

Forum:[Hack Forums](#)

Thread: [Trying to create a 'modular' structure in my project](#)

VirtualFrost - January 12, 2020, 5:10 pm

I'm starting a new project and want to try create this kind of 'modular' style to it. So far I've just used promises to load each file and was looking for some input into if there is any way to improve this.

Basically the app will start, and it will start loading each module, it should load each app, if any fail then it'll output the error and then terminate the process. If there's anything I could do better or maybe a simpler way I'd love the input

Below is the basic structure

app.js

/modules/index.js

/modules/test1.js

/modules/test2.js

app.js

Code global.server = {};

console.log('Starting test')

require('./modules/index.js');

modules/index.js

Code server.test1 = require('./test1.js');

server.test2 = require('./test2.js');

console.log('Loading modules');

function loadModules(){

 Promise.all([server.test1.start(), server.test2.start()]).then(function(values){

 console.log(values);

 }).catch((e) => { console.log(e)});

}

loadModules();

test1.js

Code module.exports = {

 'start': function(){

 return new Promise((resolve, reject) => {

 let test = Math.random();

 setTimeout(() => {

 if(test >= 0.1){

 resolve('Test 1 Complete');

 } else {

 reject('Test 1 Failed');

 }

 }, 3000);

 });

```
}  
}
```

Ignore the code inside test1.js, it's all for testing, all of this code works fine as is and will successfully return each resolve when they've all done loading. Only looking to see if this structure and loading can be done any easier/simpler than what I've done here.

doon - January 13, 2020, 10:45 am

Nice I like this pattern. I use a similar pattern for having multiple services + adding new services needed for the app.

One thing I do since someone could add a new module is load modules from a config somewhat like this.

PHP Code

```
<?php  
const config = {  
  modules: [  
    {  
      name: 'Module 1',  
      path: './modules/module1.js'  
    },  
    {  
      name: 'Module 2',  
      path: './modules/module2.js'  
    }  
  ]  
}
```

Then load the services like this

PHP Code

```
<?php  
const Config = require('<root_path>/config');  
Config.modules.forEach((module) => {  
  require(module.path).start().then(() => {  
    console.log(`${module.name} has been started.`);  
  }).catch(() => {  
    console.log(`${module.name} failed to start.`)  
  });  
})
```

Not sure if this could be relevant to your project but works well for adding modules easily. FYI I use app-root-path for getting root app path to avoid having to ../../ directories.

DrDoof - January 13, 2020, 10:57 am

Code const fs = require('fs')

```
const modules = fs.readdirSync('modules').map(moduleName =>  
require(`./modules/${moduleName}`))
```

Promise.all(modules.map(module => module.start())).then(console.log).catch(console.warn)
Why not just have it load all "modules" in the directory rather than having to add new files to a list/config file?

VirtualFrost - January 14, 2020, 4:31 pm

(Jan 13, 2020 11:57 AM)DrDoof Wrote: [\[CITE\]](#) Code const fs = require('fs')

```
const modules = fs.readdirSync('modules').map(moduleName =>  
require(`./modules/${moduleName}`))
```

Promise.all(modules.map(module => module.start())).then(console.log).catch(console.warn)

Why not just have it load all "modules" in the directory rather than having to add new files to a list/config file?

Actually this is interesting, I might play around with how this is done. I did want to avoid having to manually write in each file name into a list as I know it's just annoying to do. I appreciate this a lot!