



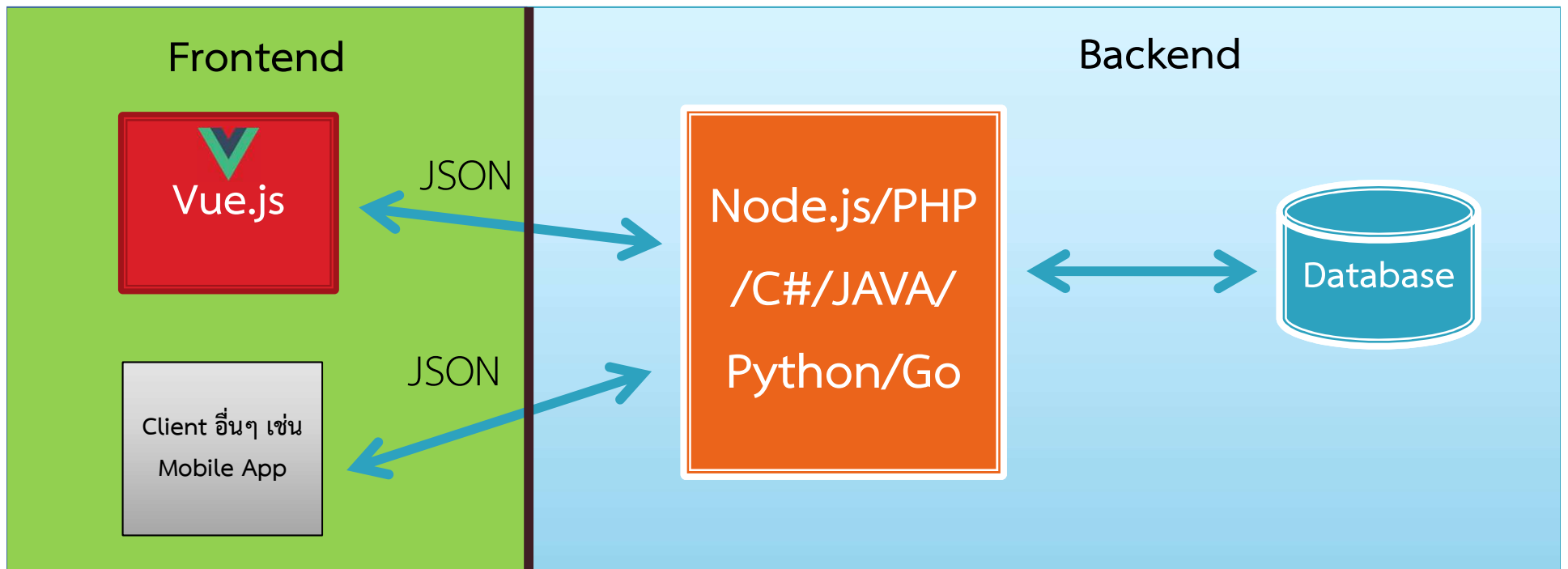
หลักสูตรออนไลน์ "พัฒนา Modern Web App ด้วย Vue.js 3 และ Composition API"

โค้ชเอก

codingthailand.com

ภาพใหญ่ของคอร์สนี้

- ▶ เขียน Frontend เป็น Web Application ด้วย Vue.js



โปรแกรมที่ต้องติดตั้งก่อน

- ▶ Git <https://git-scm.com/>
- ▶ Node.js <https://nodejs.org/en/>
- ▶ Visual Studio Code <https://code.visualstudio.com/>
- ▶ Postman <https://www.postman.com/downloads/>
- ▶ Vue 3 Devtools <https://chrome.google.com/webstore/detail/vuejs-devtools/ljjemllljcmogpfapbkkighbhppjdbg/related>



การตั้งค่า Git (ทำครั้งเดียวต่อเครื่อง)

- ▶ เปิด Command Prompt พิมพ์คำสั่ง
- ▶ `git config --global user.name "akenarin komkoon"`
- ▶ `git config --global user.email "codingthailand@gmail.com"`



