EDUARDO
SAN MARTIN MOROTE

🌍 Vue core team
👨‍💻 Freelance
📍 Paris
🏠 Malaga 🇪🇸

@posva
@posva
The official router for Vue.js.  [http://router.vuejs.org/](http://router.vuejs.org/)

- **881** commits
- **14** branches
- **56** releases
- **1** environment
- **176** contributors
- [MIT](https://choosealicense.com/licenses/mit/)

<table>
<thead>
<tr>
<th>Repository Details</th>
<th>Latest Commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>posva chore(deps): upgrade typescript</td>
<td>65b2f45 a day ago</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>.circleci</td>
<td>chore(ci): use yarn.lock for checksum</td>
<td>8 months ago</td>
</tr>
<tr>
<td>.github</td>
<td>update issue template</td>
<td>2 years ago</td>
</tr>
<tr>
<td>build</td>
<td>refactor(changelog): move release script to scripts folder</td>
<td>2 months ago</td>
</tr>
<tr>
<td>dist</td>
<td>[build] 3.0.2</td>
<td>3 months ago</td>
</tr>
<tr>
<td>docs-gitbook</td>
<td>docs: use string params in examples (#2269)</td>
<td>2 months ago</td>
</tr>
<tr>
<td>docs</td>
<td>docs(ru): Translation update (#2599)</td>
<td>10 days ago</td>
</tr>
<tr>
<td>examples</td>
<td>chore(examples): add missing semicolon in CSS file from transitions e...</td>
<td>5 months ago</td>
</tr>
<tr>
<td>flow</td>
<td>chore(flow): Add link exact active class to router options (#2236)</td>
<td>8 months ago</td>
</tr>
</tbody>
</table>
Routing in your App after a year

Routing in your App the 1st week
WHAT IS AN SPA ROUTER?
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
Micro client-side router inspired by the Express router  
http://visionmedia.github.com/page.js

<table>
<thead>
<tr>
<th>Branch: master</th>
<th>New pull request</th>
</tr>
</thead>
<tbody>
<tr>
<td>636 commits</td>
<td>14 branches</td>
</tr>
<tr>
<td>37 releases</td>
<td>83 contributors</td>
</tr>
</tbody>
</table>

Latest commit 19f60bc 10 days ago

- **examples**: Update partials example to work with relative URL  
  - File: 1.11.4, Date: a year ago

- **test**: Add coverage for non-URL supporting browsers  
  - Date: 4 months ago

- **.gitignore**: Allow development in more recent Node versions  
  - Date: a year ago

- **.jshintrc**: Add page.create() to allow multiple pages  
  - Date: 5 months ago

- **.travis.yml**: Remove last test  
  - Date: 4 months ago

- **History.md**: Update History.md  
  - Date: a year ago

- **Makefile**: Adds an ES module  
  - Date: 5 months ago

- **Readme.md**: Export and document clickHandler  
  - Date: 4 months ago
CREATING ROUTES IN PAGE.JS

```javascript
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

드리미메 기능
- Imperative
- Programmatic navigation
- Navigation Guards
- Dynamic routing (add/remove routes)

드라마기능
- Declarative navigation
<table>
<thead>
<tr>
<th>File</th>
<th>Message</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>.github</td>
<td>Added spectrum links</td>
<td>8 months ago</td>
</tr>
<tr>
<td>build</td>
<td>Prettier all the things</td>
<td>8 months ago</td>
</tr>
<tr>
<td>examples</td>
<td>Add history actions “PUSH” and “POP”</td>
<td>5 months ago</td>
</tr>
<tr>
<td>scripts</td>
<td>word</td>
<td>11 months ago</td>
</tr>
<tr>
<td>src</td>
<td>No longer using this (see parent commit)</td>
<td>5 months ago</td>
</tr>
<tr>
<td>website</td>
<td>Add missing word to &quot;Data Loading&quot; sentence.</td>
<td>7 months ago</td>
</tr>
<tr>
<td>.eslintrc</td>
<td>Update .eslintrc as do not ignore children prop types</td>
<td>8 months ago</td>
</tr>
<tr>
<td>.gitignore</td>
<td>make a compat build</td>
<td>10 months ago</td>
</tr>
</tbody>
</table>
DYNAMIC ROUTING

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

⚠ not actual API
<Router>
  <Home path="/">
  </Home>
  <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)
The official router for Vue.js. [http://router.vuejs.org/](http://router.vuejs.org/)

- vue
- javascript
- router

**Commit Activity**

- 881 commits
- 14 branches
- 56 releases
- 1 environment
- 176 contributors
- MIT License

**Files ChANGED**

- `.circleci` chore(deps): upgrade typescript
- `.github` chore(deps): upgrade typescript
- `build` chore(deps): upgrade typescript
- `dist` chore(deps): upgrade typescript
- `docs-gitbook` docs: use string params in examples (#2269)
- `docs` docs(ru): Translation update (#2599)
- `examples` chore(examples): add missing semicolon in CSS file from transitions e...
- `flow` chore(flow): Add link exact active class to router options (#2236)

Latest commit: 65b2145 a day ago
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/>

