

PREGRADO



UNIDAD 2 | FRONTEND WEB APPLICATIONS

VUE IN-APP NAVIGATION

Al finalizar la unidad de aprendizaje, el estudiante ejecuta un proceso de ingeniería de software para crear una solución de aplicación frontend, utilizando el lenguaje JavaScript y un framework de actualidad, integrada con RESTful web services, con una Arquitectura Orientada a Servicios, en un ambiente de desarrollo ágil y colaborativo, comunicando resultados y proceso de ingeniería aplicado, bajo un enfoque dirigido por la innovación e inclusión.

AGENDA

INTRO

DYNAMIC ROUTES

ROUTE NAVIGATION

NAMED ROUTES & VIEWS

REDIRECT & ALIAS



Vue Router

Módulo responsable de in-app navigation support en Vue.

Algunos features:

Route /View mapping.

Component-based router configuration.

Route params, query, wildcards.

Install

Vue CLI

```
vue add router
```

NPM

```
npm install vue-router@4
```

Migrating from Vue 2

<https://router.vuejs.org/guide/migration/index.html>

Define routes

Each route should map to a component.

```
const routes = [  
  { path: '/foo', component: Foo },  
  { path: '/bar', component: Bar }  
]
```

root instance

Create router instance and pass routes option.

Create root instance, injecting router.

```
const router = new VueRouter({  
  routes // short for `routes: routes`  
})
```

```
const app = new Vue({  
  router  
}).$mount('#app')
```

\$router

Acceso a router vía this.\$router.

```
// Home.vue
export default {
  methods: {
    goBack() {
      window.history.length > 1 ? this.$router.go(-1) : this.$router.push('/')
    }
  }
}
```


\$route

Acceso a ruta actual vía `this.$route`.

```
// Home.vue
export default {
  computed: {
    username() {
      return this.$route.params.username
    }
  },
  methods: {
    goBack() {
      window.history.length > 1 ? this.$router.go(-1) : this.$router.push('/')
    }
  }
}
```

<router-link> and <router-view>

router-link permite especificar una ruta inscrita en routes.

router-view especifica dónde se hará render del componente asociado.

```
<script src="https://unpkg.com/vue/dist/vue.js"></script>
<script src="https://unpkg.com/vue-router/dist/vue-router.js"></script>

<div id="app">
  <h1>Hello App!</h1>
  <p>
    <!-- use router-link component for navigation. -->
    <!-- specify the link by passing the `to` prop. -->
    <!-- `<router-link>` will be rendered as an `<a>` tag by default -->
    <router-link to="/foo">Go to Foo</router-link>
    <router-link to="/bar">Go to Bar</router-link>
  </p>
  <!-- route outlet -->
  <!-- component matched by the route will render here -->
  <router-view></router-view>
</div>
```

AGENDA

INTRO

DYNAMIC ROUTES

ROUTE NAVIGATION

NAMED ROUTES & VIEWS

REDIRECT & ALIAS



Dynamic route matching

Se puede establecer patrones o segmentos dinámico en los path de rutas.

```
const router = new VueRouter({  
  routes: [  
    // dynamic segments start with a colon  
    { path: '/users', component: Users },  
    { path: '/users/:id', component: User }  
  ]  
})
```

Dinamic route matching

Cuando el path concuerda con una ruta, el valor dinámico se expone vía **this.\$route.params** en el componente.

```
<div>User {{ $route.params.id }}</div>
```

Dynamic route matching

Pueden existir varios segmentos dinámicos en una ruta. Ellos se mapean hacia campos en `$route.params`.

Pattern	Matched path	<code>\$route.params</code>
<code>/users/:username</code>	<code>/users/evan</code>	<code>{ username: 'evan' }</code>
<code>/users/:username/posts/:post_id</code>	<code>/users/evan/posts/123</code>	<code>{ username: 'evan', post_id: '123' }</code>

Catch all / 404 Not found Route

Se puede utilizar el asterisco (*) como comodín para asociar rutas que concuerden con un patrón de ruta o cualquier ruta en general.

```
{  
  // will match everything  
  path: '*'  
}  
{  
  // will match anything starting with `/user-`  
  path: '/user-*'  
}
```

Catch all / 404 Not found Route

Cuando se utiliza asterisco, se adiciona a `$route.params` un parámetro con nombre `pathMatch`, conteniendo el resto del url que concuerda con el asterisco.

