Copyright Junichiro Ueno

Part 2: Let's Create a Screen with Movement with JavaScript (Vue.js)

Titech 2020/10/31

Junichiro Ueno

Copyright Junichiro Ueno

🎃 Retrospective 🎃

✅ After completing the basics of HTML, CSS, and JavaScript
The next step is to experience web application development in earnest.

If you didn't attend last time and are new to the event, please drop the latest code from Github first.

https://github.com/GuildWorks/titech-2020

参考
Previous reference (building environment)

Copyright Junichiro Ueno

✅ Previously, we used HTML, CSS, and JavaScript to create Member list and member details are ~~ completed. ~~ **Not done. **

I won't go into the details of HTML and CSS in this article, but I'll explain what you need to do as I write the code one by one. If you have any questions, please ask.

😈 **If no one asks questions, there's a chance we'll dump them on you.** 🎯

Copyright Junichiro Ueno

🎃 Member List 🎃

TOTAL SAMPLE PROJECT

メンバーリスト　メンバープロフィール

メンバーリスト

氏名	Email	担当	
仁和 泰也	niwa@example.com	リーダー	Profile
川面 崇義	kawadura@example.com	メンバー	Profile
石北 俊寛	kitaishi@example.com	メンバー	Profile
岸 美貴	kishi@example.com	メンバー	Profile
大内田 結貴	ohdauchi@example.com	メンバー	Profile

Copyright Junichiro Ueno

🎃 Member Details 🎃

TOTAL SAMPLE PROJECT

メンバーリスト　メンバープロフィール

メンバープロフィール

NAME

仁和 泰也

こんにちは。仁和泰也といいます。友人からはにわりんと呼ばれているので、にわりんと呼んでください。住まいは新宿で、大学4回生です。趣味は、写真を撮ることが好きなので、休日は山に登り、風景や草花をカメラに収めています。また、自粛生活が続いたとき、料理くらいできないとと思い、夏から料理教室に通っています。得意料理はカレーライスです。

生年月日	2000年10月10日
血液型	A型
出身地	東京都
所属・部署	東京工業大学 情報理工学院
星座	天秤座
趣味	写真・料理
ニックネーム	にわりん

Copyright Junichiro Ueno

🎃 Get the latest data! 🎃

✅ Navigate to the titech-2020 folder where you want to save the file and Let's execute the following command with `git bash` or `Terminal`.

```
git pull
```

✅ The latest data was obtained.
If you can find this document under `docs/Phase2`, it's a success.

Copyright Junichiro Ueno

🎃 What is Vue.js? 🎃

The Progressive
JavaScript Framework

親しみやすい

すでに HTML、CSS そして JavaScript を知っていますか？ ガイドを読んで、すぐにモノ作りを開始しましょう！

融通が効く

ライブラリと完全な機能を備えたフレームワークの間で拡張できる徐々に採用可能なエコシステム

高性能

20KB min+gzip ランタイム
猛烈に速い Virtual DOM
最小限の努力で最適化が可能

🇯🇵 : https://jp.vuejs.org/
🇺🇸 : https://vuejs.org/

Copyright Junichiro Ueno

🎃 Running the Development Environment 🎃

✅ Last time, we'll even get to the point where we can already `npm run dev` in building the environment.
This time we will always run this command.

📜 Go to the `titech-2020/titech-nuxt-tutorial` folder and click
Let's execute the following command with `git bash` or `Terminal`.

```
npm install
```

When you're done.

```
npm run dev
```

Copyright Junichiro Ueno

🎃 Top page 🎃

TOTAL SAMPLE PROJECT

Programming Boot Camp #2

Vue.js/Nuxt.js

Vue.js と Nuxt.js

Copyright Junichiro Ueno

🎃 Confirmation 🎃

Once you've done so, let's check the contents.

✅ Checking the source code

Let's open the following file in Visual Studio Code.

