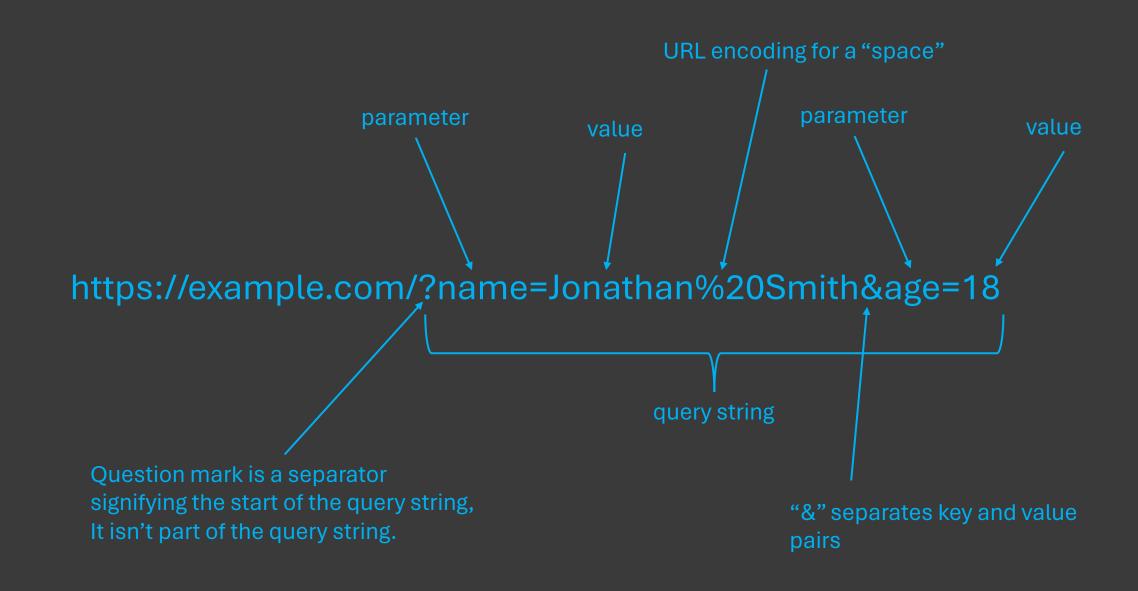


- Include data in your URL
  - https://example.com/?name=Jonathan% 20Smith&age=18
- It is made up of multiple key value pairs, key=value
- Each pair is separated by &
- Value must be encoded



#### **URL** Encoding

- URL encoding is a mechanism used to convert characters that are not allowed or reserved in a URL into a format that can be safely transmitted over the internet.
- This ensures that URLs remain valid and can be correctly interpreted by web browsers and servers.
- URL encoding works by replacing unsafe characters with a special sequence of characters called percent-encoding or percent-escaping.
- In percent-encoding, each unsafe character is replaced with a percent sign (%) followed by two hexadecimal digits that represent the character's ASCII value.
- For example:
  - The space character () is encoded as %20.
  - The ampersand (&) is encoded as %26.
  - The question mark (?) is encoded as %3F.





## Examples

- https://www.example.com/about#team%20info
- https://www.example.com/search?q=hello%20world&category=science%26tech
- https://www.example.com/products%20and%20services

- > encodeURI("https://example.com?name=Joe Smith&mood=angry!&other=[{\*%@\$@!@f")
- < 'https://example.com?name=Joe%20Smith&mood=angry!&other=%5B%7B\*%25@\$@!@%C2%A3'</pre>
- > decodeURI('https://example.com?name=Joe%20Smith&mood=angry!&other=%5B%7B\*%25@\$@!@%C2%A3')
- < 'https://example.com?name=Joe Smith&mood=angry!&other=[{\*%@\$@!@£'</pre>

# encodeURI and decodeURI

.........

Function available in the global scope of the browser

### Linking with URL Params

```
window.dotify.components.createPlaylistItem = (playlist) => {
   return
   <div class="column is-4">
     <section class="card has-text-centered">
       <header class="card-header">
        ${playlist.name}
        </header>
       <div class="card-image">
        <figure class="image">
          <img src="${playlist.imageUrl}" alt="Image">
        </figure>
       </div>
                                      Adding URL params from our data
       <article class="card-content">
        ${playlist.description}
        </article>
       <footer class="card-footer">
        <a class="card-footer-/cem button has-background-grey-lighter">Play</a>
        <a href="/playlist/?name=${playlist.name}" class="card-footer-item button has-background-grey-lighter">Open</a>
        <a class="card-footer-item button has-background-grey-lighter">Remove</a>
       </footer>
     </section>
   </div>
```

### Reading Query Parameters

http://localhost:8080/playlist/?name=Chill

```
const urlParams = new URLSearchParams(window.location.search);
const playlistName = urlParams.get('name');
```

http://localhost:8080/cityFocus/?city=Berlin

```
const urlParams = new URLSearchParams(window.location.search);
const currentCity = urlParams.get(city');
```

```
document.addEventListener('DOMContentLoaded', () => {
    const urlParams = new URLSearchParams(window.location.search);
    const playlistName = urlParams.get('name');

    const playlist = dotify.dataStore.list().find(item => item.name === playlistName);
    document.getElementById('page-heading').textContent = playlist.name;

    const main = document.querySelector('main');
    main.innerHTML = main.innerHTML + dotify.components.createPlaylistItem(playlist);
    main.innerHTML = main.innerHTML + dotify.components.createPlaylistList(playlist.songs);
});
```