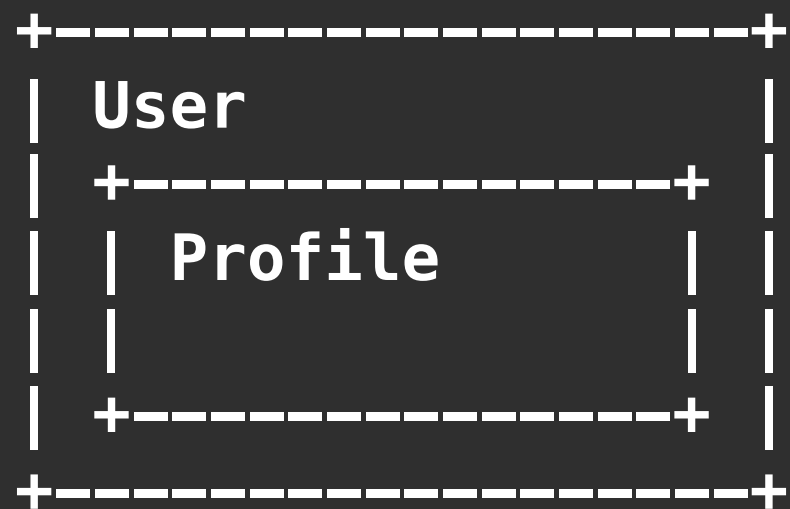
```
// Given a route { path: '/user-*' }  
this.$router.push('/user-admin')  
this.$route.params.pathMatch // 'admin'
```

```
// Given a route { path: '*' }  
this.$router.push('/non-existing')  
this.$route.params.pathMatch // '/non-existing'
```


Nested routes

Algunos componentes se pueden anidar en varios niveles.

/users/foo/profile



/users/foo/posts



```
<div class="user">
  <h2>User {{ $route.params.id }}</h2>
  <router-view></router-view>
</div>
```

Nested routes

Algunos componentes se pueden anidar en varios niveles.

```
const router = new VueRouter({
  routes: [
    { path: '/users', component: Users,
    { path: '/users/:id', component: User,
      children: [
        {
          // UserHome will be rendered inside User's <router-view>
          // when /user/:id is matched
          path: '', component: UserHome
        },
        {
          // UserProfile will be rendered inside User's <router-view>
          // when /users/:id/profile is matched
          path: 'profile',
          component: UserProfile
        },
        {
          // UserPosts will be rendered inside User's <router-view>
          // when /users/:id/posts is matched
          path: 'posts',
          component: UserPosts
        }
      ]
    }
  ]
})
```

AGENDA

INTRO

DYNAMIC ROUTES

ROUTE NAVIGATION

NAMED ROUTES & VIEWS

REDIRECT & ALIAS



router.push(location, onComplete?, onAbort?)

Navegar hacia otro URL.

Puede usar también `this.$router.push`.

```
// literal string path  
router.push('home')
```

```
// object  
router.push({ path: 'home' })
```

```
const userId = '123'
```

```
router.push({ name: 'users', params: { userId } }) // -> /user/123
```

```
router.push({ path: `/users/${userId}` }) // -> /user/123
```

AGENDA

INTRO

DYNAMIC ROUTES

ROUTE NAVIGATION

NAMED ROUTES & VIEWS

REDIRECT & ALIAS



Named Routes

Puede asignarse nombre a una ruta al momento de crear la instancia de Router.

```
const router = new VueRouter({  
  routes: [  
    {  
      path: '/users/:userId',  
      name: 'user',  
      component: User  
    }  
  ]  
})
```


Named Views

Permite mostrar múltiples views a la vez. En vez de un **router-outlet**, puede colocarse varios **router-view** con nombre.

Debe usarse **components** al declarar la ruta.

```
const router = new VueRouter({
  routes: [
    {
      path: '/',
      components: { default: Base, sidebar: Sidebar, main: Main }
    }
  ]
})
```

```
<router-view class="view base"></router-view>
<router-view class="view sidebar" name="sidebar"></router-view>
<router-view class="view main" name="main"></router-view>
```

AGENDA

INTRO

DYNAMIC ROUTES

ROUTE NAVIGATION

NAMED ROUTES & VIEWS

REDIRECT & ALIAS



Redirect

Puede especificarse redirecciones, hacia otra ruta, named route o function.

```
const router = new VueRouter({
  routes: [
    { path: '/home', name: 'home', component: Home },
    { path: '/', redirect: '/home' },
    { path: '/c', redirect: { name: 'foo' } },
    { path: '/d', redirect: to => {
      // the function receives the target route as the argument
      // return redirect path/location here.
    } }
  ]
})
```

Alias

Permite hacer render al componente asociado a la ruta original, aunque en el path se visualice el alias.

```
const router = new VueRouter({  
  routes: [  
    { path: '/home', component: Home, alias: '/start' }  
  ]  
})
```

RESUMEN

Recordemos

VueRouter brinda soporte a Vue para el manejo de rutas que permitan implementar in-app navigation.

Las rutas pueden ser estáticas o dinámicas. La navegación puede ser declarada en el template o ser programática en la sección de scripting del componente.



REFERENCIAS

Para profundizar

<https://router.vuejs.org/>



PREGRADO

Ingeniería de Software



UPC

Prolongación Primavera 2390,
Monterrico, Santiago de Surco
Lima 33 - Perú
T 511 313 3333
<https://www.upc.edu.pe>

exígete, innova