



SPA ROUTING DEEP DIVE WITH VUE ROUTER



@posva

Saint Petersburg 24 May 2019

EDUARDO SAN MARTIN MOROTE

 Vue core team

 Freelance

 Paris

 Malaga 



 @posva

 @posva

<> Code

! Issues 116

🔗 Pull requests 55

📁 Projects 1

📊 Insights

⚙️ Settings

 The official router for Vue.js. <http://router.vuejs.org/>

Edit

vue

javascript

router

Manage topics

🔄 881 commits

🔗 14 branches

🏷️ 56 releases

🚀 1 environment

👤 176 contributors

📄 MIT

Branch: dev ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



posva chore(deps): upgrade typescript

Latest commit 65b2f45 a day ago

📁 .circleci	chore(ci): use yarn.lock for checksum	8 months ago
📁 .github	update issue template	2 years ago
📁 build	refactor(changelog): move release script to scripts folder	2 months ago
📁 dist	[build] 3.0.2	3 months ago
📁 docs-gitbook	docs: use string params in examples (#2269)	2 months ago
📁 docs	docs(ru): Translation update (#2599)	10 days ago
📁 examples	chore(examples): add missing semicolon in CSS file from transitions e...	5 months ago
📁 flow	chore(flow): Add link exact active class to router options (#2236)	8 months ago



Routing in your App
after a year

Routing in you App
the 1st week

WHAT IS AN
SPARROUTER?

HISTORY

HTML5 history
Hash based history
Abstract (SSR, no url)
Phone



ROUTER

Creating routes
Matching logic
Lazy loading
Guards
Public API

COMPONENTES

View
Link

THREE KIND OF ROUTERS

- Imperative
- Declarative
- Configuration-based

<> Code

! Issues 70

🔗 Pull requests 26

📁 Projects 1

📊 Insights

Micro client-side router inspired by the Express router <http://visionmedia.github.com/page.js>

🔄 636 commits

🔗 14 branches

🏷️ 37 releases

👤 83 contributors

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



matthewpblog 1.11.4

Latest commit 19f60bc 10 days ago

📁 examples	Update partials example to work with relative URL	a year ago
📁 test	Add coverage for non-URL supporting browsers	4 months ago
📄 .gitignore	Allow development in more recent Node versions	a year ago
📄 .jshintrc	Add page.create() to allow multiple pages	5 months ago
📄 .travis.yml	Remove last test	4 months ago
📄 History.md	Update History.md	a year ago
📄 Makefile	Adds an ES module	5 months ago
📄 Readme.md	Export and document clickHandler	4 months ago

CREATING ROUTES IN PAGE.JS

```
page('/users', (context, next) => {  
  // change users page  
  next()  
})  
  
// applying multiple middlewares  
page('/users/:id/edit', checkUser, userProfileEdit)
```

NAVIGATING IN PAGE.JS

```
page('/') // go to home  
page('/users/2/edit') // go to user edition with id
```

PAGE.JS

SIMPLE & FLEXIBLE
BUT VERBOSE

- 👉 Imperative
- ✅ Programmatic navigation
- ❌ Declarative navigation
- ✅ Navigation Guards
- ❌ Dynamic routing (add/remove routes)

https://reach.tech/router

197 commits 1 branch 29 releases 19 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

Table with 3 columns: File/Folder Name, Description, and Time Ago. Includes entries for .github, build, examples, scripts, src, website, .eslintrc, and .gitignore.

DYNAMIC ROUTING

```
router.addRoute('/some-route', options)  
router.removeRoute('/some-route')
```

⚠ not actual API

REACH ROUTER

```
<Router>
  <Home path="/" />
  <UserProfileEdit path="/users/:id/edit" />
</Router>
```

```
<div>
  <Link to="/">Home</Link>
  <Link to={` /users/${this.user.id}/edit`} />
</div>
```

REACH ROUTER

IDIOMATIC FOR REACT

- 👉 Declarative
- ⚠️ Programmatic navigation (in-jsx)
- ✅ Declarative navigation
- ❌ Navigation Guards
- ✅ Dynamic routing (add/remove routes)

