

# 第二周进度报告

C-nginx测试与开发

# UI模块

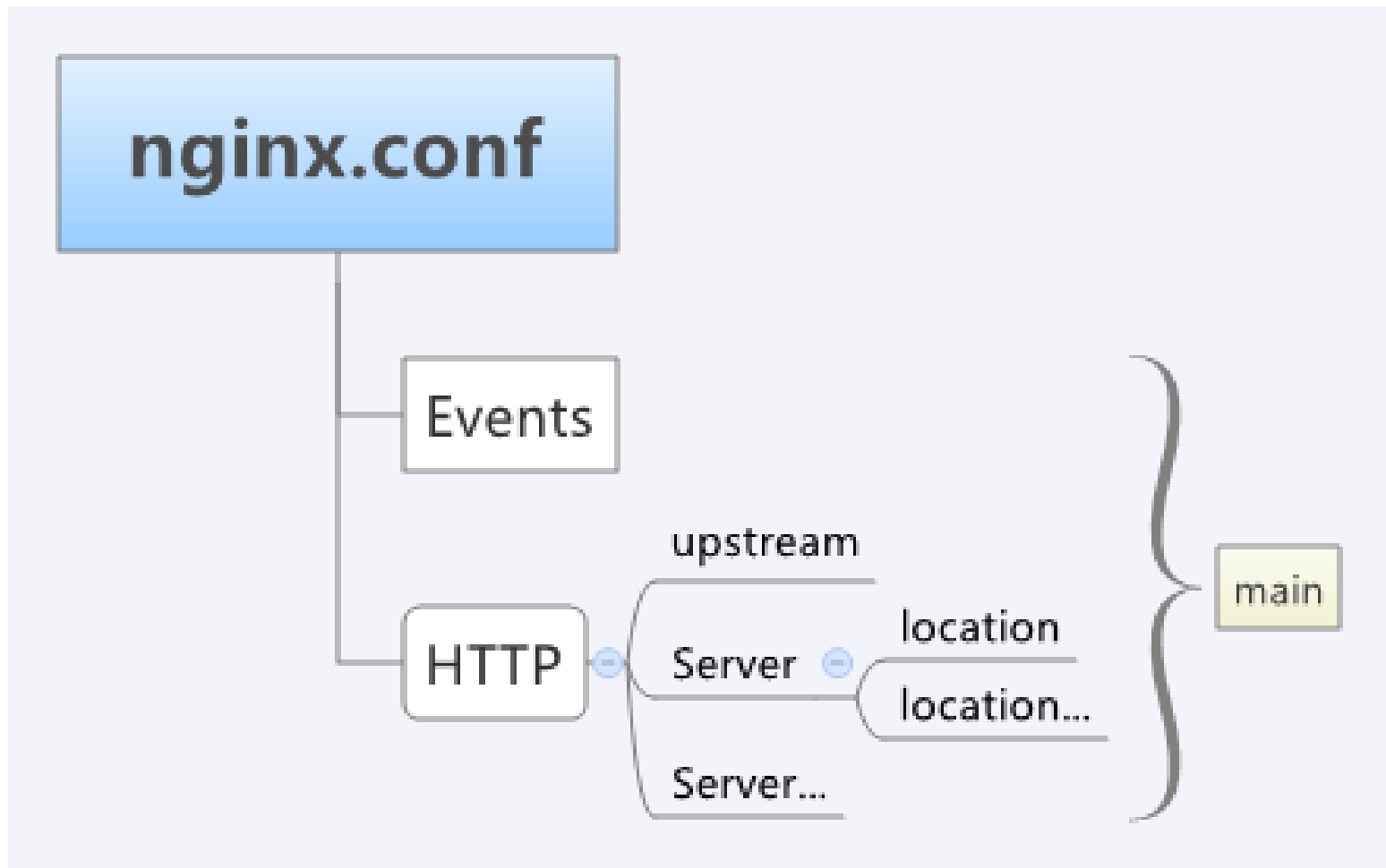
7% Nginx配置工...

—□×

open save about test

用户组	
workerProcess	
pidPath	
rootPath	
logPath	
maxConn	
errorPage	

# nginx.conf



# 已支持的配置项

```
sensetime@sensetime:~/soft/16TeamC/6.code/conf$ ./conf.py
sensetime@sensetime:~/soft/16TeamC/6.code/conf$ cat nginx.conf
user nobody nogroup;
worker_processes 4;
pid /var/run/nginx.pid;
error_log /var/log/nginx/error.log debug;
worker_rlimit_nofile 1024;
events {
    use epoll;
    worker_connections 1;
}
http {
    server {
        listen 80;
        server_name www.example.com ;
        index index.html index.htm ;
        root /var/www;
    }
}
```

# 配置项选择 v 0.1

配置项	默认值	配置描述
user	nobody nogroup	访问进程以系统中某个用户名启动
worker_processes	4	服务器并发线程个数
pid	./nginx.pid	进程文件
error_log	./nginx/error.log info	LOG输出路径，级别
nofile	1024	进程最多打开文件个数
events	-	事件配置
events.use	epoll	事件驱动方式
events.worker_connections	65535	单进程最大连接数
http	-	http服务器配置
http.server	-	虚拟服务器
http.server.port	80	端口
....		

# 配置结构

- user
- worker\_processes
- pid
- error\_log
- worker\_rlimit\_nofile
- events
  - use
  - worker\_connections
- http
  - server
    - Listen
    - Server\_name
    - Index
    - root

```
$ ls -lha
16:55 .
14:51 ..
16:26 conf.py
14:37 error_log.py
14:49 events.py
16:38 http.py
14:51 __init__.py
16:38 nginx.conf
14:37 pid.py
16:37 server.py
15:53 user.py
14:48 worker_rlimit_nofile.py
14:42 work_process.py
```

# 配置正确性检验

- 每一个配置类都有一个isValid()方法检验是否配置正确
  - 如：文件路径满足
    - `dirname = os.path.dirname(self.getPath())`
    - `return os.path.exists(dirname)`
  - 端口必须满足：
    - `self.getListen() in range(0,65536)`
  - 事件驱动模型选项：
    - `__use_set = ["kqueue","rtsig","epoll","/dev/poll","select","poll"]`
    - `return self.getUse() in __use_set`
- `isValid_all()`递归调用每个子配置的`isValid()`，当所有的配置都正确时，返回True

# 文件IO

- `conf.writeToFile()`
  - 写回配置到文件中
- `conf.setUpFromFile()`
  - 从文件中恢复配置



# To Do

- 友好的错误提示
- 支持更多的配置
- 提供更多基于机器性能的推荐配置
- 支持更多操作系统