```
router.push('/search')
```
• Store visited URLs
• JS ↔ URL
  ▪ push() replace(), ...
  ▪ listen()
• Route matching
  ▪ match()
  ▪ resolve()

• Navigation
  ▪ currentRoute
  ▪ push() replace(), ...
  ▪ beforeEach(), ...

• Creating routes
  ▪ new Router({ routes })
  ▪ addRoutes()
<router-view/>

<router-link to="/">Home</router-link>
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
this.$route

{
    path: '/users/2',
    name: 'UserProfile',
    query: {},
    params: { id: '2' },
    meta: {}
}
<p>User: {{ $route.params.id }}</p>

created () {
  fetch(`/api/users/${this.$route.params.id}`)
    .then(/* ... */)
}
const router = new Router({
  mode: 'history',
  routes: [
    { path: '/', component: Home },
    { path: '/users/:id', component: UserProfile },
  ]
})

created () {
  if (someCondition) {
    this.$router.push('/other-route')
  }
}
NO CLEAR DIVISION OF RESPONSIBILITIES
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
})
```typescript
function beforeEach(guard: NavigationGuard): ListenerRemover {
  this.beforeGuards.push(guard)
  return () => {
    const i = this.beforeGuards.indexOf(guard)
    if (i > -1) this.beforeGuards.splice(i, 1)
  }
}
```
function push (location, onComplete, onAbort) {
  this.history.push(location, onComplete, onAbort)
}
function push (location, onComplete, onAbort) {
    const route = this.router.match(location, this.currentLocation)

    try {
        // run navigation guards queue
        // ...

        // change the url in the browser (HTML5)
        window.history.pushState({}, '', route.fullPath)
        onComplete(route)
    } catch (error) {
        // handle the error
        // ...
        onAbort(error)
    }
}
HISTORY

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

 Antaroshi: 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()`

- Route matching
  - `match()`
  - `resolve()`

- Navigation
  - `currentRoute`
  - `push()`, `replace()`, ...
  - `beforeEach()`, ...

- Creating routes
  - `new Router({ routes })`
  - `addRoutes()`
**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 BASE = ~10 LOC + ~150 LOC PARSING

HISTORY HTML5 = ~200 LOC
push(to: HistoryLocation, data?: HistoryState) {
    const normalized = this.utils.normalizeLocation(to)
    // replace current entry state to add the forward value
    this.history.replaceState({
        ...this.history.state,
        forward: normalized
    }, '');
}

const state = {
    back: this.location,
    current: normalized,
    forward: null,
    replaced: false
    ...data,
};
```typescript
private setupPopStateListener() {
  const handler: PopStateListener = ({ state }: { state: StateEntry }) => {
    const from = this.location
    // save the location where we are going or build it
    // from window.location
    this.location = state ? state.current : buildFullPath()
    // call all listeners
    const navigationInfo = {
      direction:
      state.forward && from.fullPath === state.forward.fullPath
        ? NavigationDirection.back
        : NavigationDirection.forward,
    }
    this._listeners.forEach(listener =>
      listener(this.location, from, navigationInfo)
    )
  }
```

BROWSER QUIRKS

vue-router/push-state.js

```javascript
export function pushState (url?: string, replace?: boolean) {
    saveScrollPosition()
    // try...catch the pushState call to get around Safari
    // DOM Exception 18 where it limits to 100 pushState calls
    const history = window.history
    try {
        if (replace) {
            history.replaceState({ key: _key }, '')
        } else {
            _key = genKey()
            history.pushState({ key: _key }, '', u)
        }
        catch (e) {
            window.location[replace ? 'replace' : 'assign'](to);
        }
    }
}
```

reach-router/history.js