📜 `titech-2020/titech-nuxt-tutorial/pages/index.vue`

Copyright Junichiro Ueno

```
<template>
  <div
    class="flex flex-col-reverse sm:flex-row jusitfy-between items-center py-12"
  >
    <div
      class="sm:w-2/5 flex flex-col items-center sm:items-start text-center sm:text-left"
    >
      <h1
        class="text-6xl text-blue-900 font-bold leading-none tracking-wide mb-2"
      >
        Programming Boot Camp #2
      </h1>
      <h2
        class="text-4xl text-blue-500 text-secondary tracking-widest mb-6"
      >
        Vue.js/Nuxt.js
      </h2>
      <p class="text-gray-600 leading-relaxed mb-6">
        <a href="https://vuejs.org/index.html" class="text-blue-900">Vue.js</a>
        と
        <a href="https://ja.nuxtjs.org/" class="text-blue-900">Nuxt.js</a>
      </p>
    </div>
    <div class="mb-16 sm:mb-0 mt-8 sm:mt-0 sm:w-3/5 sm:pl-12">
      <SvgImage />
    </div>
  </div>
</template>
<script lang="ts">
import { defineComponent } from 'nuxt-composition-api'
import SvgImage from '@/components/svg-image.vue'
export default defineComponent({
  components: {
    SvgImage,
  },
})
</script>
<style></style>
```

Copyright Junichiro Ueno

🎃 説明 🎃

✅ The `Vue` allows you to write code in three different areas.

- It's easy to write in one file and understand ❓ ❓ ❓

✅ The contents of the `.vue` file are a collection of what we learned last time.

- Doesn't it seem to make sense to you? ❓ ❓ ❓

✅ Let's take a look at each area.

Copyright Junichiro Ueno

🎃 template 🎃

```
<template>
  ...
</template>
```

✅ This is the area for writing the `HTML` we learned last time.

You can write the `HTML` in `<template>` to `</template>`.

✅ The `Vue` allows you to write not only `HTML` but also `code`.

✅ And how to write.

Copyright Junichiro Ueno

```
<template>
  <div>
    ...
  </div>
</template>
```

⚠️ The outermost tag in the `<template>` must be a single tag.

```
<template>
  <div>...</div>
  <div>...</div>
</template>
```

❌ This will result in an error.😢

Copyright Junichiro Ueno

🎃 script 🎃

```
<script lang="ts">
  ...
</script>
```

✅ This is the area where we learned to write JavaScript last time.

`lang="ts"`

💡 It actually specifies a language called "TypeScript".
"TypeScript" is a mainstream programming language that is an alternative to "JavaScript.

Simply put, it's "type definable JavaScript".

Copyright Junichiro Ueno

```
<script  lang="ts">
import { defineComponent } from 'nuxt-composition-api'
import SvgImage from '@/components/svg-image.vue'
export default defineComponent({
  components: {
    SvgImage,
  },
})
</script>
```

```
import { defineComponent } from 'nuxt-composition-api'
```

🔰 This is a spell.

First of all, let's remember that the `defineComponent` is available.

Copyright Junichiro Ueno

```
import SvgImage from '@/components/svg-image.vue'
```

🔰 Similarly, this is a declaration for using `svg-image.vue`.

The above specification enables you to use files containing `svg(Scalable Vector Graphics)`.

💡 Components are used to divert parts, etc.

Copyright Junichiro Ueno

```
export default defineComponent({
  components: {
    SvgImage,
  },
})
```

By specifying what you want to use in the `components`, you can set the Available in the `<template>`.

use case

```
<div class="mb-16 sm:mb-0 mt-8 sm:mt-0 sm:w-3/5 sm:pl-12">
  <SvgImage />
</div>
```

Copyright Junichiro Ueno

The following sections are loaded and displayed.

Copyright Junichiro Ueno

🎃 style 🎃

```
<style>
  ...
</style>
```

✅ This is the area where you write the CSS you learned last time.

