**零配置**

不使用任何配置，fis将支持：

1. 使用 **fis install <name>** 命令安装fis仓库提供的各种 **组件、框架、示例、素材、配置等** 开发资源。
2. 使用 **fis server start** 命令启动一个本地调试服务器，完美支持php后端程序
3. 使用 **fis release --dest ./output --md5** 命令为所有静态资源文件（js、css、图片、flash等）添加md5戳，并修改文件中的引用路径
4. 使用 **fis release --dest ./output --optimize** 命令对js、css、html、htm文件进行压缩（后续还会加上对图片的自动压缩）
5. 编译中对 **js**、 **css** 以及 **类html** ( *html, htm, xhtml, tpl, php, jsp, asp* ) 文件分别支持三种扩展语言能力：
   * **资源定位**：获取任何开发中所使用资源的线上路径；
   * **内容嵌入**：把一个文件的内容(文本)或者base64编码(图片)嵌入到另一个文件中；
   * **依赖声明**：在一个文本文件内标记对其他文件的依赖关系；

以上功能可满足传统前端开发所需的基本要求

**使用配置文件**

在项目目录下新建一个 **fis-conf.js** 文件，我们可以对fis的编译系统做各种定制化配置。配置fis系统的接口是：

fis.config.set(key, value);

或者

fis.config.merge({...});

由于fis自动化工具采用nodejs作为平台，因此其配置文件也是js书写的。fis变量是全局注册，config属性是fis系统的配置对象实例，merge或set方法用以合并配置数据。

**内置的默认配置**

由于fis系统是完全插件化的，因此fis.config对象会有一些内置配置用以为用户提供零配置下的基本功能，所以配置文件中使用fis.config.merge 或者 fis.config.set 接口来追加用户配置。而内部初始化的配置数是：

fis.config.init({

project **:** {

charset **:** 'utf8',

md5Length **:** 7

}

});

**项目配置**

**charset**

* 解释：指定项目编译后产出文件的编码。
* 值类型：string
* 默认值：'utf8'
* 用法：在项目的fis-conf.js里可以覆盖为
* fis.config.set('project.charset', 'gbk');

或者

fis.config.merge({

project **:** { charset **:** 'gbk' }

});

**md5Length**

* 解释：文件MD5戳长度。
* 值类型：number
* 默认值：7
* 用法：在项目的fis-conf.js里可以修改为
* fis.config.set('project.md5Length', 8);

或者

fis.config.merge({

project **:** { md5Length **:** 8 }

});

**md5Connector**

* 解释：设置md5与文件的连字符。
* 值类型：string
* 默认值：'\_'
* 用法：在项目的fis-conf.js里可以修改为
* fis.config.set('project.md5Connector ', '.');

或者

fis.config.merge({

project **:** { md5Connector **:** '.' }

});

**include**

* 解释：设置项目源码文件include过滤器。只有命中include的文件才被视为源码，其他文件则忽略。
* 值类型：Array | string | RegExp
* 默认值：无
* 用法：在项目的fis-conf.js里可以修改为
* fis.config.set('project.include', 'src/\*\*');

或者

fis.config.merge({

project **:** { include **:** 'src/\*\*' }

});

或者

fis.config.set('project.include', ['src/\*\*', /^\/vendor\//i]);

**exclude**

* 解释：设置项目源码文件exclude过滤器。如果同时设置了 [project.include](http://fis.baidu.com/docs/api/fis-conf.html#include) 和 project.exclude 则表示在include所命中的文件中排除掉某些文件。
* 值类型：Array | string | RegExp
* 默认值：无
* 用法：在项目的fis-conf.js里可以修改为
* fis.config.set('project.exclude', /^\/\_build\//i);

或者

fis.config.merge({

project **:** { exclude **:** /^\/\_build\//i }

});

或者

fis.config.set('project.exclude', ["dist/\*\*", /^\/\_build\//i]);

**fileType.text**

* 解释：追加文本文件后缀列表。
* 值类型：Array | string
* 默认值：无
* 说明：fis系统在编译时会对文本文件和图片类二进制文件做不同的处理，文件分类依据是后缀名。虽然内部已列出一些常见的文本文件后缀，但难保用户有其他的后缀文件，内部已列入文本文件后缀的列表为： [ **'css', 'tpl', 'js', 'php', 'txt', 'json', 'xml', 'htm', 'text', 'xhtml', 'html', 'md', 'conf', 'po', 'config', 'tmpl', 'coffee', 'less', 'sass', 'jsp', 'scss', 'manifest', 'bak', 'asp', 'tmp'** ]，用户配置会 **追加**，而非覆盖内部后缀列表。
* 用法：编辑项目的fis-conf.js配置文件
* fis.config.set('project.fileType.text', 'tpl, js, css');

或者

fis.config.merge({

project **:** {

fileType **:** {

text **:** 'tpl, js, css'

}

}

});

**fileType.image**

* 解释：追加图片类二进制文件后缀列表。
* 值类型：Array | string
* 默认值：无
* 说明：fis系统在编译时会对文本文件和图片类二进制文件做不同的处理，文件分类依据是后缀名。虽然内部已列出一些常见的图片类二进制文件后缀，但难保用户有其他的后缀文件，内部已列入文本文件后缀的列表为： [ **'svg', 'tif', 'tiff', 'wbmp', 'png', 'bmp', 'fax', 'gif', 'ico', 'jfif', 'jpe', 'jpeg', 'jpg', 'woff', 'cur', 'webp', 'swf', 'ttf', 'eot'** ]，用户配置会 **追加**，而非覆盖内部后缀列表。
* 用法：编辑项目的fis-conf.js配置文件
* fis.config.set('project.fileType.image', 'swf, cur, ico');

或者

fis.config.merge({

project **:** {

fileType **:** {

image **:** 'swf, cur, ico'

}

}

});