

Routing

<https://angular.dev/guide/routing>

Routing

Kas yra URL?

`https://www.lrt.lt/lietuva/aktualijos/straipsnis?geras=true&data=siandien#antraste`

The diagram illustrates the components of the URL `https://www.lrt.lt/lietuva/aktualijos/straipsnis?geras=true&data=siandien#antraste` using hand-drawn brackets and labels below the text:

- schema**: A bracket under `https://`.
- domain**: A bracket under `www.lrt.lt`.
- path parameters**: A bracket under `/lietuva/aktualijos/straipsnis`. Above this bracket, three double-headed arrows indicate the separation of the path segments: `/lietuva/`, `aktualijos/`, and `straipsnis`.
- search/query parameters**: A bracket under `?geras=true&data=siandien`. Above this bracket, two double-headed arrows indicate the separation of the query parameters: `geras=true` and `data=siandien`.
- fragment**: A bracket under `#antraste`.

Routing

Konfigūracija

- Kiekvienas route - **vienas arba daugiau URL path parametru**.
- Standartiškai aprašoma **routes.ts** faile.
- Paprastas **Route** objektų masyvas.
- Naudojama aplikacijos app.config.ts faile per **provideRouter**.
- Galimi parametrizuoti route'ai, kurie leidžia parametrus naudoti kaip kintamuosius.

```
export const routes: Routes = [  
  {  
    path: '',  
    component: MainComponent,  
  },  
  {  
    path: 'task/:id',  
    component: TaskComponent,  
  },  
];
```

Routing

Krovimas pareikalavus (Lazy routing/Lazy loaded routes)

- `load*` (**`loadChildren`**, **`loadComponent`**) sintaksė leidžia krauti puslapio Javascript tik tada, kai ten naviguojama.
- Leidžia atskirti rutinimo konfigūracijas skirtingoms puslapių medžio šakoms.

```
export const routes: Routes = [  
  {  
    path: '',  
    loadComponent: () =>  
      import('./main/main.component').then((c) => c.MainComponent),  
  },  
  {  
    path: 'task/:id',  
    loadComponent: () =>  
      import('./task/task.component').then((c) => c.TaskComponent),  
  },  
];
```

Routing

Route guards

```
export const falseGuard: CanActivateFn = (route, state) => {  
  return false;  
};
```

- Servisai, blokuojantys prieėjimą prie puslapių pagal sąlygas.
- Skirtingi metodai blokuoja skirtingus navigacijos momentus:
 - **canActivate** - tikrina, ar galima įeiti į route.
 - **canMatch** - tikrina, ar galima “surasti” route.
 - **canActivateChild** - tikrina, ar galima įeiti į route vaiką.

Užduotis #2 – Sukurti šias puslapių struktūras:

1. Pradinis puslapis („“ maršrutas)

- Sukurkite pagrindinį puslapį, kuriame bus pasveikinimas
- Šiame puslapyje gali būti pasveikinimo pranešimas, informacija apie užduočių valdymo sistemą arba nuoroda į užduočių sąrašą.

2. Užduočių sąrašo puslapis („tasks“ maršrutas)

- Sukurkite puslapį, kuriame bus rodomas visų užduočių sąrašas.
- Kiekviena užduotis turi turėti nuorodą, vedančią į jos atskirą puslapį. Puslapyje gaukite info apie užduotį iš TaskService.

3. Kiekvienos užduoties atskiras puslapis („tasks/:id“ maršrutas)

- Sukurkite dinaminį puslapį, kuris atvaizduos vienos konkrečios užduoties informaciją.
- Informacija gali apimti užduoties aprašymą, sukūrimo datą, būseną, galutinį terminą ir galimybę ją redaguoti ar ištrinti.

4. Navigacijos juosta

- Pridėkite viršuje esantį navigacijos meniu, kuris leis vartotojui lengvai pereiti tarp skirtingų puslapių.
- Navigacijoje turėtų būti bent šios nuorodos:
 - Pradžia („Pagrindinis“)
 - Užduočių sąrašas („Užduotys“)

Lifto muzika.