I learned last time Tailwind is useful, so I may not use it much.

This is where we'll look at this later in the exercise.

Copyright Junichiro Ueno

🎃 Exercises 🎃

✅ You may have gotten the gist of it, but I don't think you're quite sure yet.

💪 Let's actually proceed.

Copyright Junichiro Ueno

🎃 Creating a list page 🎃

📜 `titech-nuxt-tutorial/pages/list.vue`

```
<template>

</template>
<script lang="ts">

</script>
<style>

</style>
```

😇 There's nothing inside yet.
✏️ Let's add the code.

Copyright Junichiro Ueno

✅ Before we do that, let's take another look at the finished image.

TOTAL SAMPLE PROJECT

メンバーリスト　メンバープロフィール

メンバーリスト

氏名	Email	担当	
仁和 泰也	niwa@example.com	リーダー	Profile
川面 崇義	kawadura@example.com	メンバー	Profile
石北 俊寛	kitaishi@example.com	メンバー	Profile
岸 美貴	kishi@example.com	メンバー	Profile
大内田 結貴	ohdauchi@example.com	メンバー	Profile

Copyright Junichiro Ueno

```
<template>
  <div>
    <h1 class="text-2xl sm:text-3xl text-blue-900 p-4 mb-4 md:mb-8 border-b">メンバーリスト</h1>
  </div>
</template>
```

✅ Let's add the above from `<template>` to `</template>`.

✅ If the `npm run dev` is running, it should automatically reflect the saved code.

💡 View http://localhost:3000/list in your browser.

✅ The next one will be more at once.
For now, let's just copy and paste.

Copyright Junichiro Ueno

```
<template>
  <div>
    <h1 class="text-2xl sm:text-3xl text-blue-900 p-4 mb-4 md:mb-8 border-b">メンバーリスト</h1>
    <div class="list-table shadow-md sm:rounded overflow-y-auto">
      <table class="w-full text-md bg-white">
        <thead>
          <tr class="border-b bg-blue-900 text-white">
            <th class="text-left p-3 px-5">氏名</th>
            <th class="text-left p-3 px-5">Email</th>
            <th class="text-left p-3 px-5">担当</th>
            <th></th>
          </tr>
        </thead>
        <tbody class="text-gray-900">
          <tr
            class="border-b bg-gray-100"
          >
            <td class="py-3 px-5 whitespace-no-wrap sm:whitespace-normal">
              仁和 泰也
            </td>
            <td class="py-3 px-5 whitespace-no-wrap sm:whitespace-normal">
              niwa@example.com
            </td>
            <td class="py-3 px-5 whitespace-no-wrap sm:whitespace-normal">
              メンバー
            </td>
            <td class="py-3 px-5">
              <div class="flex justify-end items-center">
                <a
                  :href="'/user/0001'"
                  class="text-sm bg-blue-500 hover:bg-blue-700 text-white py-1 px-2 rounded focus:outline-none focus:shadow-outline flex items-center"
                >
                  <span
                    class="rounded-full w-5 h-5 bg-white p-0 border-px border-white inline-flex items-center justify-center text-blue-500 mr-2"
                  >
                    <svg
                      fill="currentColor"
                      class="w-5 h-5"
                      xmlns="http://www.w3.org/2000/svg"
                      viewBox="0 0 24 24"
                    >
                      <path d="M0 0h24v24H0z" fill="none" />
                      <path
                        d="M12 2C6.48 2 2 6.48 2 12s4.48 10 10 10 10-4.48 10-10S17.52 2 12 2zm0 3c1.66 0 3 1.34 3 3s-1.34 3-3 3-3-1.34-3-3 1.34-3 3-3zm0 14.2c-2.5 0-4.71-1.28-6-3.22.03-1.99 4-3.08 6-3.08 1.99 0 5.97 1.09 6 3.08-1.29 1.94-3.5 3.22-6 3.22z"
                      />
                    </svg>
                  </span>
                  Profile
                </a>
              </div>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</template>
```

Copyright Junichiro Ueno

💡 Would you have gotten a screen like this?

