

# MUTIPLE PAGE ROUTING



# IMPORT COMPONENTS :

## *GLOBALY OR LOCALY !*

### Global import

main.js

```
import BaseCard
from './components/UI/BaseCard.vue';

app.component('base-card', BaseCard );
```

Every component  
Can use base-card

### Local import

myComponent.vue

```
<script>
import BaseCard
from './components/UI/BaseCard.vue';

export default {
  components: { BaseCard }
}
```

Only myComponent  
Can use base-card

**WHY  
THIS  
CLASS ?**



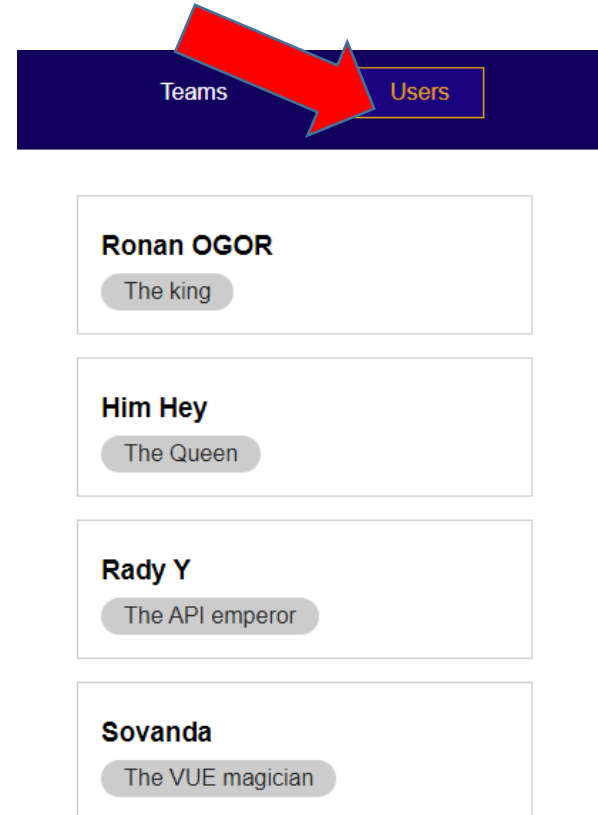
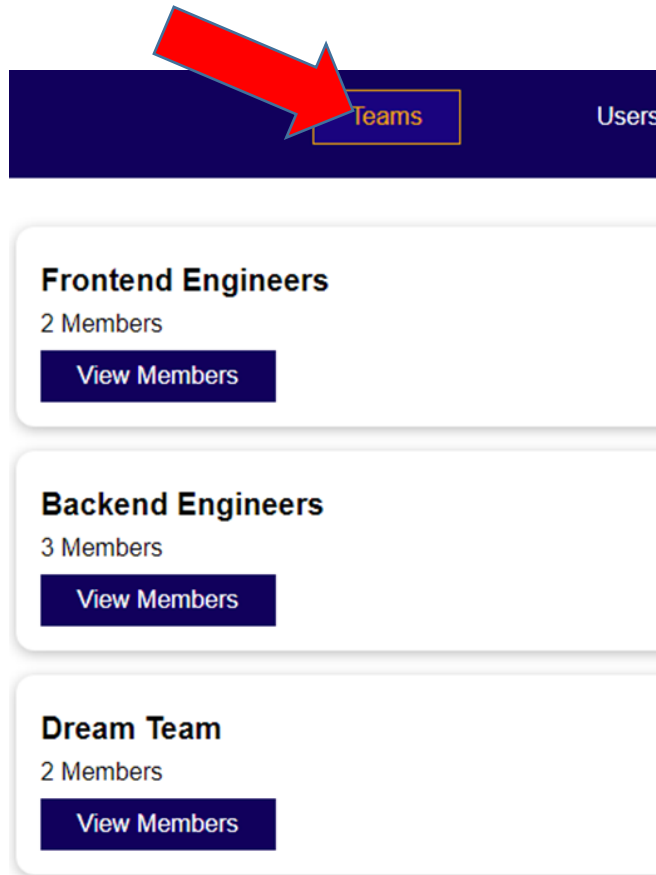
# ACTIVITY 1



