

Guida Completa ai Moduli nel Frontend

Guida Completa ai Moduli nel Frontend

Guida Completa ai Moduli nel Frontend

Una guida pratica per comprendere e costruire moduli in un'applicazione frontend modulare (come quella usata in ambienti enterprise).

1. Cos' un Modulo nel Frontend?

Un **modulo** una porzione indipendente dell'applicazione che racchiude tutto ci che serve per una funzionalit:

- Componenti
- Servizi/API
- Route
- Interfacce dati
- Permessi

> Esempi di moduli: "Utenti", "Notifiche", "Documenti", "Flussi"...

2. Perch usare un'architettura modulare?

- **Organizzazione**: ogni funzionalit ha il suo spazio

Guida Completa ai Moduli nel Frontend

- **Scalabilit **: il codice cresce senza diventare caotico
- **Isolamento**: puoi aggiornare un modulo senza toccare gli altri
- **Riutilizzabilit **: un modulo pu essere portato in altri progetti
- **Controllo dei permessi**: ogni modulo pu essere mostrato solo a utenti autorizzati

3. Come si crea un Modulo da zero?

Passaggi base:

1. **Crea la cartella**

```
`/src/module/tuo-modulo/`
```

2. **Crea il Service**

```
```ts
```

```
// NotifyService.tsx
```

```
export const NotifyModuleName = "AgilaeNotify";
```

```
export default function NotifyService() {
```

```
// fetch, API, funzioni con token ecc.
```

```
}
```

```
...
```

## Guida Completa ai Moduli nel Frontend

### 3. **\*\*Crea il Module.ts\*\***

```
``ts

// NotifyModule.ts

import { Notifications as NotifyIcon } from "@mui/icons-material";

export const NotifyModule = (token, module, callback) => {

 module.menuItems = [

 {

 text: "Notifiche",

 icon: <NotifyIcon />,

 href: "/dashboard/notify/notifiche",

 permission: "ROLE_Administrator",

 priority: 1500,

 },

];

 callback(module);

};

``
```

### 4. **\*\*Registra il modulo nel ModuleProvider\*\***

## Guida Completa ai Moduli nel Frontend

```
``ts

case NotifyModuleName:

 NotifyModule(token, module, refreshModuleCallback);

break;

``
```

### 5. **Crea le interfacce**

```
``ts

export interface NotificationDto {

 id?: number;

 text: string;

 state: string;

}

``
```

### 6. **Crea la pagina** (Next.js)

```
``

/app/dashboard/notify/notifiche/page.tsx

``
```

### 7. **Aggiungi il ModuleProvider**

# Guida Completa ai Moduli nel Frontend

```
```ts
```

```
// ModuleProvider.tsx
```

```
import { NotifyModuleName } from "@module/agilaenotify/services/NotifyService";
```

```
import { NotifyModule } from "@module/agilaenotify/NotifyModule";
```

```
const refreshModule = useCallback((module: ModuleService) => {
```

```
  switch (module.name) {
```

```
    case NotifyModuleName:
```

```
      NotifyModule(token, module, refreshModuleCallback);
```

```
      break;
```

```
    // altri moduli...
```

```
  default:
```

```
    refreshModuleCallback(module);
```

```
  }
```

```
}, [token]);
```

```
```
```

```

```

## 4. Cos' un menuItem?

l'oggetto che definisce **come il modulo appare nella sidebar**.

# Guida Completa ai Moduli nel Frontend

Esempio:

```
```ts

{

  text: "Notifiche",

  icon: <NotifyIcon />,

  href: "/dashboard/notify/notifiche",

  permission: "ROLE_Administrator",

  priority: 1500,

}

```
```

## Campi principali:

- `text`: etichetta del menu
- `icon`: icona visuale
- `href`: link da aprire
- `permission`: chi pu vederlo
- `priority`: ordine nella sidebar

---

# Guida Completa ai Moduli nel Frontend

## 5. Come si visualizza un modulo nella Sidebar?

Deve avere un `menuItem` valido

Deve essere registrato nel `ModuleProvider`

L'utente deve avere il permesso richiesto

La sidebar renderizza automaticamente tutti i `menuItems` attivi e permessi

---

## 6. Come si crea la pagina collegata al modulo?

Se usi Next.js (App Router):

...

/app/dashboard/notify/notifiche/page.tsx

...

Esempio:

```
```tsx
```

```
export default function NotifyPage() {
```

```
  return <div>Notifiche in arrivo!</div>;
```

```
}
```

```
```
```

# Guida Completa ai Moduli nel Frontend

Se usi React Router:

```
```tsx
```

```
<Route path="/dashboard/notify/notifiche" element={<NotifyPage />} />
```

```
```
```

```

```

## 7. Come gestire i permessi per modulo?

Il campo `permission` in `menuItem` indica **quale ruolo richiesto** per visualizzare il modulo nella UI.

> Il sistema deve disporre di `checkPermission(permission)` per filtrare i menu

```

```

## 8. Best practice e consigli

- Dai un `priority` coerente ai menu (1000 = in basso, 10 = in alto)
- Dividi bene i file: `/services`, `/pages`, `/interfaces`
- Crea interfacce forti (`Dto`, `Entity`) per ogni modulo
- Usa `useModule()` per accedere al modulo attivo ovunque nel codice
- Testa con utenti con e senza permessi diversi

```

```

**Fine prima versione**



# Guida Completa ai Moduli nel Frontend

Questa guida sar aggiornata man mano con nuove domande, esempi e casi pratici.

Prossimo passo: vuoi aggiungere una sezione con "Come si testano i moduli" oppure "Come creare moduli lazy loaded"?