✅ You've shown one person on the member list.

🔰 This is where it starts to feel like a real programming thing, but But first, here's a quick foundation.

Copyright Junichiro Ueno

🎃 Programming Basics 🎃

⚠️ More on types and variables in #3.

What is Type

Primitive type

`string` is a string
`number` is a number
`boolean` is a boolean

References

[TypeScriptの型入門]

Copyright Junichiro Ueno

🎃 Variable 🎃

```
const label: string = "MemberList" // OK
const label: string = 3            // NG

const num: number = 3              // OK
const num: number = "3"            // NG

const bool: boolean = true         // OK
const bool: boolean = "false"      // NG
```

✅ Incorrect type is an error.

Copyright Junichiro Ueno

```
const label: string = "MemberList"
label = "Change" // NG

let label2: string = "MemberList"
label2 = "Change" // OK
```

✅ Use `const` if it is immutable and `let` if it is variable.

Copyright Junichiro Ueno

🎃 配列 🎃

```
const name: string[] = ["上野", "今橋", "京極"]
name.push("金")                                  // 配列の最後に`金`を追加する

console.log(name)                               // ["上野", "今橋", "京極", "金"]
console.log(name[1])                            // "今橋"
```

If you have more than one piece of data, use an array.

⚠️ Note that the array starts from `0`.

supplement

`shift()` removes the head, `pop()` removes the tail

Copyright Junichiro Ueno

For now, that's it for the basics 💨

✅ Is there anything that is not clear to you? ❓ ❓ ❓

💪 So let's get back to it.

Copyright Junichiro Ueno

🎃 ダミーデータの用意 🎃

📜 `titech-nuxt-template/mock/userlist.json`

```
{
  "userlistData": [
    {
      "id": "0001",
      "name": "仁和 泰也",
      ...
    },
  ]
}
```

This is a file called `json` that defines the dummy data.

Copyright Junichiro Ueno

🎃 Importing dummy data 🎃

📜 `titech-nuxt-tutorial/pages/list.vue`

✅ First, let's copy and paste.
Then, let's see what's inside.

Copyright Junichiro Ueno

```
<script lang="ts">
import { defineComponent, reactive } from 'nuxt-composition-api'
import userlistJson from '@/mock/userlist.json'
type UserList = {
  id: string
  name: string
  email: string
  role: string
  iconUrl: string
  profile: {
    title: string
    detail: string
  }[]
}
export default defineComponent({
  setup(_) {
    const userList = reactive<UserList[]>(userlistJson.userlistData)
    return {
      userList,
    }
  },
})
</script>
```

Copyright Junichiro Ueno

```
import userlistJson from '@/mock/userlist.json'
```

✅ The name of `userlistJson` is defined as the dummy data mentioned above.

📝 I said `import` is a declaration to use the file.
You should give it a name when you use it.
In this case, we named the file `userlistJson`.

Copyright Junichiro Ueno

```
type UserList = {
  id: string
  name: string
  email: string
  role: string
  iconUrl: string
  profile: {
    title: string
    detail: string
  }[]
}
```

✅ It's called a type alias, but
It's a useful way to define types.

💡 If the type is different, you'll notice the error.

Copyright Junichiro Ueno

```
export default defineComponent({
  setup(_) {
    const userList = reactive<UserList[]>(userlistJson.userlistData)
    return {
      userList,
    }
  },
})
```

⁉ There are more unfamiliar ones again.

The `setup(_)` is what is executed when this page is called.
It returns `return` if you want to use it in `<template>`.

Copyright Junichiro Ueno

```
const userList = reactive<UserList[]>(userlistJson.userlistData)
```

📜 In the file `titech-nuxt-tutorial/mock/userlist.json`, you can find Get the dummy data with `userlistJson.userlistData`.