10 MIN

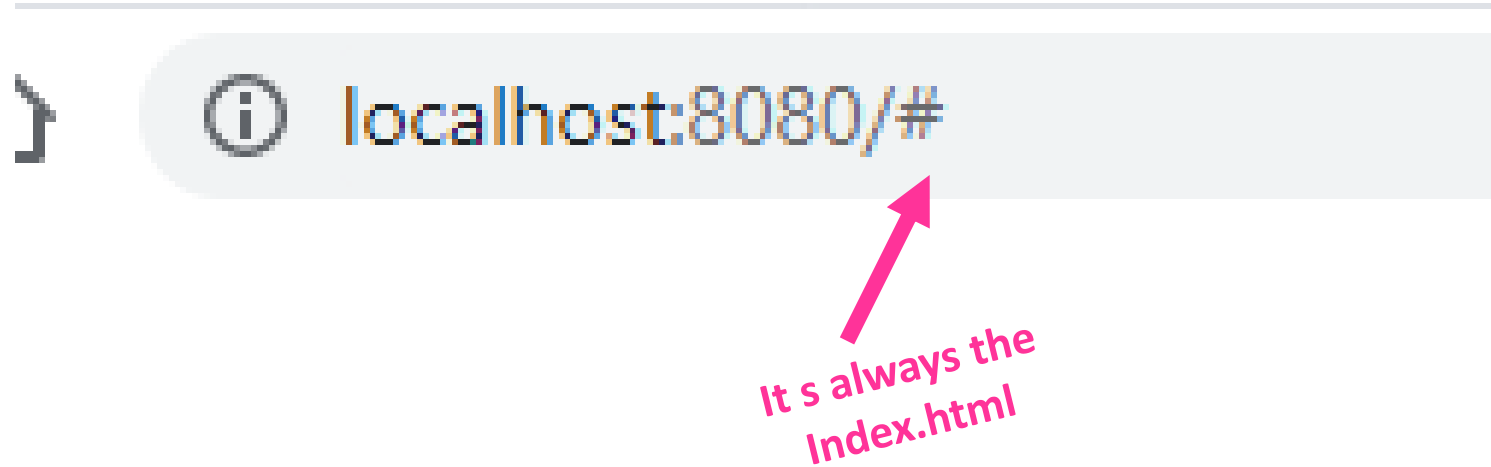
1. Open and execute the project
2. **Understand** how the MENU work
3. Does the **URL change** when you change the view ?

On click :  
show  
only the  
team-list  
compone  
nt



On click : show only  
the user-list  
component

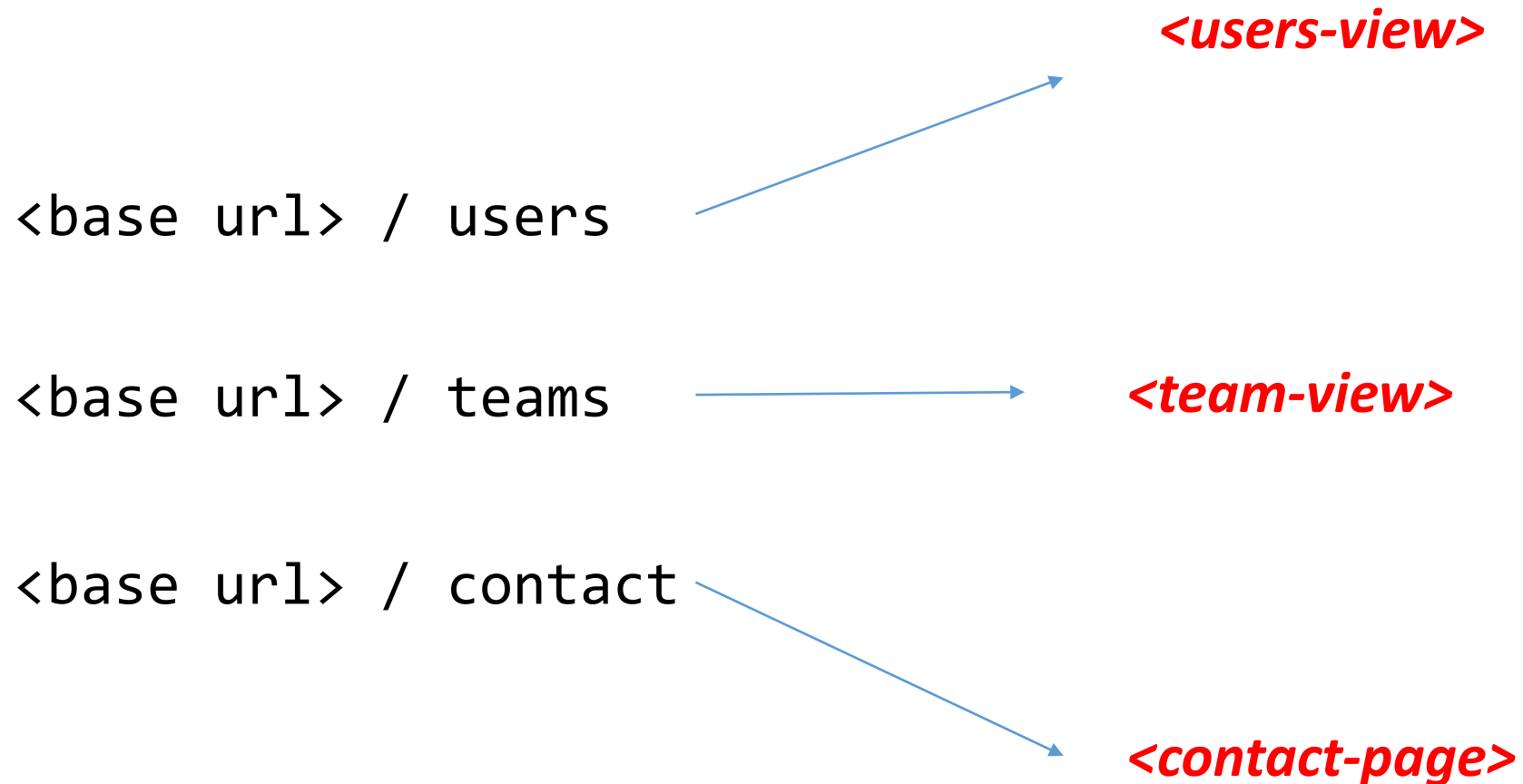
Because we build a **SPA**, the URL never changes :



