

Everything you need to know about Pinia in Vue.js

Are you a Vue developer looking to streamline management and take your app development to the next level? Say 'hello' to Pinia , the game change management library for Vue developers .



What is Pinia?

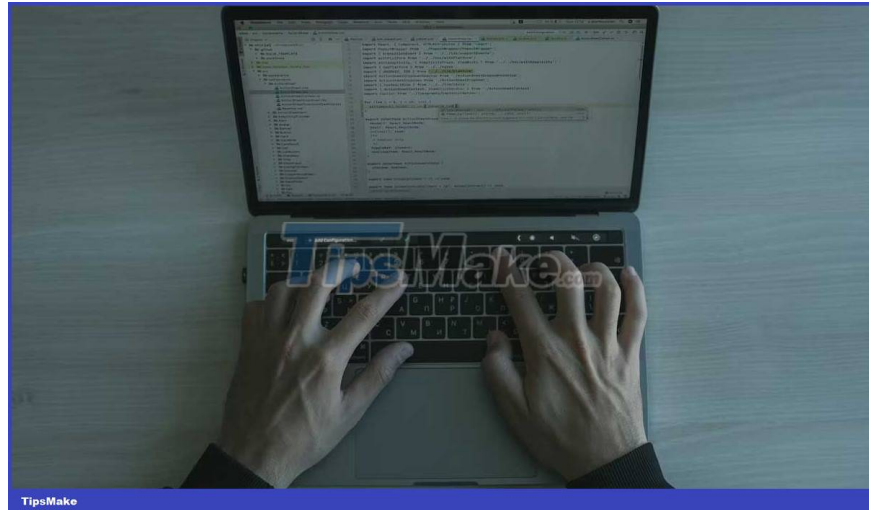
Pinia is a Vue-specific management library, designed to bring unparalleled simplicity and efficiency to Vue projects. Pinia is inspired by the best of modern management, and integrates into your application extremely light and simple.

The philosophy behind Pinia is to keep things to a minimum & tasteful, making it easy for developers to manage their apps. With an intuitive approach, Pinia lets you focus on the most common problems while delivering an exceptional user experience when building Vue apps.

Core Concepts of Pinia

1. **Getters** are responsible for extracting and returning specific values ??from the store. By defining getters you can efficiently access & process data without having to directly tweak the underlying.
2. **Actions** play an important role in Pinia, allowing you to modify the state of the store by performing asynchronous or synchronous operations.
3. **The Store** is the 'heart' of Pinia, encapsulating the app's state, getter, actions, and mutations.
4. **State** refers to the data that the store manages.

What is Vuex?



Vuex is well known as a powerful solution for state management. So Pinia compared to Vuex, which option is more outstanding?

Pinia uses Vue's reactive system for state management, avoiding the need for any external dependencies. That means the Pinia ecosystem is more centralized and avoids potential long code bugs. Plus, the TypeScript support & tight typing it provides, allows you to catch bugs early in development, leading to more complete applications with fewer bugs.

If you're starting a new Vue project or considering a Vuex migration, Pinia is an attractive alternative that streamlines state management without sacrificing flexibility or performance.

Build a simple Vue app using Pinia

To better understand Pinia, try building a basic to-do list management app. Pinia provides the tools you need to manage the state for such applications.

Necessary conditions

First, prepare the required environment for this project, starting with Vue CLI.

```
# ?? cài Vue CLI: npm install -g @vue/cli # ?? t?o th? m?c d? án v?  
i Vue CLI: vue create pinia-todo-app
```

At this stage, you can see in the terminal you need to select a preset. Continue by selecting **Vue 3** from the default settings.

You can now install Pinia in the project directory:

```
cd pinia-todo-app npm install pinia
```

File Setup

You only need to edit a few files to complete this sample project. First, create a file named **store.js** in the **src** directory . This file will contain, add, and remove items you add to your to-do list. It will do all this using core Pinia concepts.

```
// src/store.js import { defineStore } from "pinia"; export const useTodoStore =
```

But, of course, this file alone is not enough. You need to link the **store.js** file with **App.vue** . To do that, change the **src/App.vue** file as follows:

```
// src/App.vue
```

The logic here is quite simple. The events you define in **store.js** will occur as you specify in the **App.vue** file . It's like the **App.vue** file will act as your control mechanism.

Of course, this way the app isn't everything because you don't use templates or styles. To add them, you can edit the **App.vue** file as follows:

```
// src/App.vue
```

My Pinia Todo List

1. {{ todo.text }} Done!

Now you have everything you need. Using the core concept features in Pinia, you can check the status, add & remove new items from the to-do list as you wish. But there is one last step. You should update the **main.js** file in the **src** directory and include the necessary directories for the application.

```
// src/main.js import { createApp } from "vue"; import { createPinia } from "pinia";
```

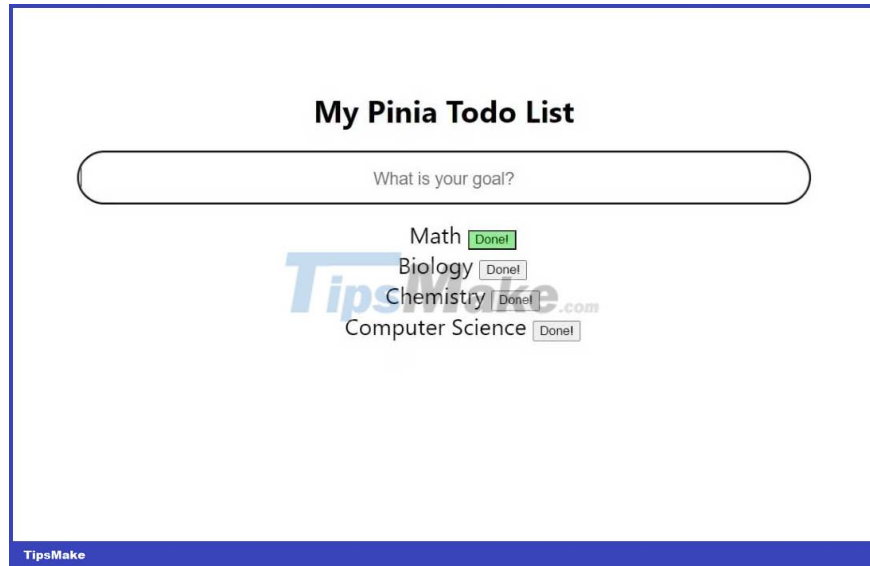
Everything needed to test the application is ready.

Run the sample Pinia app

To see the results of this simple but instructive to-do list, use the following command:

```
npm run serve
```

You'll see a clean, simple interface for adding and removing tasks:



Overall, Pinia is a valuable tool when you need a smooth and efficient data flow in your application. By providing a modular, secure, responsive architecture, Pinia simplifies state management and increases operational efficiency.

You finished reading the article "**Everything you need to know about Pinia in Vue.js**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.