Copyright Junichiro Ueno

```
{
  "userlistData": [
    ...
  ]
}
```

✅ Get the `userlistData` in a file with the name specified in the `import`.

✅ An error is avoided by specifying the array of `UserList` which is defined by the type alias.

✅ Set the dummy data to `return` so that it can be used in `<template>`.

Copyright Junichiro Ueno

✅ Now that we have the dummy data, let's display it in the member list.

```
<tr
  v-for="(user, index) in userList"
  :key="index"
  class="border-b bg-gray-100"
>
```

✅ Add the above to `tr` on line 17.

💡 If you view it in your browser, you'll see a lot of `仁和`.
Still, it's a success for once.

Copyright Junichiro Ueno

🎃 for statement 🎃

```
v-for="(user, index) in userList" :key="index"
```

✅ In `Vue.js`, you can run a loop in `template`.

✅ userList `This is the one set to` return `in the` <script>`.
It contains an array of dummy data.

✅ Here, we want to get the data one by one from the array, so we use `for`.

Copyright Junichiro Ueno

- ✅ `(user, index)` Now one of the arrays is stored in `user`.
- ✅ `index` contains the index (counter).

Copyright Junichiro Ueno

❓ The only thing that comes up in a loop is 仁和
This is because we are not using the data we have acquired.

```
<td class="py-3 px-5 whitespace-no-wrap sm:whitespace-normal">
  仁和 泰也
</td>
```

The name is specified as a fixed value, so it doesn't matter how many times you loop over it, it's always the same name! 😓

Copyright Junichiro Ueno

✅ Make changes to make your data available.

```
<td class="py-3 px-5 whitespace-no-wrap sm:whitespace-normal">
  {{ user.name }}
</td>
```

That's it.
✅ You can use variables in `<template>` by enclosing them in `{{ }}` and raising them.

✅ Because the variable `user` contains one data each, you can get the name of a user with `user.name`.

💡 When you view it in your browser, it's just the name that has changed to dummy data.

Copyright Junichiro Ueno

✅ Let's change the other data as well.

```
<td class="py-3 px-5 whitespace-no-wrap sm:whitespace-normal">
  {{ user.email }}
</td>
<td class="py-3 px-5 whitespace-no-wrap sm:whitespace-normal">
  <template v-if="user.role === 'admin'">リーダー</template>
  <template v-else>メンバー</template>
</td>
```

✅ The `Email` and `in-charge` have also been changed.

⁉ If you look closely, there's another unfamiliar `v-if` and `v-else`.

✅ This is also available in `Vue`.

Copyright Junichiro Ueno

🎃 if statement 🎃

```
<template v-if="user.role === 'admin'">リーダー</template>
<template v-else>メンバー</template>
```

✅ If the condition of `v-if` is `true`, it is executed.
✅ If the condition is `false`, `else` is executed.

👀 Dummy data is `admin` only for `仁和` and `member` for the others.

Therefore, only `仁和` becomes a leader since the contents of `user.role` are checked for `admin`.
(`===` will be `true` if the value is the same, or `false` if the value is different.)

Copyright Junichiro Ueno

💡 When you click on the Profile button, you'll see
You want to be taken to that person's details page.

✅ Let's make some changes there too.
(We don't have a detailed page yet.)

```
:href="'/user/' + user.id"
```

👲 仁和
http://localhost:3000/user/0001
👳 川面
http://localhost:3000/user/0002

Copyright Junichiro Ueno

✅ Also adjust the look and feel at the end of the list.

```
<style>
tbody tr:nth-child(odd) {
  @apply bg-white;
}
</style>
```

✅ Added CSS in `style`.
When a line is `odd` (odd number), the color of the line is set to white.