But with this approach:

- We can just share the default page URL
- **We cannot share the URL of a specific page**

But we can manage specific URL using the **VUE**  
**ROUTER !**

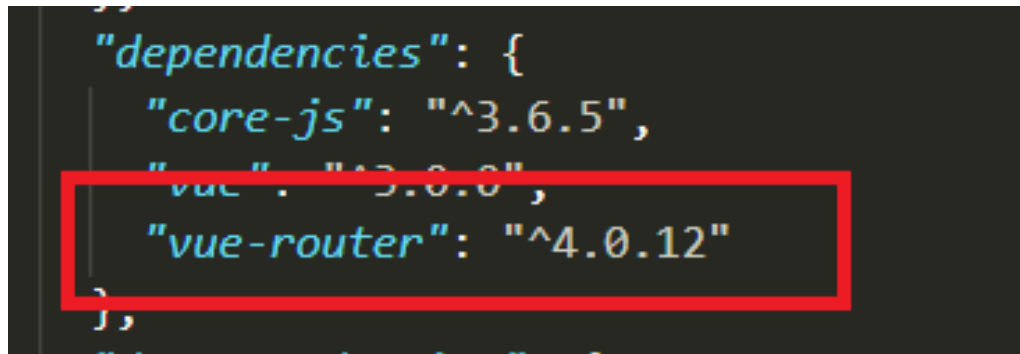


# Configuration of router

#1 Install the VUE ROUTER using NPM

```
npm install vue-router@next
```

#2 You should the router on package.json

A screenshot of a code editor showing the 'dependencies' section of a package.json file. The text is as follows: 

```
"dependencies": {  
  "core-js": "^3.6.5",  
  "vue": "^3.0.0",  
  "vue-router": "^4.0.12"  
},
```

 A red rectangular box highlights the line `"vue-router": "^4.0.12"`.



## DEMO 1 *router and <router-view> !*

```
import { createApp } from 'vue';
import { createRouter, createWebHistory } from 'vue-router';

import App from './App.vue';
import TeamsList from './components/teams/TeamsList.vue';
import UsersList from './components/users/UsersList.vue';

const router = createRouter({
  history: createWebHistory(),
  routes: [
    { path: '/teams', component: TeamsList }, // our-domain.co
    { path: '/users', component: UsersList },
  ]
});

const app = createApp(App)
```



# ***ROUTERS: HOW DO IT WORKS ?***

1 – Setup the router on main.ts

```
const router = createRouter({  
  history: createWebHistory(),  
  routes: [  
    { path: '/teams', component: TeamsList },  
    { path: '/users', component: UsersList },  
  ]  
});
```



Link URL paths  
To  
Components

2 – Anywhere you need, use the `<router-view>` component to render the current routed component

```
<template>  
  <router-view></router-view>  
</template>
```