ตรวจสอบโปรแกรม และเครื่องมือต่างๆ

- ▶ `node -v`
- ▶ `npm -v`
- ▶ `git --version`



ติดตั้ง VS Code Extensions

- ▶ Vue VSCode Extension Pack
- ▶ Vue Peek
- ▶ Vue 3 Snippets



ตั้งค่า prettier extensions ที่ไฟล์ settings.json

```
"[javascript]": {  
  "editor.formatOnSave": false,  
  "editor.formatOnPaste": false,  
  "editor.formatOnType": false,  
  "editor.defaultFormatter": "esbenp.prettier-vscode"  
},
```



ติดตั้ง Vue CLI

- ▶ <https://cli.vuejs.org/guide/installation.html>
- ▶ `npm install -g @vue/cli`
- ▶ `vue --version`



สร้างโปรเจคใหม่

- ▶ <https://cli.vuejs.org/guide/creating-a-project.html#vue-create>
- ▶ vue create ชื่อโปรเจค
- ▶ สร้างโปรเจคใหม่แบบ GUI ใช้คำสั่ง
- ▶ vue ui





Vue.js



What is Vue.js?

- ▶ Vue (pronounced /vju:/, like **view**) is a progressive framework for **building user interfaces**.
- ▶ <https://v3.vuejs.org/>



ฟีเจอร์ใหม่ใน Vue 3

- ▶ **ประสิทธิภาพดีกว่า** เวอร์ชัน 2 และมีขนาดเล็กลง
- ▶ Codebase เขียนด้วย TypeScript ทำให้**รองรับ TypeScript** ได้ดีขึ้น
- ▶ เปลี่ยนวิธีการเขียนมาเป็นรูปแบบที่เรียกว่า **Composition API** แทน Options API แบบเดิม (แบบเดิมก็ยังเขียนได้อยู่ แต่แนะนำแบบใหม่ เพราะอนาคตเขียนด้วย TypeScript ได้เลย) ข้อดีคือ ทำให้เราสามารถอ่าน และเรียนรู้โค้ดได้ง่ายขึ้น ตรงไปตรงมา เพราะเป็นฟังก์ชัน JavaScript ปกติที่เขียนกันปกติ



ฟีเจอร์ใหม่ใน Vue 3

- ▶ เพิ่ม **Fragment** เข้ามาเลย ไม่จำเป็นต้องเขียน root node มาครอบที่ template แล้วครับ
- ใน Vue 3 จะมีการแทรก virtual element (Fragment) มาให้เลย ต่อไปก็เขียนแบบนี้ได้เลย (ไม่ต้องครอบ root node ให้กับ h1 และ p แล้ว)

```
<template>  
  <h1>Hello</h1>  
  <p><a href="https://codingthailand.com">CodingThailand.com</a></p>  
</template>
```

ไม่ต้องเขียนแบบนี้ หรือลง library เพิ่ม

```
<template>  
  <div>  
    <h1>Hello</h1>  
    <p><a href="https://codingthailand.com">CodingThailand.com</a></p>  
  </div>  
</template>
```



ฟีเจอร์ใหม่ใน Vue 3

- ▶ **Suspense** ทำให้เราสามารถแสดง content (ชั่วคราวตามเงื่อนไข) บางอย่างก่อนที่จะแสดง content จริงๆ เช่น การทำ Loading ถ้าเราดึงข้อมูล User จาก Backend ถ้าดึงข้อมูลยังไม่เสร็จให้แสดง Loading... ก่อน ถ้าเสร็จแล้วค่อยแสดง UserProfile เป็นต้น

```
<Suspense>  
<template #default>  
<UserProfile />  
</template>  
<template #fallback>  
<div>Loading...</div>  
</template>  
</Suspense>
```