Copyright Junichiro Ueno

🎃 Complete the list 🎃

TOTAL SAMPLE PROJECT

メンバーリスト

氏名	Email	担当	
仁和 泰也	niwa@example.com	リーダー	Profile
川面 崇義	kawadura@example.com	メンバー	Profile
石北 俊寛	kitaishi@example.com	メンバー	Profile
岸 美貴	kishi@example.com	メンバー	Profile
大内田 結貴	ohdauchi@example.com	メンバー	Profile

Copyright GuildWorks

Copyright Junichiro Ueno

🎃 One more thing 🎃

💡 Let's give it one more try.

❓ It's not so easy to use that you have to click on a button to go to a detailed page.

✅ Let's change it so that you can click anywhere in a row to transition.

✅ Let's also make it easier to understand when the cursor is on a line.

Copyright Junichiro Ueno

```
setup(_) {
  const userList = reactive<UserList[]>(userlistJson.userlistData)
  const userLink = (userId: string): void => {
    window.location.href = '/user/' + userId
  }
  return {
    userList,
    userLink
  }
},
```

✅ Let's change the `setup`.

Copyright Junichiro Ueno

```
<tr
  v-for="(user, index) in userList"
  :key="index"
  class="border-b bg-gray-100"
  @click="userLink(user.id)"
>
```

✅ I'll also add a line in the `tr` tag with `v-for`.

Copyright Junichiro Ueno

💡 In addition, you'll learn something new.

```
const userLink = (userId: string): void => {
  window.location.href = '/user/' + userId
}
```

```
@click="userLink(user.id)"
```

✅ Adds a function to be executed when clicked by `@click`.
✅ Function `userLink` is called with the target `user.id` as an argument
✅ `userLink` function creates a URL as if the user clicked the profile button.

Copyright Junichiro Ueno

🎃 function 🎃

```
const userLink = (userId: string): void => {
  window.location.href = '/user/' + userId
}
```

😛 To be precise, it's called an arrow function. (Since `->` and `=>` are arrow-like.)

`(userId: string)` part is the list of arguments.
`: void` part defines the type of the return value.
`{ }` is what you want to execute.

✅ If `userId` is `"0001"`, we go to `/user/0001`.

Copyright Junichiro Ueno

✅ Let's also add one more thing to make it easier to understand when the cursor is above the line.

```
<tr
  v-for="(user, index) in userList"
  :key="index"
  class="border-b bg-gray-100 hover:bg-orange-100 cursor-pointer"
  @click="userLink(user.id)"
>
```

```
hover:bg-orange-100 cursor-pointer
```

✅ Let's add these two things to the list.

Copyright Junichiro Ueno

A little improvement can change the usability quite a bit.👍

It's fun to think about things and create something good!❤️

✅ That's the end of the list, now let's move on to the details✨

Copyright Junichiro Ueno

🍪 Create a detail page 👻

📜 `titech-nuxt-tutorial/pages/user/_id.vue`

```
<template>

</template>
<script lang="ts">

</script>
<style>

</style>
```

😇 Also, there is nothing inside.

Copyright Junichiro Ueno

💡 If you look closely, the file name is a little strange.

`_id.vue`

Isn't it `detail.vue` because it's detailed?

In fact, by setting `_id.vue`, you will be able to support a variable url, such as `/0001`.

💡 View http://localhost:3000/0001 in your browser.

✅ I'm sure the screen will show up, albeit empty.

✏️ Now let's add the code.

Copyright Junichiro Ueno

✅ Before we do that, let's take another look at the finished image.

Copyright Junichiro Ueno

```
<template>
  <div class="container mx-auto">
    <h1 class="text-2xl sm:text-3xl text-blue-900 p-4 mb-4 md:mb-8 border-b">
      メンバープロフィール
    </h1>
  </div>
</template>
```

✅ You've displayed the headline of your member profile.

⚠️ It's the same headline on this list.