<> Code

! Issues 116

🔗 Pull requests 55

📁 Projects 1

📊 Insights

⚙️ Settings

🚦 The official router for Vue.js. <http://router.vuejs.org/>

Edit

vue

javascript

router

Manage topics

🔄 881 commits

🔗 14 branches

🏷️ 56 releases

🚀 1 environment

👤 176 contributors

📄 MIT

Branch: dev

New pull request

Create new file

Upload files

Find file

Clone or download



posva chore(deps): upgrade typescript

Latest commit 65b2f45 a day ago

📁 .circleci	chore(ci): use yarn.lock for checksum	8 months ago
📁 .github	update issue template	2 years ago
📁 build	refactor(changelog): move release script to scripts folder	2 months ago
📁 dist	[build] 3.0.2	3 months ago
📁 docs-gitbook	docs: use string params in examples (#2269)	2 months ago
📁 docs	docs(ru): Translation update (#2599)	10 days ago
📁 examples	chore(examples): add missing semicolon in CSS file from transitions e...	5 months ago
📁 flow	chore(flow): Add link exact active class to router options (#2236)	8 months ago

VUE ROUTER

```
const router = new Router({
  mode: 'history',
  routes: [
    { path: '/', component: Home },
    { path: '/users/:id', component: UserProfile },
  ]
})
```

```
<div>
  <router-link to="/">Home</router-link>
  <router-link :to="`/users/${this.user.id}`">
    My Profile
  </router-link>
</div>
```

VUE ROUTER

DECOUPLED: ROUTER INSTANCE / COMPONENTS

- 👉 Configuration-based
- ✅ Programmatic navigation
- ✅ Declarative navigation
- ✅ Navigation Guards
- ⚠️ Dynamic routing (add/remove routes)

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



COMPONENTS



ROUTER



HISTORY

```
router.push('/search')
```

- Store visited URLs
- JS ↔ URL
 - `push()` `replace()`, ...
 - `listen()`

 HISTORY

- Route matching
 - `match()`
 - `resolve()`
- Navigation
 - `currentRoute`
 - `push()` `replace()`, ...
 - `beforeEach()`, ...
- Creating routes
 - `new Router({ routes })`
 - `addRoutes()`



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

```
<router-link to="/">Home</router-link>
```



COMPONENTES

THE GOOD PARTS



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



Router Instance

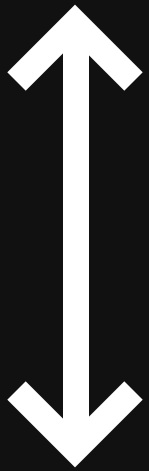
- Dynamically render current view
- Pass params as props



router-view


```
<router-link to="/">Home</router-link>
```

```
<router-link :to="{ name: 'User', params: {id: '2'}}">.
```



- Resolve target location
- Render an anchor tag with link
- Handles click event
- Applies active classes

Router Instance



router-link

`this.$route`

```
{  
  path: '/users/2',  
  name: 'UserProfile',  
  query: {},  
  params: { id: '2' },  
  meta: {}  
}
```

this.\$route

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

```
created () {  
  fetch(`/api/users/${this.$route.params.id}`)  
    .then(/* ... */)  
}
```

this.\$router

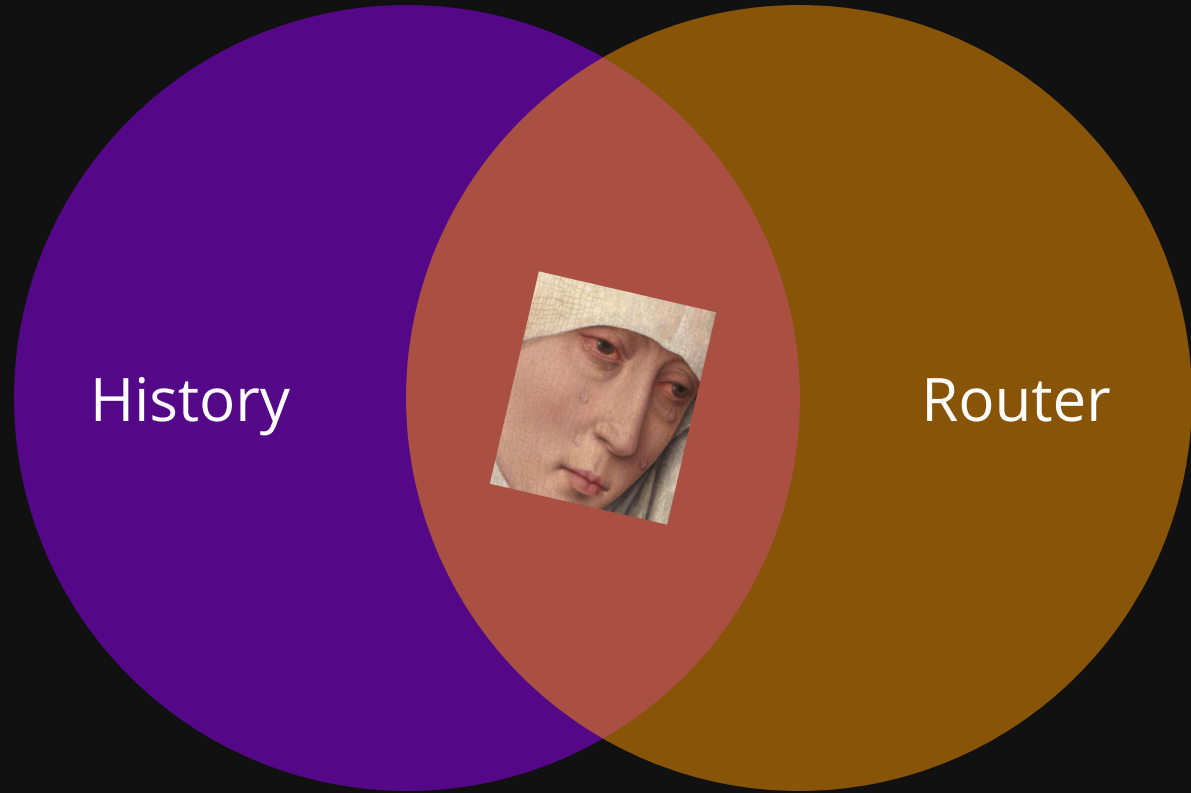
```
const router = new Router({
  mode: 'history',
  routes: [
    { path: '/', component: Home },
    { path: '/users/:id', component: UserProfile },
  ]
})
```

```
created () {
  if (someCondition) {
    this.$router.push('/other-route')
  }
}
```

