que queremos crear una url con determinados parámetros de búsqueda, por ejemplo, https://google.com/search?query=JavaScript.

Podemos proporcionarlos en el string URL:

```
new URL('https://google.com/search?query=JavaScript')
```

Pero los parámetros necesitan estar codificados si contienen espacios, letras no latinas, entre otros.

Por lo que existe una propiedad URL para eso: url.searchParams, un objeto de tipo URLSearchParams.

# Parámetros de búsqueda “?...”

El URLSearchParams proporciona métodos convenientes para los parámetros de búsqueda:

- append(name, value): añade el parámetro por name,
- delete(name): elimina el parámetro por name,
- get(name): obtiene el parámetro por name,
- getAll(name): obtiene todos los parámetros con el mismo name (Eso es posible, por ej. `?user=Flor&user=Javi`),
- has(name): comprueba la existencia del parámetro por name,
- set(name, value): establece/reemplaza el parámetro,

# *Codificación*

Existe un estándar que define cuales caracteres son permitidos en URLs y cuales no.

Esos que no son permitidos, deben ser codificados, por ejemplo letras no latinas y espacios – reemplazados con sus códigos UTF-8, con el prefijo %, tal como %20 (un espacio puede ser codificado con +, por razones históricas, pero esa es una excepción).

La buena noticia es que los objetos URL manejan todo eso automáticamente. Nosotros sólo proporcionamos todos los parámetros sin codificar, y luego convertimos la URL a string:

Como podemos ver, ambos Тест en la ruta url y ъ en el parámetro están codificados.

La URL se alarga, ya que cada letra cirílica es representada con dos bytes en UTF-8, por lo que hay dos entidades %...

```
// Usando algunos caracteres
cirílicos para este ejemplo

let url = new
URL("https://ru.wikipedia.org/wiki/T
ест");

url.searchParams.set("key", "ъ");
alert(url);
//https://ru.wikipedia.org/wiki/%D0%
A2%D0%B5%D1%81%D1%82?key=%D1%8A
```

MÓDULO 4

# GRACIAS

Aquí finaliza la clase n°2 del módulo 4

OTEC PLATAFORMA 5 CHILE