

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hiểu và áp dụng các lifecycle hooks trong Vue để quản lý trạng thái và tài nguyên trong component.
- ✓ Sử dụng Axios để thực hiện các cuộc gọi API và quản lý dữ liệu từ server.

NỘI DUNG

PHẦN I:

Bài 1 (2 điểm): Sử dụng Lifecycle Hooks để theo dõi trạng thái của component

Yêu cầu:

- Tạo một component UserStatus.vue theo dõi trạng thái người dùng (online/offline).
- Sử dụng các lifecycle hooks như onMounted, onUnmounted để đăng ký và hủy bỏ sự kiện online và offline.

Hướng dẫn:

Bước 1: Tạo component **Labs/Lab-6/Bai1.vue**

```
<template>
<div class="alert" :class="statusClass">Trạng thái: {{ isOnline ? "Online" : "Offline" }}</div>
</template>
<script setup>
import { ref, onMounted, onUnmounted } from "vue";
const isOnline = ref(true);
const statusClass = ref("alert-success");
const updateStatus = () => {
    isOnline.value = navigator.onLine;
    statusClass.value = isOnline.value ? "alert-success" : "alert-danger";
};
onMounted(() => {
    window.addEventListener("online", updateStatus);
    window.addEventListener("offline", updateStatus);
    updateStatus();
});
onUnmounted(() => {
    window.removeEventListener("online", updateStatus);
    window.removeEventListener("offline", updateStatus);
});
</script>
```

Bước 2: Nhúng **Bai1.vue** ở **App.vue**

```
<!-- App.vue -->
```

```
<template>

<div class="container mt-4">
<Bai1 />
</div>
</template>

<script setup>

import Bai1 from "./Labs/Lab-6/Bai1.vue";

</script>
```

Bài 2 (2 điểm): Sử dụng Lifecycle Hooks để thực hiện các tác vụ khi component được cập nhật

Yêu cầu:

1. Tạo một component DataFetcher.vue sử dụng lifecycle hooks onBeforeUpdate và onUpdated để quản lý việc gọi API và cập nhật dữ liệu.
2. Sử dụng json-server để làm API Fake

Hướng dẫn:

Bước 1: Cài đặt json-server:

- npm install json-server
- Tạo file db với nội dung sau

```
{
  "posts": [
    {
      "userId": 1,
      "id": 1,
      "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
      "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem eveniet architecto"
    },
    {
      "userId": 1,
      "id": 2,
      "title": "qui est esse",
      "body": "est rerum tempore vitae\\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\\nqui aperiam non debitis possimus qui neque nisi nulla"
    },
    {
      "userId": 1,
      "id": 3,
      "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
      "body": "et iusto sed quo iure\\nvolutatem occaecati omnis eligendi aut ad\\nvolutatem doloribus vel accusantium quis pariatur\\nmolestiae porro eius odio et labore et velit aut"
    }
  ]
}
```

```
{
  "userId": 1,
  "id": 4,
  "title": "eum et est occaecati",
  "body": "ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi
nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit"
}
}
```

Bước 2: Chạy db: json-server --watch db.json

```
\{^_}/ hi!
Loading db.json
Done

Resources
http://localhost:3000/posts
http://localhost:3000/users

Home
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
Watching...
```

Bước 3: Tạo component **Labs/Lab-6/Bai2.vue**:

```
<template>
<div>
<button class="btn btn-primary" @click="fetchData">Tải bài viết</button>
<div v-if="loading">Đang tải dữ liệu...</div>
<ul>
  <li v-for="post in posts" :key="post.id">
    <h3>{{ post.title }}</h3>
    <p>{{ post.body }}</p>
  </li>
</ul>
</div>
</template>
<script setup>
import { ref, onBeforeUpdate, onUpdated } from "vue";
import axios from "axios";
const posts = ref([]);
const loading = ref(false);
const fetchData = async () => {
  loading.value = true;
  const response = await axios.get("http://localhost:3000/posts");
  posts.value = response.data;
  loading.value = false;
};
onBeforeUpdate(() => {
```

```
        console.log("Component sẽ cập nhật.");
    });
onUpdated(() => {
    console.log("Component đã cập nhật.");
});
</script>
```