THE UGLY PARTS



NO CLEAR DIVISION OF RESPONSIBILITIES



GLOBAL GUARDS

```
router.beforeEach((to, from, next) => {  
  // verify roles based on `to`  
  // ...  
  
  // we can only call `next` once  
  if (!isLoggedIn) next('/login') // redirect to login  
  else if (!isAuthorized) next(false) // abort  
  else next() // allow navigation  
})  
  
router.afterEach((to, from) => {  
  // Analytics, etc  
  // cannot modify navigation anymore  
})
```

router.beforeEach

```
function beforeEach(guard: NavigationGuard): ListenerRemover {  
  this.beforeGuards.push(guard)  
  return () => {  
    const i = this.beforeGuards.indexOf(guard)  
    if (i > -1) this.beforeGuards.splice(i, 1)  
  }  
}
```


router.push

```
function push (location, onComplete, onAbort) {  
  this.history.push(location, onComplete, onAbort)  
}
```

history.push

```
1 function push (location, onComplete, onAbort) {
2   const route = this.router.match(location, this.currentLocation)
3
4   try {
5     // run navigation guards queue
6     // ...
7
8     // change the url in the browser (HTML5)
9     window.history.pushState({}, '', route.fullPath)
10    onComplete(route)
11  } catch (error) {
12    // handle the error
13    // ...
14    onAbort(error)
15  }
16 }
```

 Simplified version

HISTORY

- Base: +300 LoC
- HTML5: 70 LoC
- Hash: 130 LoC

 Complex codebase

ROUTER

- Router Class: ~200 LoC
- Matcher: ~180 LoC

 Simple codebase

NO CLEAR DIVISION OF RESPONSIBILITIES

- Harder to fix bugs
- Harder to add features
- Harder to contribute
- Harder to extend

- Store visited URLs
- JS ↔ URL
 - `push()` `replace()`, ...
 - `listen()`



HISTORY

- Route matching
 - `match()`
 - `resolve()`
- Navigation
 - `currentRoute`
 - `push()` `replace()`, ...
 - `beforeEach()`, ...
- Creating routes
 - `new Router({ routes })`
 - `addRoutes()`



ROUTER

API

- `push()` `replace()`
- `listen()`
- URL parsing

Responsibilities / Expectations

- Modify the Location
- Parses URL
 - path
 - query
 - hash
- Notifies when Location changes
- Handles Encoding problems
- Can be overloaded



HISTORY



HISTORY BASE =

~10 LOC + ~150 LOC PARSING



HISTORY HTML5 =

~200 LOC



HTML5 PUSH

```
1 push(to: HistoryLocation, data?: HistoryState) {
2   const normalized = this.utils.normalizeLocation(to)
3   // replace current entry state to add the forward value
4   this.history.replaceState({
5     ...this.history.state,
6     forward: normalized
7   },
8   ''
9  )
10
11  const state = {
12    back: this.location,
13    current: normalized,
14    forward: null,
15    replaced: false
16    ...data,
17  }
18
```




