## 读取.yaml配置文件(Go)

*当前实现仅支持基础.yaml配置文件解析，不支持内容包括但不限于日期类型、注释、类型嵌套（map和list的相互嵌套）。 ——2018.12.20*

YAML是一种类似XML、JSON的标记性语言，一种较为广泛的用途是用来编辑配置文件，其扩展名有“.yaml”和“.yml”两种。

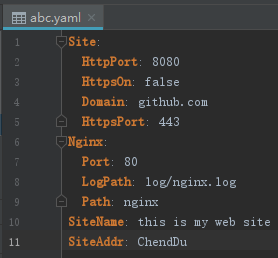


图1：.yaml配置文件示例

使用Go语言读取.yaml格式的配置文件中某个配置项的值，需要知道该配置项的名称、其值的类型（如int、string等）以及该配置项在整个配置文件中的从属关系。

首先，根据地址，使用一个map[interface { }]interface { }类型的映射将配置文件内容加载到程序中：

代码1：加载配置文件内容（附结果输出）

**type ConfigEngine struct {**

**data map[interface{}]interface{}**

**}**

**func test() {**

**c := ConfigEngine{}**

**err := c.Load("abc.yaml")**

**fmt.Println(****"c : ", c)**

**if err != nil { log.Println(err) }**

**}**

//fmt.Println("c : ", c) :

c : {map[SiteNumber:10010 SiteAddr:ChendDu Site:map[HttpPort:8080 HttpsOn:false Domain:github.com HttpsPort:443] Nginx:map[Port:80 LogPath:log/nginx.log Path:nginx]]}

从打印的结果可以看出，此时程序已经获得了配置文件的内容（保存在变量c中），可以从c中获取需要的配置值了（以HttpPort为例）：

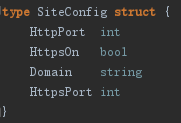


图2：根据已知文件结构定义的结构体

**sc := SiteConfig{}**

**s := c.GetStruct("Site", &sc)**

**switch s.(type) {**

**case SiteConfig:**

**sc = s.(SiteConfig)**

**}**

**fmt.Println("HttpPort : ", sc.HttpPort)**

// fmt.Println("HttpPort : ", sc.HttpPort)

HttpPort : 8080

代码2：获取目标配置项的值（附结果输出）

以下是程序中能够获取的其他类型配置值：

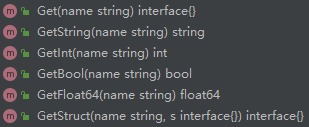


图3：程序能够获取的配置值类型