第七节 Fis3

1.定义

定制的前端工具构建工具

解决前端开发中自动化工具、性能优化、模块化框架、开发规范、代码部署、开发流程等问题

官网http://fis.baidu.com/fis3/index.html

但是不维护了,仅限于学习参考

2.特点

3.基本使用

3.1 安装FIS3

```
1 yarn global add fis3
```

3.2 构建发布到项目目录

```
1 fis3 release -d ./output
```

构建过程中对资源 URI 进行了替换，替换成了绝对 URL。通俗点讲就是相对路径换成了绝对路径。
这是一个 FIS 的很重要的特性，**资源定位**。
release 之前

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content=
"width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content=
"ie=edge">
  <title>FIS Sample</title>
  <link rel="stylesheet" href="css/style.scss">
</head>
<body>
  <h1>FIS Sample</h1>
  <img src="img/icon.png" alt="logo">
  <button>Click me~</button>
  <script src="js/app.js"></script>
</body>
</html>
```

release 之后

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content=
"width=device-width, initial-scale=1.0" />
        <meta http-equiv="X-UA-Compatible" content="ie=edge" />
        <title>FIS Sample</title>
        <link rel="stylesheet" href="/03-16-fis-sample/css/style.scss"
 />
    </head>
    <body>
        <h1>FIS Sample</h1>
        <img src="/03-16-fis-sample/img/icon.png" alt="logo" />
        <button>Click me~</button>
        <script src="/03-16-fis-sample/js/app.js"></script>
    </body>
</html>
```

3.3 配置文件

默认配置文件为 `fis-conf.js`，FIS3 编译的整个流程都是通过配置来控制的。FIS3 定义了一种类似 CSS 的配置方式。固化了构建流程，让工程构建变得简单。

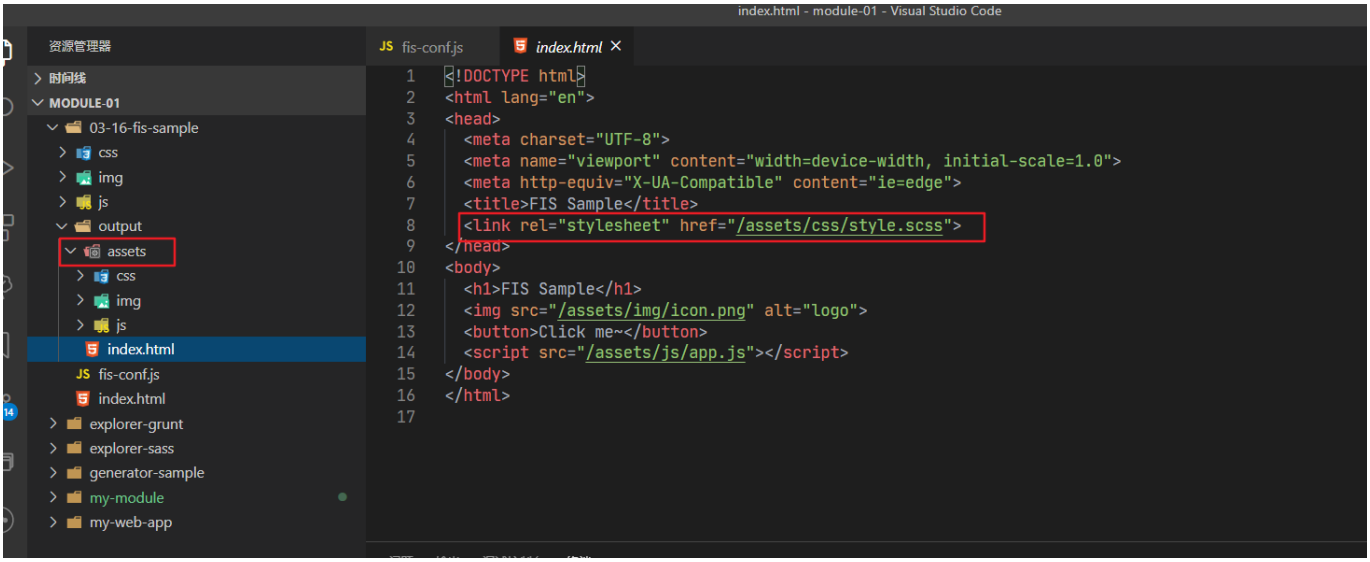
fis.match()

首先介绍设置规则的配置接口

```
fis.match(selector, props);
```

- `selector` ： FIS3 把匹配文件路径的路径作为selector，匹配到的文件会分配给它设置的 `props`。关于 selector 语法，请参看 Glob 说明
- `props` ： 编译规则属性，包括文件属性和插件属性

```
fis.match('*.{js,scss,png}',{
    release: '/assets/$0'   // $0就是当前文件原始的目录结构
})
```



3.4 片段编译

例如对scss文件进行编译

安装scss转换插件

```
yarn global add fis-parser-node-sass
```

配置文件

```
fis.match('**/*.scss', {
    rExt: '.css', // 转换之后的扩展名
    parser: fis.plugin('node-sass'), // 指定插件
    optimizer: fis.plugin('clean-css'), // 压缩css
});
```

转换js

安装babel转换插件

```
yarn global add fis-parser-babel-6.x
```

```
fis.match('**/*.js', {
    parser: fis.plugin('babel-6.x'),     // fis3不更新了只能用babel6了
```

```
    optimizer: fis.plugin('uglify-js'),  // 混淆压缩代码
});
```

3.5 压缩资源

为了减少资源网络传输的大小，通过压缩器对 js、css、图片进行压缩是一直以来前端工程优化的选择。在 FIS3 中这个过程非常简单，通过给文件配置压缩器即可。

```
// 清除其他配置，只保留如下配置
fis.match('*.js', {
  // fis-optimizer-uglify-js 插件进行压缩，已内置
  optimizer: fis.plugin('uglify-js')
});

fis.match('*.css', {
  // fis-optimizer-clean-css 插件进行压缩，已内置
  optimizer: fis.plugin('clean-css')
});

fis.match('*.png', {
  // fis-optimizer-png-compressor 插件进行压缩，已内置
  optimizer: fis.plugin('png-compressor')
});
```