```javascript
navigate(to, { state, replace = false } = {})
    state = { ...state, key: Date.now() + '' }
    // try...catch iOS Safari limits to 100 pushState calls
    try {
        if (transitioning || replace) {
            source.history.replaceState(state, null, to);
        } else {
            source.history.pushState(state, null, to);
        } catch (e) {
            source.location[replace ? "replace" : "assign"](to);
        }
    }
```
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
**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
function addRouteRecord(
    record: RouteRecord,
    parent?: RouteMatcher
): void {
  // create the matcher, link it to parent
  const matcher = createMatcher(record, parent)
  if (parent) parent.children.push(matcher)

  // handle nested routes
  if ('children' in record) {
    for (const childRecord of record.children) {
      this.addRouteRecord(childRecord, matcher)
    }
  }

  this.insertMatcher(matcher)
}
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

Here’s a “table” showing different paths and their scores:

<table>
<thead>
<tr>
<th>Score</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>/</td>
</tr>
<tr>
<td>7</td>
<td>/groups</td>
</tr>
<tr>
<td>13</td>
<td>/groups/:groupId</td>
</tr>
<tr>
<td>14</td>
<td>/groups/mine</td>
</tr>
<tr>
<td>19</td>
<td>/groups/:groupId/users/*</td>
</tr>
<tr>
<td>20</td>
<td>/groups/:groupId/users</td>
</tr>
<tr>
<td>21</td>
<td>/groups/mine/users</td>
</tr>
<tr>
<td>24</td>
<td>/one/:two/:three/:four</td>
</tr>
<tr>
<td>26</td>
<td>/groups/:groupId/users/:userId</td>
</tr>
<tr>
<td>27</td>
<td>/groups/:groupId/users/me</td>
</tr>
<tr>
<td>28</td>
<td>/groups/mine/users/me</td>
</tr>
<tr>
<td>30</td>
<td>/one/:two/:three/:four/:five</td>
</tr>
</tbody>
</table>
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
async function push(to: RouteLocation): Promise<RouteLocationNormalized> {
  // match the location
  const toLocation = this.resolveLocation(to, this.currentRoute)

  // navigate, handle guards
  try {
    await this.navigate(toLocation, this.currentRoute)
  } catch (error) {
    if (error instanceof NavigationGuardRedirect) {
      // trigger the whole navigation again
      return this.push(error.to)
    } else {
      throw error
    }
  }

  // change the URL
  if (to.replace === true) this.history.replace(toLocation)
function resolveLocation(
    location: MatcherLocation,
    currentLocation: MatcherLocationNormalized,
    redirectedFrom?: MatcherLocationNormalized
) : MatcherLocationNormalized {
    // resolve against the matcher
    const matchedRoute = this.matcher.resolve(location, currentLocation)
    // handles in-record redirects
    if ('redirect' in matchedRoute) {
        const { redirect, normalizedLocation } = matchedRoute
        // match the redirect instead
        return this.resolveLocation(
            this.history.utils.normalizeLocation(redirect),
            currentLocation,
            // pass down the location we tried to navigate to
            normalizedLocation
        )
    }
}
async function push(to: RouteLocation): Promise<RouteLocationNormalized> {
  // match the location
  const toLocation = this.resolveLocation(to, this.currentRoute)

  // navigate, handle guards
  try {
    await this.navigate(toLocation, this.currentRoute)
  } catch (error) {
    if (error instanceof NavigationGuardRedirect) {
      // trigger the whole navigation again
      return this.push(error.to)
    } else {
      throw error
    }
  }

  // change the URL
  if (to.replace === true) this.history.replace(toLocation)
NAVIGATION
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

```javascript
{
    path: '/admin',
    component: AdminPanel,
    beforeEnter (to, from, next) {
        if (isAdmin) next()
        else next(false)
    },
}
```
async beforeRouteEnter (to, from, next) {
    const adminInfo = await getAdminInfo()
    next(vm => {
        vm.adminInfo = adminInfo
    })
}
`/posts` ➡ `/admin`

- `beforeEach`
- `next()`

- `beforeEnter`
- `next(false)`

- `beforeRouteEnter`
  - Admin.vue

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

const Calendar = () => import('@/components/Calendar.vue')
/posts ➡ /admin

beforeEnter

await Admin()

beforeRouteEnter

next()
async function navigate(
  to: RouteLocationNormalized,
  from: RouteLocationNormalized
): Promise<void> {
  let guards: Array<() => Promise<any>>

  // check beforeRouteLeave guards
  guards = extractComponentsGuards(
    from.matched.filter(record => to.matched.indexOf(record) < 0).reverse(),
    'beforeRouteLeave',
    to,
    from
  )

  // run the queue of per beforeRouteLeave
  for (const guard of guards) await guard()

  // check global guards beforeEach
  // check beforeRouteLeave guards
  // run the queue of per beforeRouteLeave
  // check global guards beforeEach
}

// check beforeRouteLeave guards
// run the queue of per beforeRouteLeave
// check global guards beforeEach

type GuardType = 'beforeRouteEnter' | 'beforeRouteUpdate' | 'beforeRouteLeave'

export async function extractComponentsGuards(
    matched: MatchedRouteRecord[],
    guardType: GuardType,
    to: RouteLocationNormalized,
    from: RouteLocationNormalized
) : Array<() => Promise<void>> {
    return matched.map(record => {
        const { component } = record
        return async () => {
            // resolve async components from cache
            const resolvedComponent = await resolveComponent(component)
            // await the guard if it exists
            const guard = resolvedComponent[guardType]
            if (guard) {
                await guardToPromiseFn(guard, to, from)()
            }
        }
    })
}
• Try to equally distribute code complexity **and** size
• Use Typescript
• Refactor often
• Tests in many browsers
• Make it easier to contribute
СПАСИБО! 👋

@posva

@posva