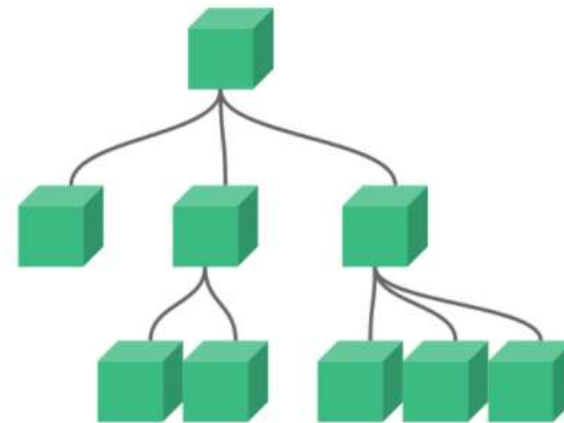
- ▶ **Teleport** (เมื่อก่อนเรียกว่า Portal) ไว้สำหรับ render DOM นอก component หรือที่ใดๆ ก็ได้ใน document ครับ เช่น เอาไว้ทำพวก modals หรือ popups



Vue.js พื้นฐาน



Composing with Components



Workshop



ติดตั้ง Bootstrap 5

- ▶ <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- ▶ `npm install bootstrap@5`



แนะนำ Vue Router เวอร์ชัน 4 (รองรับ Vue 3)

▶ <https://next.router.vuejs.org/>



Workshop: เกี่ยวกับ Component และ Router

- ▶ ทดลองนำ Admin Template เข้ามาใช้งาน
- ▶ <https://github.com/startbootstrap/startbootstrap-sb-admin>
- ▶ `npm i startbootstrap-sb-admin`




Template Syntax



Interpolations

```
<span>Message: {{ msg }}</span>
```



```
<span v-once>This will never change: {{ msg }}</span>
```

Using JavaScript Expressions

```
{{ number + 1 }}
```

```
{{ ok ? 'YES' : 'NO' }}
```

```
{{ message.split('').reverse().join('') }}
```

```
<div v-bind:id="'list-' + id"></div>
```



Attributes

```
<div v-bind:id="dynamicId"></div>
```

```
<!-- full syntax -->
```

```
<a v-bind:href="url"> ... </a>
```

```
<!-- shorthand -->
```

```
<a :href="url"> ... </a>
```

```
<!-- shorthand with dynamic argument -->
```

```
<a :[key]="url"> ... </a>
```


Conditional Rendering




```
1 <h1 v-if="awesome">Vue is awesome!</h1>
```

It is also possible to add an "else block" with `v-else` :

```
1 <h1 v-if="awesome">Vue is awesome!</h1>
2 <h1 v-else>Oh no 😞</h1>
```



List Rendering



```
<div v-for="item in items" :key="item.id">  
  <!-- content -->  
</div>
```

Event Handling

```
<div id="event-with-method">  
  <!-- `greet` is the name of a method defined below -->  
  <button @click="greet">Greet</button>  
</div>
```



Event Modifiers

```
<!-- the click event's propagation will be stopped -->
```

```
<a @click.stop="doThis"></a>
```

```
<!-- the submit event will no longer reload the page -->
```

```
<form @submit.prevent="onSubmit"></form>
```




```
<!-- modifiers can be chained -->
```

```
<a @click.stop.prevent="doThat"></a>
```

```
<!-- just the modifier -->
```

```
<form @submit.prevent></form>
```

Form Input Bindings



```
<input v-model="message" placeholder="edit me" />
<p>Message is: {{ message }}</p>
```



```
<div id="v-model-select" class="demo">
  <select v-model="selected">
    <option disabled value="">Please select one</option>
    <option>A</option>
    <option>B</option>
    <option>C</option>
  </select>
  <span>Selected: {{ selected }}</span>
</div>
```

Composition API



Composition API

- ▶ <https://v3.vuejs.org/guide/composition-api-introduction.html#why-composition-api>
- ▶ <https://v3.vuejs.org/api/composition-api.html#setup>



The end

