

Task Home

Task Home

Data: 12/09/2024

Task: Il team che si occuperà della grafica della Homepage dovrà gestire tutti i tab della pagina e renderla simile graficamente al mockup previsto dal Dipartimento.

Creato servizio mock

Visto la necessità di dati di prova, si è provveduto ad installare la libreria javascript json-server che crea un mock api server. Il server può essere avviato con il comando

```
npm run startDB
```

nella cartella frontend. Si possono aggiungere servizi andando a modificare il file db/db.json.

Si è creato un servizio finto chiamato ufficiRubrica (nome temporaneo, de definire quello definitivo), che risponde al seguente indirizzo

<http://localhost:3000/ufficiRubrica>

e che restituisce un array di oggetti del tipo **IOffice**.

Al momento per la creazione della home abbiamo bisogno dei campi definiti dal model **IOffice**, composto da:

```
export interface IOffice{
  codiceUfficio:string;
  codiceUfficioSuperiore?:string;
  nomeUfficio:string;
  nomeTitolare:string;
  coloreSfondo?:string;
  descrizioneUfficio?: string;
  children:[];
  personale?: Array<IPersonale>;
}
```

```
export interface IPersonale{
  nominativo: string;
  qualifica: string;
  codiceUfficio: string;
  contatti: Array<IContatto>;
}
```

```

}

export interface IContatto{
  tipo:string;
  contatto:string;
}

```

Il campo children è a sua volta un array di oggetti IOffice contenente gli uffici figli.

Il campo personale è un array di personale appartenente all'ufficio con un contatto.

Il campo contatti è un array di contatti dell'utente.

I servizio ufficiRubrica dovrebbe restituire l'elenco degli uffici e subuffici centrali del dipartimento (da vedere se usare lo stesso anche per gli uffici periferici).

N.B.: da rivedere la struttura con gli array, è un pò macchinoso la ricerca su valori e sottovalori.

store nxgrx

Si è provveduto all'installazione e configurazione dello store nxgrx. Lo store sarà usato per mantenere lo stato dell'applicazione. Al momento è composto da:

```

{
  userState: {
    user: [
      {
        nome: 'Lamberto'
      },
      {
        nome: 'Angela'
      }
    ]
  },
  authUser: {
    token: 'token-autenticazione-jwt',
    decodeToken: {
      nbf: 1721291746,
      iat: 1721291746,
      jti: 'id123456',
      aud: 'http://localhost',
      name: '',
      sub: 'AAABBB00F10B123K',
      iss: 'AppRubDip',
      tipo: 'VVF',
      sedi: [
        {
          codSede: 1,
          isSuperUser: true,

```

```

        uffici: [
            {
                idUff: 1,
                idProfilo: 1
            },
            {
                idUff: 2,
                idProfilo: 1
            }
        ]
    },
    loggedUser: '',
    isLoading: false,
    error: ''
},
rubrica: {
    rubrica: [
        {
            codiceUfficio: '00.0',
            nomeTitolare: 'nomeTitolare',
            nomeUfficio: 'nomeUfficio',
            coloreSfondo: '#495380',
            descrizione: 'descrizione',
            children: [
                .....
            ]
        }
    ]
},
]
}
}

```

I campi **userState** e **authUser** sono delle prove che stavo facendo.

Il campo **authUser** dovrebbe contenere il token jwt dell'applicativo e il token decodificato; ma erano delle prove.

In rubrica ho appoggiato temporaneamente i dati che restituisce il servizio ufficiRubrica, così da poterli manipolare sul client (probabilmente è da cambiare nome al campo).

Al campo rubrica si può accedere con le istruzioni del tipo:

```

this._storeApp$.select(selectHome).subscribe(
  items => {
    // console.log('homeItems', items);
    this.homeItems = [...items?.rubrica];
    this.childrenSelected=this.homeItems[0].children;
  };
);

```

dove **selectHome** è un selettore di ngrx che trovate definito nella cartella **store/selector**.

Lo store può essere visualizzato con l'estensione Redux DevTools del browser.

Componenti

app-root rinominato in vvfrubrica-root

Si sono apportate modifiche per sostituire i dati statici con i dati dinamici presi dallo store.

Si è creata funzione per ricevere i dati sull'ufficio selezione dall'componente **app-item-aoo**; i dati ricevuti vengono passati al componente **app-sottouffici**

app-item-aoo rinominato in vvfrubrica-uffici

Il component rappresenta il singolo ufficio.

Si è reso espandibile il component: se si clicca si estende e mostra dei dati (da valutare se può servire)

Se si clicca sul component vengono visualizzati gli uffici figli sul lato sinistro.

Al click restituisce al component chiamante l'array con gli uffici figli.


app-sottouffici rinominato in vvfrubrica-sottouffici

Il component rappresenta i sottouffici ma al momento richiama app-item-aoo. Da adeguare in base al layout dei sottouffici (che ancora non ho capito come impaginare !!!!!)

vvfrubrica-personale

Il component rappresenta la singola persona dell'ufficio collegata ad un contatto o più (non ho ancora capito che layout deve avere)

Screenshot



Do

Terza riga
(in basso)

Capo Dipartimento

Prefetto Franceschelli

Capo del Corpo Nazionale
dei Vigili del Fuoco Vice
Capo Dipartimento Vicario

Ing. Carlo Dall'Oppio

Ufficio di collegamento del
Capo Dipartimento e del
Capo del CNVVF


Ing. Carlo Dall'Oppio

Ufficio number X

Uffici dipendenti

Componenti

Descrizione




figlio1

figlio1

figlio2

nome titolare



Do

Terza riga
(in basso)

Capo Dipartimento

Prefetto Franceschelli

Descrizione: Lorem ipsum dolor sit amet consectetur adipiscing elit. Asperiores aliquid optio nihil eos. Officis, dignissimos delectus. Sit rerum unde repellat deleniti dignissimos, dolor cupiditate odio facere nihil aut, necessitatibus sunt.

Capo del Corpo Nazionale
dei Vigili del Fuoco Vice
Capo Dipartimento Vicario


Ing. Carlo Dall'Oppio

Ufficio di collegamento del
Capo Dipartimento e del
Capo del CNVVF

Uffici dipendenti

Componenti

Descrizione



figlio1

figlio1

Descrizione: Lorem ipsum dolor sit amet consectetur adipiscing elit. Asperiores aliquid optio nihil eos. Officis, dignissimos delectus. Sit rerum unde repellat deleniti dignissimos, dolor cupiditate odio facere nihil aut, necessitatibus sunt.

figlio2

nome titolare