



Vue.js 2 Training Course

by Blair Lee



Blair — vue create vue-example — vue — node /usr/local/bin/vue create vue-example — 80x24

Vue CLI v4.5.13

? Please pick a preset:

Default ([Vue 2] babel, eslint)

Default (Vue 3) ([Vue 3] babel, eslint)

❯ Manually select features



Blair — vue create vue-example — vue — node /usr/local/bin/vue create vue-example — 80x24

Vue CLI v4.5.13

? Please pick a preset: Manually select features

? Check the features needed for your project:

- Choose Vue version
- Babel
- TypeScript
- Progressive Web App (PWA) Support
- Router
- Vuex
- CSS Pre-processors
- Linter / Formatter
- Unit Testing
- E2E Testing



Blair — vue create vue-example — vue — node /usr/local/bin/vue create vue-example — 80x24

Vue CLI v4.5.13

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel, Router,
  CSS Pre-processors, Linter
? Choose a version of Vue.js that you want to start the project with (Use arrow
keys)
❯ 2.x
  3.x
```

Blair — vue create vue-example — vue — node /usr/local/bin/vue create vue-example — 80x24

Vue CLI v4.5.13

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel, Router,
  CSS Pre-processors, Linter
? Choose a version of Vue.js that you want to start the project with 2.x
? Use history mode for router? (Requires proper server setup for index fallback
[in production) Yes
? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported
by default): Sass/SCSS (with node-sass)
? Pick a linter / formatter config: Basic
? Pick additional lint features: Lint on save
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config
  files
? Save this as a preset for future projects? (y/N) N
```



Blair — Blair@MacBook-Pro — ~ — -zsh — 80x24

```
success Saved lockfile.  
✨ Done in 126.20s.  
🚀 Invoking generators...  
📦 Installing additional dependencies...
```

```
yarn install v1.13.0  
[1/4] 🔎 Resolving packages...  
[2/4] 🚚 Fetching packages...  
[3/4] 🔗 Linking dependencies...  
[4/4] 🏭 Building fresh packages...  
success Saved lockfile.  
✨ Done in 45.05s.  
⚓ Running completion hooks...
```

```
📄 Generating README.md...  
🎉 Successfully created project vue-example.  
👉 Get started with the following commands:
```

```
$ cd vue-example  
$ yarn serve
```



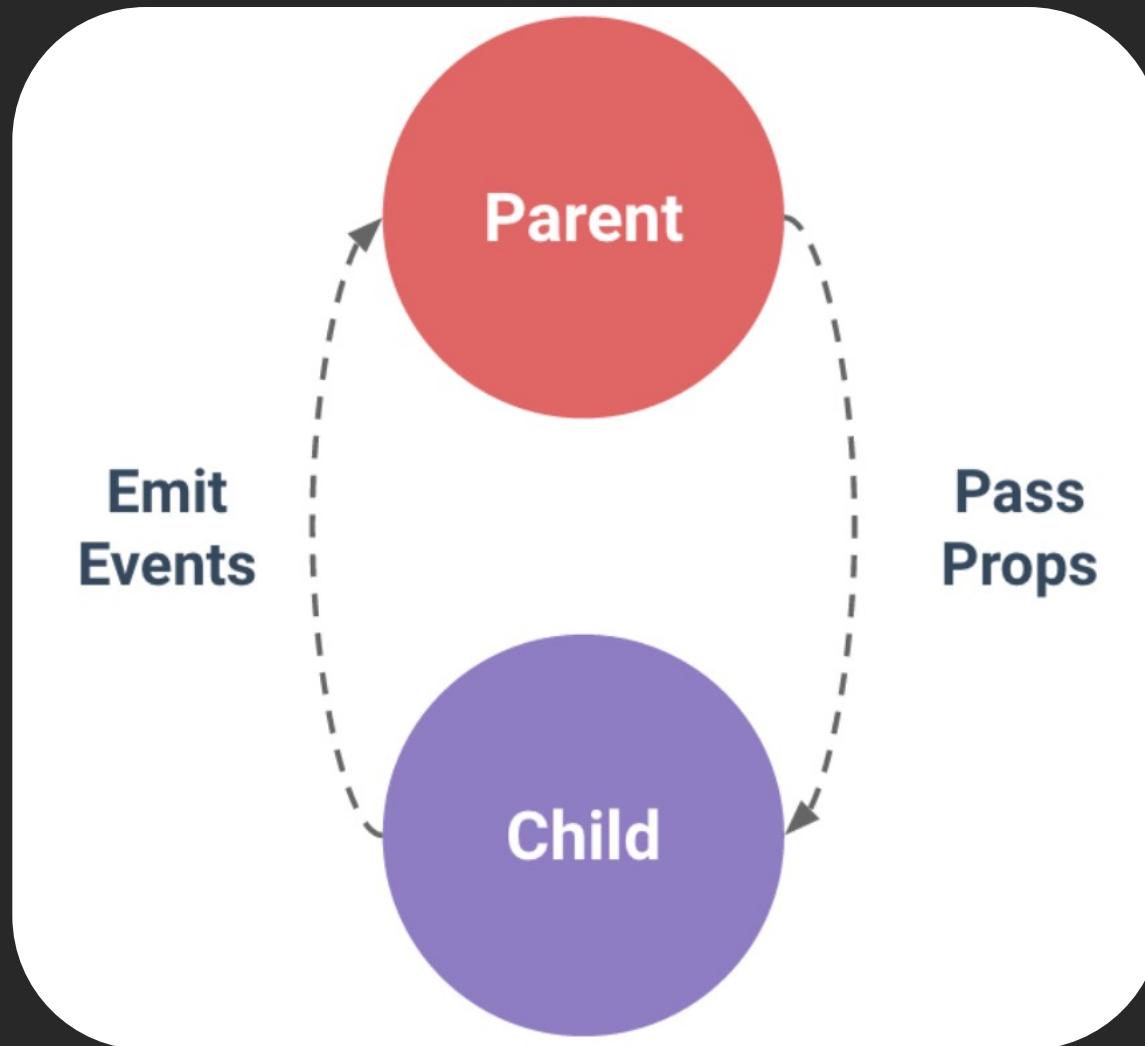
Vue Options

```
1 var vm = new Vue({  
2   el: "#app",           // 用來 mount Vue 實體  
3   data: { },            // 要綁定的資料  
4   props: { },           // 接收外部資料的屬性  
5   methods: { },         // 定義 Vue 實體內使用的函式  
6   watch: { },           // 觀察實體內資料變動  
7   computed: { },         // 自動計算的屬性  
8   template: "...",      // 提供 Vue 變更後的模板  
9   component: { }        // 定義元件  
10});
```

