# HTML5新特性 -- Unit04

# 1.Vuex

## 1.1 什么是Vuex?

Vuex是一个**专门为Vue应用程序开发的状态管理模式**，它采用**集中式的管理方式来管理Vue组件中共享的状态（数据）**。

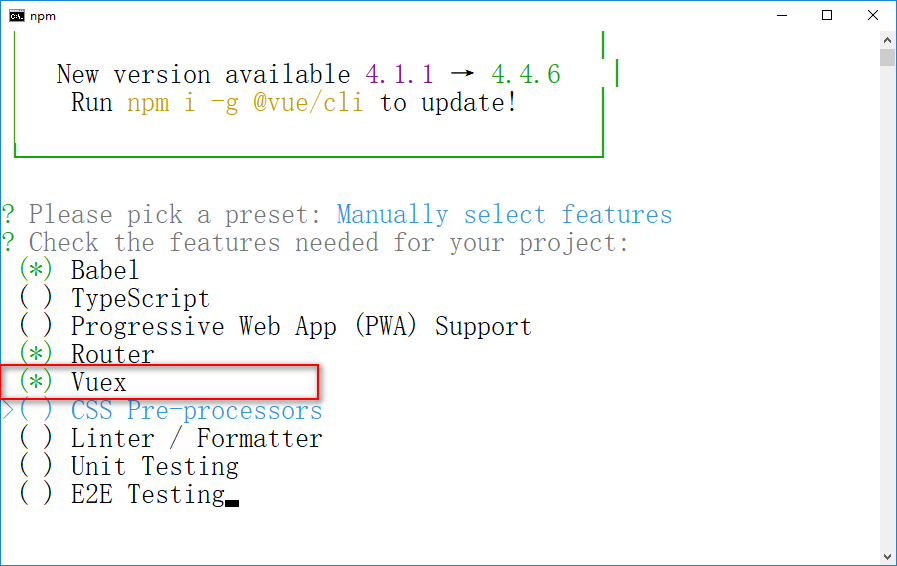
Vuex的应用场景：

· 用户登录的状态

· 购物车的状态等

## 1.2 安装Vuex

方式1：在通过**vue create**命令创建脚手架项目时，选择Vuex插件



方式2：通过npm进行安装

**npm install --save vuex**

Vuex本质就是Vue的一个插件

**Vuex建议存储到store目录下的index.js中**

## 

## 1.3 基本用法

**Vuex的核心是Store**，其就是一个**容器**，包含了**Vue应用中的状态**。其基本结构是：

export default new Vuex.Store({   
 state: {   
 },   
 mutations: {   
 },   
 actions: {   
 },   
 modules: {   
 }   
})

### · state

**state**定义了**应用状态的数据结构**，其**数据类型**可以为**string、number、boolean、array、object等**。示例代码如下：

state: {   
 username:'Tom',   
 age:23,   
 sex:true,   
 products:[   
 {   
 id:1,   
 productName:'商品1',   
 salePrice:2389.99   
 },   
 {   
 id:3,   
 productName:'商品2',   
 salePrice:6999.99   
 }   
 ]   
}

### · Vue组件访问Vuex中的状态

this**.$store.state.xxx**

示例代码如下：

<template>   
 <div>   
 <h1>访问Vuex的数据--页面1</h1>

**<!-- this 可加可不加，但是脚本里面必须加，不然报错 -->**

<p>姓名:{{$store.state.username}}</p>   
 <p>年龄:{{this.$store.state.age}}</p>   
 <p>性别:{{$store.state.sex ? '男' : '女'}}</p>   
 </div>   
</template>   
<script>   
 export default {   
 mounted(){   
 console.log(this.$store.state.age);   
 }   
 }   
</script>

### · getters

Vuex允许在store中定义getter(可以认为是**store的计算属性**)，getter的**返回值会自动缓存**，只有其依赖的值发生了变化才会被重新计算，其语法结构是：

getters:{   
 //获取商品的数量 -- **这些方法被认为是state的计算属性**  //**state参数自动代表当前store的state(其实名称可以为任意合法名称)**   
 productNum(state){   
 return state.products.length;   
 }   
},

<h2>商品列表中有**{{this.$store.getters.productNum}}**件,具体如下:</h2>

### ·

### mutations

**mutations**是用于**改变状态的方法**，也是**唯 一修改state的推荐方法**，示例代码如下：

在day04的代码中，分别用两个实例代码演示，

changeage.vue——增加年龄和删除年龄；

addproduct.vue——增加商品

## · 在页面组件中调用mutations中的方法

//payload意为载荷，意思是参数，其类型可以是任何数据类型

this**.$store.commit('方法名称',[payload])**

**方法名记得加引号**

# 

# 2.WebStorage

分类sessionStorage和localStorage两种存储机制。

length属性

setItem()方法

getItem()方法

removeItem()方法

clear()方法