

Building Custom Vue.js Services and Plugins

Kevin Griffin
@1kevgriff

About Me



Kevin Griffin

10x Microsoft MVP
ASP.NET Core, Azure, and Web

consultwithgriff.com

twitter.com/1kevgriff
twitch.tv/1kevgriff

Say nice things about me on twitter.com/1kevgriff



All details for THIS talk at:

<https://consultwithgriff.com/vue-services-plugins/>

Say nice things about me on twitter.com/1kevgriff

Components are Self-Contained

```

○○○

<template>
  <div class="hello">
    <h1>{{ msg }}</h1>
  </div>
</template>

<script lang="ts">
import { Component, Prop, Vue } from 'vue-property-decorator';

@Component
export default class HelloWorld extends Vue {
  @Prop() private msg!: string;
}
</script>

```

Say nice things about me on twitter.com/1kevgriff

Components Can Do Stuff

```

○○○

<script lang="ts">
import { Component, Prop, Vue } from 'vue-property-decorator';
import axios, { AxiosResponse } from 'axios';

@Component
export default class HelloWorld extends Vue {
  @Prop() private msg!: string;

  loadFromApi(){
    axios.get("https://consultwithgriff.com")
      .then((res)=> this.gotData(res));
  }

  gotData(data : AxiosResponse){
    // yay
  }
}
</script>

```

Say nice things about me on twitter.com/1kevgriff

What If Components Need The Same Stuff?

○○○

```
<script lang="ts">
import { Component, Prop, Vue } from 'vue-property-decorator';
import axios, { AxiosResponse } from 'axios';
```

```
@Component
export default class HelloWorld extends Vue {
  @Prop() private msg!: string;

  loadFromApi(){
    axios.get("https://consultwithgriff.com")
      .then((res)=> this.getData(res));
  }

  getData(data : AxiosResponse){
    // yay
  }
}
```

○○○

```
<script lang="ts">
import { Component, Prop, Vue } from 'vue-property-decorator';
import axios, { AxiosResponse } from 'axios';
```

```
@Component
export default class AnotherComponent extends Vue {
  @Prop() private somethingSomething!: string;

  loadFromApi(){
    axios.get("https://consultwithgriff.com")
      .then((res)=> this.getData(res));
  }

  getData(data : AxiosResponse){
    // yay
  }
}
```

Let's Install The Plugin!

○○○

```
import { ServicePlugin } from "../servicePlugin";

const settings: ServiceOptions = {
  isEnabled: process.env.NODE_ENV !== "production"
};

Vue.use(ServicePlugin, settings);

new Vue({
  render: h => h(App),
}).$mount('#app')
```

Say nice things about me on twitter.com/1kevgriff

Adding Instance Objects/Methods/Properties

○○○

```
export const ServicePlugin: PluginObject<OptionsObject> = {
  install(Vue: VueConstructor<Vue>,
    options: OptionsObject | undefined) {
    Vue.prototype.$service = new Service(options);
  }
}
```

○○○

```
someMethod(){
  this.$service.serviceMethod();
}
```

Adding Global Objects/Methods/Properties

○○○

```
export const ServicePlugin: PluginObject<OptionsObject> = {
  install(Vue: VueConstructor<Vue>,
    options: OptionsObject | undefined) {
    Vue.$service = new Service(options);
  }
}
```

○○○

```
someMethod(){
  Vue.$service.serviceMethod();
}
```

Custom Components

○ ○ ○

```
import { ServiceComponent } from './serviceComponent.vue';

export const ServicePlugin: PluginObject<OptionsObject> = {
  install(Vue: VueConstructor<Vue>, options: OptionsObject | undefined) {

    // creates <service-component /> component
    Vue.component('service-component', ServiceComponent);

  }
}
```

Say nice things about me on twitter.com/1kevgriff

Mixins

○ ○ ○

```
export const ServicePlugin: PluginObject<OptionsObject> = {
  install(Vue: VueConstructor<Vue>, options: OptionsObject | undefined) {

    Vue.mixin({
      created() {
        Vue.$logger.warn(
          'Catching the component creation event for all components'
        )
      },
    });

  }
}
```

Say nice things about me on twitter.com/1kevgriff

Less Talk, More Code

Let's build the "logger" example

Thanks!



Kevin Griffin

twitter.com/1kevgriff

twitch.tv/1kevgriff

consultwithgriff.com