‼️ When you realize that, it's an opportunity for commonality. ‼️

💡 Speaking of which, you mentioned the components. 🍫🍫🍫

Copyright Junichiro Ueno

🍪 Creating Components 👻

📜 `titech-nuxt-template/components/page-heading.vue`

```
<template>
  <h1 class="text-2xl sm:text-3xl text-blue-900 p-4 mb-4 md:mb-8 border-b">
    <slot></slot>
  </h1>
</template>
```

💡 `<h1>` The same content is already in the content as the part of

✅ There's a code called `<slot></slot>` for the first time.
This will be explained together with the call.

Copyright Junichiro Ueno

📜 titech-nuxt-tutorial/pages/user/_id.vue

```
<script lang="ts">
import { defineComponent, reactive, SetupContext } from 'nuxt-composition-api'
import PageHeading from '@/components/page-heading.vue'

export default defineComponent({
  components: {
    PageHeading,
  },
})
</script>
```

```
import PageHeading from '@/components/page-heading.vue'
```

✅ Here's the definition.

Copyright Junichiro Ueno

```
components: {
  PageHeading,
},
```

✅ This is where the components used are registered.

```
<template>
  <div class="container mx-auto">
    <PageHeading>メンバープロフィール</PageHeading>
  </div>
</template>
```

✅ Change the contents of `template` from an `h1` tag to a `PageHeading` tag.

Copyright Junichiro Ueno

💡 component, you can now use the `PageHeading` tag.

```
<PageHeading>メンバープロフィール</PageHeading>
```

```
<h1 class="text-2xl sm:text-3xl text-blue-900 p-4 mb-4 md:mb-8 border-b">
  <slot></slot>
</h1>
```

✅ The `slot` part is replaced by the `メンバープロフィール` between the `PageHeading` tags.

Copyright Junichiro Ueno

✅ For the list, you can set up a メンバーリスト for them.

📜 `titech-nuxt-tutorial/pages/list.vue`

🍩 We'll skip it here, but if you make the same changes, the commonality is complete

Copyright Junichiro Ueno

✅ We'll continue on.

📜 `titech-nuxt-tutorial/pages/user/_id.vue`

🃏 Replace all of the `script` with one.

Copyright Junichiro Ueno

```
<script lang="ts">
import { defineComponent, reactive, SetupContext } from 'nuxt-composition-api'
import PageHeading from '@/components/page-heading.vue'
import ProfileNameIcon from '@/components/profile-name-icon.vue'
import ProfileTable from '@/components/profile-table.vue'
import userlistJson from '@/mock/userlist.json'

type UserList = {
  id: string
  name: string
  email: string
  role: string
  iconUrl: string
  profile: {
    title: string
    detail: string
  }[]
}

export default defineComponent({
  components: {
    PageHeading,
    ProfileTable,
    ProfileNameIcon,
  },
  setup(_, { root }: SetupContext) {
    const userList = reactive<UserList[]>(userlistJson.userlistData)
    const userData = (): UserList => {
      if (
        userList.filter((user) => user.id === root.$route.params.id).length > 0
      )
        return userList.filter((user) => user.id === root.$route.params.id)[0]
      else
        return {
          id: '',
          name: '',
          email: '',
          role: '',
          iconUrl: '',
          profile: [
            {
              title: '',
              detail: '',
            },
          ],
        }
    }
    return {
      userData,
    }
  }
})
</script>
```

Copyright Junichiro Ueno

✅ Is this what you're not used to seeing?

```
const userData = (): UserList => {
  if (
    userList.filter((user) => user.id === root.$route.params.id).length > 0
  )
    return userList.filter((user) => user.id === root.$route.params.id)[0]
  else
    return {
      id: '',
      ...
    }
}
```

💡 The `if` is basically the same as in `templete`.

Copyright Junichiro Ueno

```
if (
  userList.filter((user) => user.id === root.$route.params.id).length > 0
)
```

✅ The `filter` extracts a specific condition from an array of dummy data.

