

微店技术周刊(总第 25 期)-本期主编-张尚金



历史上的今天都发生了什么科技大事？

1963年7月26日，美国发射了世界上第一颗地球同步轨道通讯卫星“辛康2号”

关于

微店技术周刊整理每周有趣有用有料的各类内容，为大家呈现多样的技术世界。

周刊编辑：@何会会 / @张尚金 / @杨力 / @赵兴 / @刘远洋

周刊招募内容作者，主题不限，向团队发出你的声音吧！

泛前端

- [随机不只是 Math.random —— 前端噪声应用](#) (推荐者: @蒋涛)

将图形学领域的噪声应用到前端里面，没想到也能实现很多意想不到的随机动画效果，可以在一些营销、促销活动业务里面尝试

- [HTTP2 打包与资源合并策略, 文章一](#) (推荐者: @龙佳)
- [HTTP2 打包与资源合并策略, 文章二](#) (推荐者: @龙佳)
- [css margin 的一些你可能不知道的细节](#) (推荐者: @张尚金)
- [Sketch 大家肯定都用过, 但如何写一个 Sketch 插件呢?](#) (推荐者: @张尚金)
- [javascript 能力如何提升? 有效方法之一就是去读优秀库的源码](#) (推荐者: @张尚金)
- [A complete guide to the Node.js event loop - LogRocket Blog](#) (推荐者: @张尚金)

对 node.js 中的 event loop 的一次全面介绍

- [A simple guide to help you understand closures in JavaScript](#) (推荐者: @张尚金)

通过更简单的方式来理解 js 中的闭包

- [The 3 Types of Design Patterns All Developers Should Know](#) (推荐者: @张尚金)

设计模式大概有 26 种之多，完全掌握有点难，但这 26 种大概可以归纳为 3 种类型的设计模式，分别就是创建型、结构型、行为型，具体细节可以看文章。

- [Introducing Hooks – React](#) (推荐者: @张尚金)

react 官方对 hooks 的一次全面介绍，比如 hooks 诞生的动机，如何去解决的等等，值得一看

- [微信小程序运行流程看这篇就够了](#) (推荐者: @刘远洋)

小程序运行流程简明介绍

- [开发做了这么多年, 你真的了解 JS 工作机制吗](#) (推荐者: @刘远洋)

- [阿里云前端技术周刊](#)(推荐者: @刘远洋)
- [双精度浮点数](#)(推荐者: @刘远洋)
- [Mobx 思想的实现原理, 及与 Redux 对比](#)(推荐者: @刘远洋)
- [Taro 多端开发实现原理与项目实战](#)(推荐者: @刘远洋)
- [中国古诗词库, 很赞!](#)(推荐者: @刘远洋)
- [Proxy 的巧用](#)(推荐者: @刘远洋)
- [21 个 VSCode 快捷键, 让代码更快, 更有趣 - 终身学习者](#)(推荐者: @刘远洋)
- [为什么视频网站的视频链接地址是 blob?](#)(推荐者: @刘远洋)
- [hexo 主题: 三段式设计, 极简, 方便](#)(推荐者: @刘远洋)
- [webpack 原理](#)(推荐者: @刘远洋)
- [VIO 出品 - vChart 图表基础组件](#)(作者: @蒋迪 - 生产力平台)

vChart 的目标是用最简单的方式作图。

程序人生

- [稻盛和夫告别演讲: 人生因思维方式而改变](#)(推荐者: @张尚金)
- [《逃离北上广》各城市薪资水平统计](#)(推荐者: @罗炜)
- [程序员养生指南?](#)(推荐者: @罗炜)

投稿

Vuex源码解析

@罗炜 - 生产力平台

原文链接: <https://github.com/luoway/blog/issues/9>

正文部分

Vuex 源码解析

[Vuex 是什么？](#)

Vuex 使用方式

1. 安装 Vuex

```
import Vue from "vue";
import Vuex from "vuex";
Vue.use(Vuex);
```

1. 创建 store

```
const store = new Vuex.Store({
  //store 内容
});
```

1. 注入 store

```
const app = new Vue({
  store
  //vue 实例配置
});
```

接下来按照使用方式的步骤进行分析。

Vuex.install——安装 Vuex

[src/index.js](#)

```
import { Store, install } from "../store";
//import ...
export default {
  Store,
  install,
  version: "__VERSION__",
  mapState,
  mapMutations,
  mapGetters,
  mapActions,
  createNamespacedHelpers
};
```

根据[Vue.use\(\)](#)用法说明, `Vue.use(Vuex)` 执行时会调用 `Vuex.install()`, 内容如下:

[src/store.js](#)

```
import applyMixin from "../mixin";
//...
let Vue; // bind on install
//...
export function install(_Vue) {
  if (Vue && _Vue === Vue) {
    //省略开发环境提示
    return;
  }
  Vue = _Vue;
  applyMixin(Vue);
}
```

再看 `applyMixin(Vue)`

[src/mixin.js](#)

```
export default function(Vue) {
  //省略Vue 1.x、2.x版本区分
  Vue.mixin({ beforeCreate: vuexInit });

  /**
   * Vuex初始化钩子，注入到每个实例的初始化钩子列表。
   */

  function vuexInit() {
    const options = this.$options;
    // store injection
    if (options.store) {
      this.$store =
        typeof options.store === "function" ? options.store() : options.store;
    } else if (options.parent && options.parent.$store) {
      this.$store = options.parent.$store;
    }
  }
}
```

执行 `Vue.use(Vuex)` 目的就是在每个 Vue 实例初始化时, 往 `beforeCreate` 生命周期中加入 `vueInit` 函数。

在“注入 store”这一步:

```
const app = new Vue({
```

```
store
//vue 实例配置
});
```

执行 Vue 实例的 `beforeCreate` 生命周期，`vueInit` 函数被执行，运行时 `this` 指向当前 vue 实例上下文，因此 `this.$options` 指向当前 vue 组件配置对象

配置对象是指，`new Vue(options)` 时传入的 `options`，实例内可以通过 `this.$options` 访问到

传入的 `store` 可以被 `options.store` 访问到，此时执行

```
this.$store =
  typeof options.store === "function" ? options.store() : options.store;
```

将 `options.store` 内容赋值给 `this.$store`，即完成了 `store` 的注入。

而 `options.store` 不存在时，即没有给当前组件配置 `store` 时，`vueInit` 会尝试访问父组件已经注入完成的 `$store`，若存在则赋值给当前组件 `this.$store`，实现组件树内共享 `store`。

至此，Vuex 的安装、注入看完了，接下来具体看创建 Store。

Vuex.Store(options)——创建 Store

[src/store.js](#)

```
export class Store {
  constructor() {}

  get state() {}
  set state(v) {}
  commit() {}
  dispatch() {}
  //...
}
```

Vuex.Store 是一个 ES6 class，JS 里是没有 Class 数据类型的，本质上还是个构造函数。

实例化过程

.....

[查看全部文章内容](#)