Bước 2: Nhúng Bai2.vue Ở App.vue

```
<!-- App.vue -->

<template>
<div class="container mt-4">
    <Bai2 />
</div>
</template>

<script setup>
    import Bai2 from "./Labs/Lab-6/Bai2.vue";
</script>
```

PHẦN II:

Bài 3 (2 điểm): Sử dụng Axios để thực hiện gọi API

Yêu cầu:

1. Tạo một component UserList.vue để hiển thị danh sách sản phẩm từ một API.
2. Sử dụng Axios để gọi API và hiển thị danh sách user

[Tải người dùng](#)

Leanne Graham
Sincere@april.biz
Địa chỉ: Kulas Light, Gwenborough

Ervin Howell
Shanna@melissa.tv
Địa chỉ: Victor Plains, Wisokyburgh

Clementine Bauch
Nathan@yesenia.net
Địa chỉ: Douglas Extension, McKenziehaven

Patricia Lebsack
Julianne.OConner@kory.org
Địa chỉ: Hoeger Mall, South Elvis

Chelsey Dietrich
Lucio_Hettinger@annie.ca
Địa chỉ: Skiles Walks, Roscoeview

Mrs. Dennis Schulist
Karley_Dach@jasper.info
Địa chỉ: Norberto Crossing, South Christy

Kurtis Weissnat
Telly.Hoeger@billy.biz
Địa chỉ: Rex Trail, Howemouth

Nicholas Runolfsdottir V
Sherwood@rosamond.me
Địa chỉ: Ellsworth Summit, Aliyaview

Glenna Reichert
Chaim_McDermott@dana.io
Địa chỉ: Dayna Park, Bartholomebury

Clementina DuBuque
Rey.Padberg@karina.biz
Địa chỉ: Kattie Turnpike, Lebsackbury

Hướng dẫn:

Bước 1: Tạo collection users trong db.json

```
"users": [
    {
        "id": 1,
        "name": "Leanne Graham",
        "username": "Bret",
        "email": "Sincere@april.biz",
        "address": {
            "street": "Kulas Light",
            "city": "Gwenborough"
        },
        "phone": "1-770-736-8031 x56442",
        "website": "hildegard.org"
    },
    {
        "id": 2,
        "name": "Ervin Howell",
        "username": "Antonette",
        "email": "Shanna@melissa.tv",
        "address": {
            "street": "Victor Plains",
            "city": "Wisokyburgh"
        },
        "phone": "010-692-6593 x09125",
        "website": "anastasia.net"
    },
    {
        "id": 3,
        "name": "Clementine Bauch",
        "username": "Samantha",
        "email": "Nathan@yesenia.net",
        "address": {
            "street": "Douglas Extension",
            "city": "McKenziehaven"
        },
        "phone": "1-493-170-9643 x13625",
        "website": "kharma.net"
    }
]
```

```

"address": {
    "street": "Douglas Extension",
    "city": "McKenziehaven"
},
"phone": "1-463-123-4447",
"website": "ramiro.info"
},
{
"id": 4,
"name": "Patricia Lebsack",
"username": "Karianne",
"email": "Julianne.OConner@kory.org",
"address": {
    "street": "Hoeger Mall",
    "city": "South Elvis"
},
"phone": "493-170-9623 x156",
"website": "kale.biz"
},
{
"name": "3213",
"email": "2131321",
"phone": "312",
"website": "213123",
"address": {
    "street": "213123",
    "city": "1231"
},
"id": 5
}
]