*You don't need to register the component related to the routing any more!*



## DEMO 2 *navigation using <router-link>*

```
<header>
  <nav>
    <ul>
      <li>
        <router-link to="/teams">Teams</router-link>
      </li>
      <li>
        <router-link to="/users">Users</router-link>
      </li>
    </ul>
  </nav>
</header>
```

# ***HOW DOES **ROUTER-LINK** WORKS ?***

Router-link are like <a> elements, linking to a path of the ROUTER :

```
<router-link to="/teams">Teams</router-link>
```



On click

```
<router-view></router-view>
```

The router-view is now on the teamList component



## DEMO 4 *Navigation programmatically*

```
methods: {  
  confirmInput() {  
    this.$router.push('/teams');  
  }  
}
```

# HOW DOES *ROUTER PUSH* WORKS ?

You also can programmatically change the active route:

```
methods: {  
  confirmInput() {  
    this.$router.push('/teams');  
  }  
}
```



```
<router-view></router-view>
```

The router-view is now on the teamList component



## DEMO 5 *Path with parameters*

```
const router = createRouter({
  history: createWebHistory(),
  routes: [
    { path: '/teams', component: TeamsList },
    { path: '/users', component: UsersList },

    { path: '/teams/:teamId', component: TeamMembers }
  ],
  linkActiveClass: 'active'
});
```

# HOW DOES *ROUTE WITH PAMETERS* WORKS ?

1- Define the route with some **parameters** :

```
{ path: '/teams/:teamId', component: TeamMembers }
```

2- The parameter can be retrieved using the **\$route parameters**:

```
const teamId = this.$route.params.teamId;
```

3- You can also get your **teamID** as a **prop** :

```
{ path: '/teams/:teamId', component: TeamMembers, props :true }
```



## ACTIVITY 2



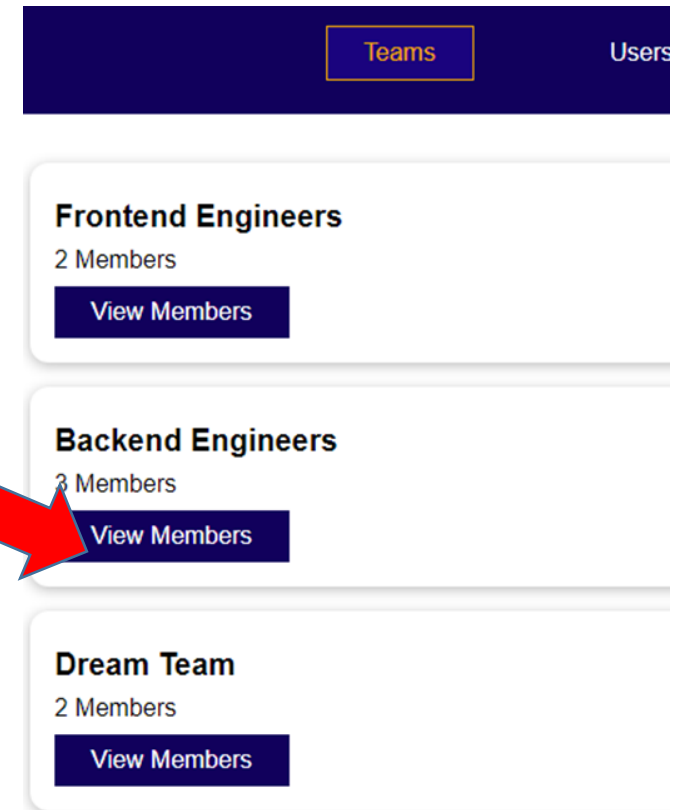
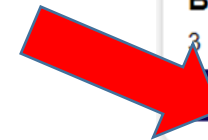
10 MIN

When we click on the View member, we want to route to `/teams/:teamID`  
*To see the list of members of the team related to the team ID*

→ Try to do this using the router link component :

```
<router-link to:”/yourPath” >
```

On click





# ***ROUTERS: A LOT MORE !!!!***

```
routes: [  
  { path: '/:notFound(.*)', component: NotFound }  
  
  { path: '/', redirect: '/teams' },  
  {  
    name: 'teams',  
    path: '/teams',  
    meta: { needsAuth: true },  
    components: { default: TeamsList, footer: TeamsFooter },  
    beforeEnter(to, from, next) {  
      console.log('users beforeEnter');  
      console.log(to, from);  
      next();  
    }  
  }  
]
```

Route to a not found component

Redirect the path

Require authentication for this path

Provide a footer

Do something before entering to The component