**掛載區域JS切割:**

1. 將此頁面，所需用到的到component及父組件設置寫在獨立的js中，如下:

**home.js**

Vue.component('full-in-vue', require('./components/FullInVue.vue').default);

Vue.component('blade-call-vue', require('./components/BladeCallVue.vue').default);

const app = new Vue({

    el: '#app',

    data: { }

});

1. 然後將此JS放到Webpack裡面，執行npm run dev將js打包，如下:

webpack.mix.JS

mix.js('resources/js/home.js', 'public/js');

1. 所需blade引用此js，並建立作用域，如下:

@extends('dashboard.base')

@section('content')

  <div id="app">  {{--建立作用域--}}

     <full-in-vue><full-in-vue>

  </div>

@endsection

@section('javascript')

<script src="{{ asset('js/home.js') }}"></script>

endsection

**補充:**

Blade可以引用多個components的js，但是因為每個js裡都是一個父組件，所以作用域不可以重疊，如下圖:

{{-- 正確方法 --}}

@extends('dashboard.base')

@section('content')

    <div id="app"> {{--建立作用域--}}

        <full-in-vue><full-in-vue>

    </div>

    <div id="app2">{{--建立作用域--}}

        <helf-in-vue><helf-in-vue>

    </div>

@endsection

@section('javascript')

<script src="{{ asset('js/app.js') }}"></script>

<script src="{{ asset('js/app2.js') }}"></script>

@endsection

{{-- 錯誤方法 --}}

@extends('dashboard.base')

@section('content')

    <div id="app"> {{--建立作用域--}}

        <full-in-vue><full-in-vue>

        <div id="app2">{{--建立作用域--}}

            <helf-in-vue><helf-in-vue>

        </div>

    </div>

@endsection

@section('javascript')

<script src="{{ asset('js/app.js') }}"></script>

<script src="{{ asset('js/app2.js') }}"></script>

@endsection

**Components撰寫方法(以modal為例):**

**全包法:**

<template>

<div>

<!-- modal button -->

    <button type="button"

        class="btn btn-primary"

        data-toggle="modal"

        data-target="#exampleModalLong"

        @click="getItems"

    >

  <!-- @click="getItems" 使用vue的方法 -->

        full-in-vue

    </button>

     <!-- modal view -->

<div class="modal fade" id="exampleModalLong" tabindex="-1" role="dialog" aria-labelledby="exampleModalLongTitle" aria-hidden="true">

        <div class="modal-dialog" role="document">

            <div class="modal-content">

            <div class="modal-header">

                <h5 class="modal-title" id="exampleModalLongTitle">Modal title</h5>

                <button type="button" class="close" data-dismiss="modal" aria-label="Close">

                <span aria-hidden="true">&times;</span>

                </button>

            </div>

            <div class="modal-body">

                {{msg}} <!--使用vue的資料 -->

            </div>

            <div class="modal-footer">

                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

                <button type="button" class="btn btn-primary">Save changes</button>

            </div>

            </div>

        </div>

    </div>

</div>

</template>

<script>

    export default {

        data () {

            return {

                msg: 'This is a Vue full-in-vue.'

            }

        },

        methods: {

            getItems() {

                console.log(this.msg)

            }

        }

    }

</script>

**引用法:**

**home.blade.php**

@extends('dashboard.base')

@section('content')

  <div id="app">  {{--建立作用域--}}

     <full-in-vue ref="appcomponent"><full-in-vue> {{--在component中建立引用--}}

  </div>

@endsection

@section('javascript')

<script src="{{ asset('js/home.js') }}"></script>

endsection

**home.js**

Vue.component('full-in-vue', require('./components/FullInVue.vue').default);

Vue.component('blade-call-vue', require('./components/BladeCallVue.vue').default);

const app = new Vue({

    el: '#app',

    data: { } ,

    methods: {

        callChildGetItem : function() {

            this.$refs.appcomponent.getItems()

        }

    }

});

**FullInVue.vue**

<template>

    <div>

        ...

    </div>

</template>

<script>

    export default {

        data () {

            return {

                msg: 'This is a Vue full-in-vue.'

            }

        },

        methods: {

            getItems() {

                console.log(this.msg)

            }

        }

    }

</script>

home.blade.php

<div id="app">

   <full-in-vue ref="appcomponent" :data='123'></full-in-vue>

   <button type="button" @click="callChildGetItem">

    Get Item

   </button>

</div>

**Blade傳值進Vue**

<div id="app">

      <full-in-vue ref="appcomponent" :youDataName='123'></full-in-vue>

</div>

<template>

    <div>

       {{ youDataName }}

    </div>

</template>

<script>

    export default {

        props:['youDataName'],

        data () {

            return {

                msg: 'This is a Vue full-in-vue.'

            }

        },

        methods: {

            getItems() {

                console.log(this.msg)

            }

        }

    }

</script>