```

Bước 2: Tạo component **Labs/Lab-6/Bai3.vue**

```

<template>
<div class="container mt-4">
<button class="btn btn-primary mb-3" @click="fetchData">Tải người dùng</button>
<div v-if="loading" class="alert alert-info">Đang tải dữ liệu...</div>
<div class="row">
<div v-for="user in users" :key="user.id" class="col-md-4 mb-3">
<div class="card">
<div class="card-body">
<h5 class="card-title">{{ user.name }}</h5>
<h6 class="card-subtitle mb-2 text-muted">{{ user.email }}</h6>
<p class="card-text">
Địa chỉ: {{ user.address.street }}, {{ user.address.city }}
</p>
</div>
</div>
</div>
</div>
</div>
</template>

```

```
<script setup>
import { ref, onBeforeUpdate, onUpdated } from "vue";
import axios from "axios";
const users = ref([]);
const loading = ref(false);
const fetchData = async () => {
    loading.value = true;
    const response = await axios.get("http://localhost:3000/users");
    users.value = response.data;
    loading.value = false;
};
onBeforeUpdate(() => {
    console.log("Component sẽ cập nhật.");
});
onUpdated(() => {
    console.log("Component đã cập nhật.");
});
</script>
```

Bước 2: Nhúng component Bai3.vue ở App.vue

```
<!-- App.vue -->

<template>
<div class="container mt-4">
    <Bai3 />
</div>
</template>

<script setup>
    import Bai3 from "./Labs/Lab-6/Bai3.vue";
</script>
```

Bài 4 (2 điểm): Thêm người dùng mới

Yêu cầu:

- Dựa vào Component UserList.vue ở bài số 3

Tạo một component UserDashboard.vue để hiển thị thông tin người dùng và cho phép người dùng thêm tài khoản mới qua một form.

- Sử dụng các lifecycle hooks để thực hiện gọi API và quản lý trạng thái khi user được thêm mới.

Bước 1: Tạo component Labs/Lab-6/Bai4.vue

```
<<template>
<div class="user-dashboard">
```

```

<h1>Thêm người dùng</h1>

<!-- Form thêm người dùng mới -->

<form @submit.prevent="addUser">

<div class="form-group">
<label for="name">Tên:</label>
<input type="text" v-model="newUser.name" class="form-control" required />
</div>

<div class="form-group">
<label for="email">Email:</label>
<input type="email" v-model="newUser.email" class="form-control" required />
</div>

<div class="form-group">
<label for="password">Mật khẩu:</label>
<input type="password" v-model="newUser.password" class="form-control" required />
</div>

<button type="submit" class="btn btn-primary">Thêm người dùng</button>
</form>

<h2>Danh sách người dùng</h2>

<ul>
<li v-for="user in users" :key="user.id">{{ user.name }} - {{ user.email }}</li>
</ul>
</div>
</template>

<script setup>
import { ref, onMounted } from "vue";
import axios from "axios";
const users = ref([]); // Danh sách người dùng
const newUser = ref({ name: "", email: "", password: "" }); // Thông tin người dùng mới
// Hàm gọi API để lấy danh sách người dùng
const fetchUsers = async () => {
try {
const response = await axios.get("http://localhost:3000/users");
}

```

```

users.value = response.data;

} catch (error) {

console.error("Có lỗi khi lấy danh sách người dùng:", error);

}

};

// Hàm thêm người dùng mới

const addUser = async () => {

try {

const response = await axios.post("http://localhost:3000/signup", newUser.value);

users.value.push(response.data); // Thêm người dùng mới vào danh sách

newUser.value = { name: "", email: "", password: "" }; // Reset form

} catch (error) {

console.error("Có lỗi khi thêm người dùng:", error);

}

};

onMounted(() => {

fetchUsers();

});

</script>

```

Bước 2: Nhúng component Bai4.vue ở App.vue

```

<!-- App.vue -->

<template>

<div class="container mt-4">

<Bai4 />

</div>

</template>

<script setup>

import Bai4 from "./Labs/Lab-6/Bai4.vue";

</script>

```

**** Giảng viên có thể cho các bài tập phù hợp phục vụ assignment.**

***** Yêu cầu nộp bài:**

SV nén file (*hoặc share thư mục google drive*) bao gồm các yêu cầu đã thực hiện

trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM

--- Hết ---