HTML5 POPSTATE LISTENER

```
1 private setupPopStateListener() {
2     const handler: PopStateListener = ({ state }: { state: StateEntry }) => {
3         const from = this.location
4         // save the location where we are going or build it
5         // from window.location
6         this.location = state ? state.current : buildFullPath()
7
8         // call all listeners
9         const navigationInfo = {
10            direction:
11                state.forward && from.fullPath === state.forward.fullPath
12                    ? NavigationDirection.back
13                    : NavigationDirection.forward,
14        }
15        this._listeners.forEach(listener =>
16            listener(this.location, from, navigationInfo)
17        )
18    }
19
```

BROWSER QUIRKS

vue-router/push-state.js

```
40 export function pushState (url?: string, replace?: boolean) {
41   saveScrollPosition()
42   // try...catch the pushState call to get around Safari
43   // DOM Exception 18 where it limits to 100 pushState calls
44   const history = window.history
45   try {
46     if (replace) {
47       history.replaceState({ key: _key }, '', url)
48     } else {
49       _key = genKey()
50       history.pushState({ key: _key }, '', url)
51     }
52   } catch (e) {
53     window.location[replace ? 'replace' : 'assign'](url)
54   }
55 }
```

reach-router/history.js

```
45 navigate(to, { state, replace = false } = {}) {
46   state = { ...state, key: Date.now() + "" };
47   // try...catch iOS Safari limits to 100 pushState calls
48   try {
49     if (transitioning || replace) {
50       source.history.replaceState(state, null, to);
51     } else {
52       source.history.pushState(state, null, to);
53     }
54   } catch (e) {
55     source.location[replace ? "replace" : "assign"](to);
56   }
```

URI ENCODING ISSUES

Directly navigate to `/é?é=é#é` `history.pushState({}, '', '/é?é=é#é')`



- URL Bar: `/é?é=é#é`
- `location.pathname`: `'/%C3%A9'`
- `location.search`: `'%C3%A9=%C3%A9'`
- `location.search`: `'#%C3%A9'`

- URL Bar: `/é?é=é#é`
- `location.pathname`: `'/é'`
- `location.search`: `'é=é'`
- `location.search`: `'#é'`

- Route matching
 - `resolve()`
- Adding Route Records
 - `addRouteRecord`
 - `removeRouteRecord`
- Navigation
 - `currentRoute`
 - `push()` `replace()`, ...
- Navigation Guards
 - `beforeEach()`, ...
- Dynamic Routing
 - `addRoute` / `removeRoute`



MATCHER



ROUTER



ROUTER

API

- Route matching
 - `resolve()`
- Adding Route Records
 - `addRouteRecord`
 - `removeRouteRecord`

Responsibilities / Expectations

- Resolving a Router Location to a Route Record
- Only handles the path of URL
- Handles priority of Route Records
- Parses/handle params



MATCHER



ADD ROUTE RECORD

```
1 function addRouteRecord(  
2   record: RouteRecord,  
3   parent?: RouteMatcher  
4 ): void {  
5   // create the matcher, link it to parent  
6   const matcher = createMatcher(record, parent)  
7   if (parent) parent.children.push(matcher)  
8  
9   // handle nested routes  
10  if ('children' in record) {  
11    for (const childRecord of record.children) {  
12      this.addRouteRecord(childRecord, matcher)  
13    }  
14  }  
15  
16  this.insertMatcher(matcher)  
17 }
```

- and finally wildcard segments get a 1 point penalty

Here's a "table" showing different paths and their scores:

Score	Path
5	/
7	/groups
13	/groups/:groupId
14	/groups/mine
19	/groups/:groupId/users/*
20	/groups/:groupId/users
21	/groups/mine/users
24	/:one/:two/:three/:four
26	/groups/:groupId/users/:userId
27	/groups/:groupId/users/me
28	/groups/mine/users/me
30	/:one/:two/:three/:four/:five

REACH ROUTER RANKING

- Each segment gets 4 points and then...
- Static segments get 3 more points
- Dynamic segments 2 more
- Root segments 1
- and finally wildcard segments get a 1 point penalty

API

- Navigation
 - `currentRoute`
 - `push()` `replace()`
- Navigation Guards
 - `beforeEach()`
- Dynamic Routing
- Lazy loading Pages
- In-component Guards
- Error handlers

Responsibilities / Expectations

- Async navigation
- Trigger Navigation guards
- Expose current Route Location
- Handle redirections



ROUTER



PUSH / REPLACE

```
1  async function push(to: RouteLocation): Promise<RouteLocationNormalized> {
2    // match the location
3    const toLocation = this.resolveLocation(to, this.currentRoute)
4
5    // navigate, handle guards
6    try {
7      await this.navigate(toLocation, this.currentRoute)
8    } catch (error) {
9      if (error instanceof NavigationGuardRedirect) {
10         // trigger the whole navigation again
11         return this.push(error.to)
12       } else {
13         throw error
14       }
15     }
16
17     // change the URL
18     if (to.replace === true) this.history.replace(toLocation)
```