📌 Checking that `user.id` and `root.$route.params.id` are the same.

⭕ http://localhost:3000/user/0001
❌ http://localhost:3000/user/0009

`0001` is `true` because it exists and `0009` is `false` because it does not exist.

Copyright Junichiro Ueno

🍪 Tips 👻

❓ When you write logic, what's inside? ❓ ❓ ❓

✅ You can check it by logging it.

```
console.log()
```

For example

```
console.log(userList.filter((user) => user.id === root.$route.params.id).length > 0)
```

💡 Let's view it in your browser.

Copyright Junichiro Ueno

✅ If you're in Chrome, go to the Developer Tools.

When you view the following url, `console` will be marked as `true` or `false`.

⭕ http://localhost:3000/user/0001は `true`
❌ http://localhost:3000/user/0009は `false`

🐛 It allows you to develop with real data instead of just thinking in your head.

🎁 If you're wondering what's in it, give it a try.

⬅️ BACK Back to the explanation.

Copyright Junichiro Ueno

```
<template>
  <div class="container mx-auto">
    <PageHeading>メンバープロフィール</PageHeading>
    <div class="lg:w-11/12 mx-auto flex flex-wrap">
      <div class="p-4 lg:px-8 lg:w-1/2 w-full">
        <ProfileNameIcon
          :icon-url="userData().iconUrl"
          :user-name="userData().name"
          :email="userData().email"
        />
        <hr class="my-4 sm:my-8">
        <p class="leading-relaxed">{{ userData().comment }}</p>
      </div>
      <ProfileTable
        class="mt-8 lg:w-1/2 w-full"
        :profile="userData().profile"
      />
    </div>
  </div>
</template>
```

Copyright Junichiro Ueno

📜 `titech-nuxt-tutorial/pages/user/_id.vue`

🃏 Replace `template` at once.

Copyright Junichiro Ueno

TOTAL SAMPLE PROJECT

メンバープロフィール

NAME

仁和 泰也

こんにちは。仁和 泰也といいます。友人からはにわりんと呼ばれているので、にわりんと呼んでください。住まいは新宿で、大学4回生です。趣味は、写真を撮ることが好きなので、休日は山に登り、風景や草花をカメラに収めています。また、自粛生活が続いたとき、料理くらいできないとと思い、夏から料理教室に通っています。得意料理はカレーライスです。

生年月日	2000年10月10日
血液型	A型
出身地	東京都
所属・部署	東京工業大学 情報理工学院
星座	天秤座
趣味	写真・料理
ニックネーム	にわりん

Copyright GuildWorks Inc.

✅ As for the movement, it's complete.

Copyright Junichiro Ueno

👀 Let's look at what might be the point.

📜 `titech-nuxt-tutorial/pages/user/_id.vue`

```
<ProfileNameIcon
  :icon-url="userData().iconUrl"
  :user-name="userData().name"
  :email="userData().email"
/>
```

The `ProfileNameIcon` tag associates the following three parameters.

- `:icon-url`
- `:user-name`
- `:email`

Copyright Junichiro Ueno

titech-nuxt-tutorial/components/profile-name-icon.vue

```
<script lang="ts">
import { defineComponent } from 'nuxt-composition-api'
export default defineComponent({
  props: {
    iconUrl: { type: String },
    userName: { type: String },
    email: { type: String },
  },
  setup(props) {
    return {
      props,
    }
  }
})
</script>
```

Copyright Junichiro Ueno

- 🍭 Receives a parameter with `props`.
- 🍨 Return `is specified as an argument of` setup`.
- 🍬 The usage is the same as the variable `{{ props.userName }}`.

Copyright Junichiro Ueno

🍪 summary 👻

✅ Listings and details are the basis of many websites.

☝️ Next time, if you can register and log in around membership and login, you'll be able to cover most of it.

🧁 Thank you for your long hours today.🤷

🎉 I look forward to being part of something that will change the world! 🎉