常見的幾種 Component 間的資料溝通傳遞方式

- ✓ **props** Parent -> Child
- ✓ **\$emit** : Child -> Parent
- **\$attr / \$listeners** : 多層 component 間的屬性傳遞
- **provide / inject** : 依賴注入
- **Vuex** : 狀態管理

props in, events out



通訊錄 Wireframe

Contacts.

Add

No	Name	Phone
----	------	-------

1	A	0912345678
2	B	0912345679

共 2 筆

Contact

Add

No	Name
----	------

1	A
2	B

共 2 筆

Add.

X

Name

Phone

Save

通訊錄 Wireframe

Contact.vue Page

Header.vue Component

Contacts.

Add

No	Name	Phone
1	A	0912345678
2	B	0912345679

Table.vue Component

共 2 筆

Modal.vue Component

Contact

Add.

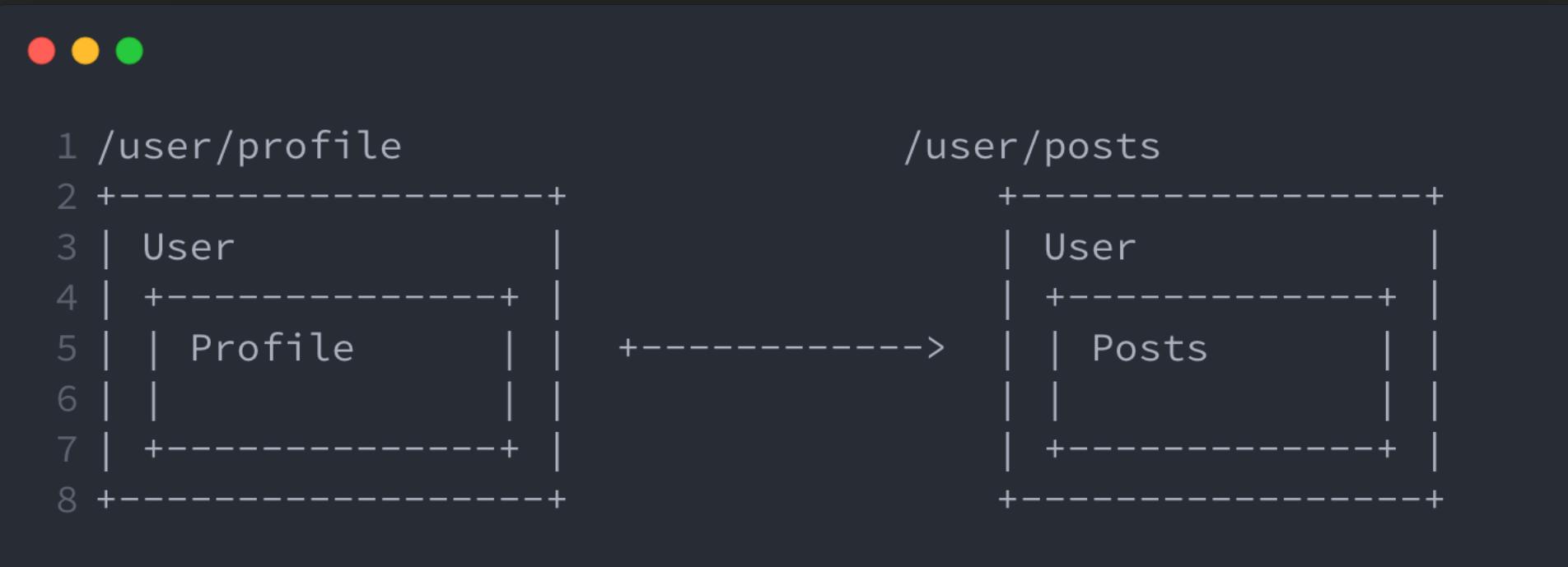
X

Name

Phone

Save

Nested Route



Nested Route



```
1 const router = new VueRouter({
2   routes: [
3     {
4       path: '/user',
5       component: User,
6       children: [
7         {
8           path: 'profile',
9           component: UserProfile
10      },
11      {
12        path: 'posts',
13        component: UserPosts
14      }
15    ]
16  }
17 ]
18 })
```



```
1 // User Component
2 <template>
3   <div>
4     ...
5     <router-view />
6   </div>
7 </template>
8
```