BCSCHOOL

Empowering Digital Skills For The Jobs Of The Future

BC Soft
consulenza sistemi informativi

by

AiRED
Artificial Intelligence Ricerca e Sviluppo

Dart Client Side

Docente

Claudia Infante

claudia.infante@bcsoft.net

Sommario

- Elementi di navigazione in Dart

Elementi di navigazione in Dart

Gli utenti sono abituati a utilizzare gli strumenti di navigazione integrati nel browser, come i pulsanti avanti e indietro per navigare tra le applicazioni web nello stesso modo in cui fanno da anni con i normali siti web.

Elementi di navigazione in Dart

Gli utenti si aspettano anche che le app abbiano tempi di risposta rapidi, cosa che le app ottengono persistendo i dati offline.

La moderna tecnologia dei browser consente di soddisfare le aspettative degli utenti.

Elementi di navigazione in Dart

In un'applicazione web a pagina singola, quando l'utente passa da una vista all'altra, ad esempio da una vista elenco a una vista modifica, si aspetta di poter usare il pulsante indietro del browser per tornare indietro.

WINDOW BACK BUTTON

Elementi di navigazione in Dart

Quando l'utente passa a un'altra pagina web e l'URL cambia, il browser web inserisce il nuovo URL nell'elenco della cronologia.

pushState()

Grazie al metodo pushState() è possibile definire le pagine che si desidera raggiungere.

pushState()

Affinchè ci si posta spostare da una view all'altra, si utilizza il metodo pushState().

pushState()

La funzione pushState() accetta tre parametri: alcuni dati di stato, un titolo (attualmente non utilizzato dalla maggior parte dei browser) e un nuovo path da visualizzare nell'URL.

navigateTo

La funzione navigateTo definisce una funzione che ha come parametri la pagina da raggiungere e il titolo.

Tramite la notazione a punti, si accede alla proprietà history dell'oggetto window.
Grazie al metodo pushState() si inviano le informazioni dello stato allo stack della history. Questo aggiorna l'url del browser senza causare un reload della pagina

```
19   void navigateTo(String page, String title) {
20     window.history.pushState(null, title, page);
```

pushState

I parametri di pushState sono:

- Null : lo stato dell'oggetto associato alla richiesta
- Title : il titolo della pagina
- Page : l'URL della pagina da raggiungere

```
19   void navigateTo(String page, String title) {
20     window.history.pushState(null, title, page);
```

pushState

Si effetua poi una richiesta http per recuperare il contenuto della pagina specifica.

Il metodo then gestisce la risposta ricevuta che, grazie al recupero di un elemento del DOM, sarà mostrato nella view.

In questo modo si aggiornerà solo quella porzione di codice, senza richiedere ulteriori informazioni al server.

```
22    HttpRequest.getString(page).then((response) {
23      querySelector('#content')?.innerHtml = response;
24    });
```

navigateTo()

navigateTo() riceverà come parametro la pagina html da raggiungere e il path che sarà mostrato in fase di navigazione:

```
void main() {
  querySelector('#home')?.onClick.listen((_) {
    navigateTo('home.html', 'Home');
  });

  querySelector('#about')?.onClick.listen((_) {
    navigateTo('about.html', 'About');
  });

  querySelector('#contact')?.onClick.listen((_) {
    navigateTo('contact.html', 'Contact');
  });

  navigateTo('home.html','');
}
```

Navigazione

main.dart

main.dart index.html

web > main.dart > ...

```dart
import 'dart:html';

Run | Debug
void main() {
  querySelector('#home')?.onClick.listen((_) {
    navigateTo('home.html', 'Home');
  });

  querySelector('#about')?.onClick.listen((_) {
    navigateTo('about.html', 'About');
  });

  querySelector('#contact')?.onClick.listen((_) {
    navigateTo('contact.html', 'Contact');
  });

  navigateTo('home.html','');
}

void navigateTo(String page, String title) {
  window.history.pushState(null, title, page);

  HttpRequest.getString(page).then((response) {
    querySelector('#content')?.innerHtml = response;
  });
}

```

Navigazione

Index.html

main.dart index.html

web > index.html > ...

```
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Dart Web Browser Navigation</title>
7   </head>
8   <body>
9       <button id="home">Home</button>
10      <button id="about">About</button>
11      <button id="contact">Contact</button>
12      <div id="content"></div>
13
14      <script defer src="main.dart.js"></script>
15  </body>
16  </html>
17
```

Parametri

In fase di navigazione è a volte necessario ricevere delle informazioni in modo dinamico tramite dei parametri.

