

Internationalisation

La langue par défaut d'Element est le Chinois. Si vous souhaitez utiliser une autre langue, il vous faudra la configurer. Par exemple, dans votre fichier d'entrée, si vous importez Element entièrement:

```
import Vue from 'vue'
import ElementUI from 'element-ui'
import locale from 'element-ui/lib/locale/lang/en'

Vue.use(ElementUI, { locale })
```

Ou si vous importez Element à la demande:

```
import Vue from 'vue'
import { Button, Select } from 'element-ui'
import lang from 'element-ui/lib/locale/lang/en'
import locale from 'element-ui/lib/locale'

// configure la langue
locale.use(lang)

// importe les composants
Vue.component(Button.name, Button)
Vue.component(Select.name, Select)
```

La pack de la langue Chinois est importé par défaut, même si vous configurez une autre langue. En utilisant le `NormalModuleReplacementPlugin` fournit par webpack vous pouvez remplacer la locale par défaut:

webpack.config.js

```
{
  plugins: [
    new webpack.NormalModuleReplacementPlugin(/element-
ui[\\/]lib[\\/]locale[\\/]lang[\\/]zh-CN/, 'element-ui/lib/locale/lang/en')
  ]
}
```

Compatible avec `vue-i18n@5.x`

Element est compatible avec [vue-i18n@5.x](#), ce qui rend le changement de langue encore plus simple.

```
import Vue from 'vue'
import VueI18n from 'vue-i18n'
import Element from 'element-ui'
import enLocale from 'element-ui/lib/locale/lang/en'
import zhLocale from 'element-ui/lib/locale/lang/zh-CN'

Vue.use(VueI18n)
Vue.use(Element)
```

```
Vue.config.lang = 'zh-cn'
Vue.locale('zh-cn', zhLocale)
Vue.locale('en', enLocale)
```

Compatible avec d'autres plugins i18n

Element n'est pas forcément compatible avec d'autres plugins i18n que vue-i18n, mais vous pouvez configurer le fonctionnement i18n.

```
import Vue from 'vue'
import Element from 'element-ui'
import enLocale from 'element-ui/lib/locale/lang/en'
import zhLocale from 'element-ui/lib/locale/lang/zh-CN'

Vue.use(Element, {
  i18n: function (path, options) {
    // ...
  }
})
```

Compatible avec `vue-i18n@6.x`

Vous devrez le configurer manuellement pour la compatibilité avec `6.x`.

```
import Vue from 'vue'
import Element from 'element-ui'
import VueI18n from 'vue-i18n'
import enLocale from 'element-ui/lib/locale/lang/en'
import zhLocale from 'element-ui/lib/locale/lang/zh-CN'

Vue.use(VueI18n)

const messages = {
  en: {
    message: 'hello',
    ...enLocale // Ou utilisez `Object.assign({ message: 'hello' }, enLocale)`
  },
  zh: {
    message: '你好',
    ...zhLocale // Ou utilisez `Object.assign({ message: '你好' }, zhLocale)`
  }
}

// Crée l'instance de VueI18n avec ses options
const i18n = new VueI18n({
  locale: 'en', // indique la locale
  messages, // indique les messages
})
```

```
Vue.use(Element, {
  i18n: (key, value) => i18n.t(key, value)
})

new Vue({ i18n }).$mount('#app')
```

i18n personnalisée dans les composants à la demande

```
import Vue from 'vue'
import DatePicker from 'element/lib/date-picker'
import VueI18n from 'vue-i18n'

import enLocale from 'element-ui/lib/locale/lang/en'
import zhLocale from 'element-ui/lib/locale/lang/zh-CN'
import ElementLocale from 'element-ui/lib/locale'

Vue.use(VueI18n)
Vue.use(DatePicker)

const messages = {
  en: {
    message: 'hello',
    ...enLocale
  },
  zh: {
    message: '你好',
    ...zhLocale
  }
}

// Crée l'instance de VueI18n avec ses options
const i18n = new VueI18n({
  locale: 'en', // indique la locale
  messages, // indique les messages
})

ElementLocale.i18n((key, value) => i18n.t(key, value))
```

Import via un CDN

```
<script src="//unpkg.com/vue"></script>
<script src="//unpkg.com/element-ui"></script>
<script src="//unpkg.com/element-ui/lib/umd/locale/en.js"></script>

<script>
  ELEMENT.locale(ELEMENT.lang.en)
</script>
```

Compatible avec `vue-i18n`

```
<script src="//unpkg.com/vue"></script>
<script src="//unpkg.com/vue-i18n/dist/vue-i18n.js"></script>
<script src="//unpkg.com/element-ui"></script>
<script src="//unpkg.com/element-ui/lib/umd/locale/zh-CN.js"></script>
<script src="//unpkg.com/element-ui/lib/umd/locale/en.js"></script>

<script>
  Vue.locale('en', ELEMENT.lang.en)
  Vue.locale('zh-cn', ELEMENT.lang.zhCN)
</script>
```

Actuellement, Element supporte les langues suivantes:

- Chinois simplifié (zh-CN)
- Anglais (en)
- Allemand (de)
- Portugais (pt)
- Espagnol (es)
- Danois (da)
- Français (fr)
- Norvégien (nb-NO)
- Chinois traditionnel (zh-TW)
- Italien (it)
- Coréen (ko)
- Japonais (ja)
- Néerlandais (nl)
- Vietnamien (vi)
- Russe (ru-RU)
- Turquie (tr-TR)
- Portugais brésilien (pt-br)
- Farsi (fa)
- Thaï (th)
- Indonésien (id)
- Bulgare (bg)
- Polonais (pl)
- Finnois (fi)
- Suédois (sv-SE)
- Grec (el)
- Slovaque (sk)
- Catalan (ca)
- Tchèque (cs-CZ)
- Ukrainien (ua)
- Turkmène (tk)
- Tamoul (ta)
- Letton (lv)
- Afrikaans (af-ZA)
- Estonien (ee)
- Slovène (sl)
- Arabe (ar)

- Hébreu (he)
- Lituanien (lt)
- Mongol (mn)
- Kazakh (kz)
- Hongrois (hu)
- Roumain (ro)
- Kurde (ku)
- Ouïghour (ug-CN)
- Khmer (km)
- Serbe (sr)
- Basque (eu)
- Kirghize (kg)
- Croate (hr)
- Arménien (hy)
- Espéranto (eo)

Si votre langue n'apparaît pas dans la liste, n'hésitez pas à contribuer: ajoutez simplement un fichier de configuration [ici](#) et créez une pull request.