RESOLVE LOCATION

```
1 function resolveLocation(  
2   location: MatcherLocation,  
3   currentLocation: MatcherLocationNormalized,  
4   redirectedFrom?: MatcherLocationNormalized  
5 ): MatcherLocationNormalized {  
6   // resolve against the matcher  
7   const matchedRoute = this.matcher.resolve(location, currentLocation)  
8  
9   // handles in-record redirects  
10  if ('redirect' in matchedRoute) {  
11    const { redirect, normalizedLocation } = matchedRoute  
12    // match the redirect instead  
13    return this.resolveLocation(  
14      this.history.utils.normalizeLocation(redirect),  
15      currentLocation,  
16      // pass down the location we tried to navigate to  
17      normalizedLocation  
18    )
```



NAVIGATE

```
1 async function push(to: RouteLocation): Promise<RouteLocationNormalized> {
2   // match the location
3   const toLocation = this.resolveLocation(to, this.currentRoute)
4
5   // navigate, handle guards
6   try {
7     await this.navigate(toLocation, this.currentRoute)
8   } catch (error) {
9     if (error instanceof NavigationGuardRedirect) {
10      // trigger the whole navigation again
11      return this.push(error.to)
12    } else {
13      throw error
14    }
15  }
16
17  // change the URL
18  if (to.replace === true) this.history.replace(toLocation)
```

NAVIGATION GUARDS



GLOBAL GUARDS

```
router.beforeEach((to, from, next) => {  
  // verify roles based on `to`  
  // ...  
  
  // we can only call `next` once  
  if (!isLoggedIn) next('/login') // redirect to login  
  else if (!isAuthorized) next(false) // abort  
  else next() // allow navigation  
})
```

PER-ROUTE GUARDS

```
1 {
2   path: '/admin',
3   component: AdminPanel,
4   beforeEnter (to, from, next) {
5     if (isAdmin) next()
6     else next(false)
7   },
8 }
```

IN-COMPONENT GUARDS

```
1 export default {
2   name: 'AdminPanel',
3
4   data: () => ({ adminInfo: null }),
5
6   async beforeRouteEnter (to, from, next) {
7     const adminInfo = await getAdminInfo()
8     next(vm => {
9       vm.adminInfo = adminInfo
10    })
11  },
12 }
```

/posts → /admin

next()



beforeEach



next(false)



beforeEnter



/posts



beforeRouteEnter

Admin.vue

STATIC TO LAZY IN VUE

```
import Calendar from '@components/Calendar.vue'
```



```
const Calendar = () => import('@components/Calendar.vue')
```

next ()

/posts → /admin



beforeEnter



beforeRouteEnter



```
await Admin()
```



NAVIGATE

```
1 async function navigate(  
2   to: RouteLocationNormalized,  
3   from: RouteLocationNormalized  
4 ): Promise<void> {  
5   let guards: Array<() => Promise<any>>  
6  
7   // check beforeRouteLeave guards  
8   guards = extractComponentsGuards(  
9     from.matched.filter(record => to.matched.indexOf(record) < 0).reverse(),  
10    'beforeRouteLeave',  
11    to,  
12    from  
13  )  
14  
15  // run the queue of per beforeRouteLeave  
16  for (const guard of guards) await guard()  
17  
18  // check global guards beforeEach
```



EXTRACT COMPONENTS GUARDS

```
1 type GuardType = 'beforeRouteEnter' | 'beforeRouteUpdate' | 'beforeRouteLeave'
2 export async function extractComponentsGuards(
3   matched: MatchedRouteRecord[],
4   guardType: GuardType,
5   to: RouteLocationNormalized,
6   from: RouteLocationNormalized
7 ): Array<() => Promise<void>> {
8   return matched.map(record => {
9     const { component } = record
10    return async () => {
11      // resolve async components from cache
12      const resolvedComponent = await resolveComponent(component)
13
14      // await the guard if it exists
15      const guard = resolvedComponent[guardType]
16      if (guard) {
17        await guardToPromiseFn(guard, to, from)()
18      }
19    }
20  })
21 }
```



ROUTER



HISTORY



MATCHER

- Try to equally distribute code complexity **and** size
- Use Typescript
- Refactor often
- Tests in many browsers
- Make it easier to contribute



[PATREON.COM/POSVA](https://patreon.com/posva)

PATREONS 🙌

СПАСИБО! 🖐️



 @posva

 @posva