Ad esempio, interagendo con un elemento del DOM, si desidera ricevere informazioni più specifiche dirottando l'utente su un'altra pagina.

Parametri

Nell'esempio riportato, la pagina Contact presenta una lista di nominativi

contact.html

web > contact.html > ...

```
1   <!-- contacts.html -->
2   <!DOCTYPE html>
3   <html lang="en">
4   <head>
5       <meta charset="UTF-8">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Contacts</title>
8   </head>
9   <body>
10      <ul id="contactList">
11          <li><button class="contact" data-name="Alice" data-telephone="123456789">Alice</button></li>
12          <li><button class="contact" data-name="Bob" data-telephone="987654321">Bob</button></li>
13      </ul>
14
15      <script defer src="contacts.js"></script>
16  </body>
17  </html>
18
```

Parametri

Ogni contatto dovrà essere un elemento cliccabile in modo da determinare la navigazione. Si stabiliscono anche gli attributi da custom come data-nomeattributo.

Parametri

contact.html

web > contact.html > ...

```
1   <!-- contacts.html -->
2   <!DOCTYPE html>
3   <html lang="en">
4   <head>
5       <meta charset="UTF-8">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Contacts</title>
8   </head>
9   <body>
10      <ul id="contactList">
11          <li>
12              <button
13              class="contact"
14              data-name="Italian Customer center"
15              data-telephone="123456789">
16              Italian Customer center
17          </button>
18      </li>
19          <li>
20              <button
21              class="contact"
22              data-name="English Customer center"
23              data-telephone="987654321">
24              English Customer center
25          </button>
26      </li>
27      </ul>
28
29      <script defer src="contacts.js"></script>
30  </body>
31  </html>
32
```

Parametri

Si definisce un file javascript associato che, recuperato l'elemento che scatena l'evento, implementa il metodo per la navigazione.

contacts.js

```
// contacts.js
document.addEventListener("DOMContentLoaded", function () {
  const contactButtons = document.querySelectorAll(".contact");

  contactButtons.forEach((button) => {
    button.addEventListener("click", function () {
      const name = button.getAttribute("data-name");
      const telephone = button.getAttribute("data-telephone");
      navigateToContactDetails(name, telephone);
    });
  });
});

function navigateToContactDetails(name, telephone) {
  const url = `contactDetails.html?name=${name}&telephone=${telephone}`;
  window.location.href = url;
}
```

Parametri

Si definisce la pagina di dettaglio :

contactDetails.html

```
<!-- contactDetails.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Details</title>
</head>
<body>
    <h1>Contact Details</h1>
    <p>Name: <span id="contactName"></span></p>
    <p>Telephone: <span id="contactTelephone"></span></p>

    <script defer src="contactDetails.js"></script>
</body>
</html>
```

Parametri

E il file javascript associato che consentirà di estrarre le informazioni necessarie per mostrare gli attributi ricevuti nella view della pagina dettaglio:

JS contactDetails.js

web > JS contactDetails.js > ...

```
1   // contactDetails.js
2   document.addEventListener("DOMContentLoaded", function () {
3     const urlParams = new URLSearchParams(window.location.search);
4     const name = urlParams.get("name");
5     const telephone = urlParams.get("telephone");
6
7     document.getElementById("contactName").textContent = name;
8     document.getElementById("contactTelephone").textContent = telephone;
9   });
10
```

Domande e approfondimenti