

给 API 增加启动脚本

本节核心内容

- 如何管理 apiserver 启动命令，包括启动、重启、停止和查看运行状态

本小节源码下载路径：[demo13](#)

(https://github.com/lexkong/apiserver_demos/tree/master)

可先下载源码到本地，结合源码理解后续内容，边学边练。

本小节的代码是基于 [demo12](#)

(https://github.com/lexkong/apiserver_demos/tree/master) 来开发的。

为什么要添加启动脚本

通过添加服务器启动脚本可以很方便地启动、重启、停止和查看服务的状态。一些监控系统、发布系统需要有方法告诉它怎么启停和查看服务状态，这时候如果有个服务控制脚本就可以很方便地添加，要不然输入一堆启动参数不仅烦琐还容易出错。

添加启动脚本

apiserver 是通过 `admin.sh` 脚本来实现服务启动、重启、停止和查看服务状态操作的（详见 [demo13/admin.sh](#) (https://github.com/lexkong/apiserver_demos/blob/master/)

源码为：

```
#!/bin/bash

SERVER="apiserver"
BASE_DIR=$PWD
INTERVAL=2

# 命令行参数，需要手动指定
ARGS=""

function start()
{
    if [ "`pgrep $SERVER -u $UID`" != "" ];then
        echo "$SERVER already running"
        exit 1
    fi

    nohup $BASE_DIR/$SERVER $ARGS server
    &>/dev/null &

    echo "sleeping..." && sleep $INTERVAL

    # check status
    if [ "`pgrep $SERVER -u $UID`" == "" ];then
        echo "$SERVER start failed"
        exit 1
    fi
}

function status()
{
    if [ "`pgrep $SERVER -u $UID`" != "" ];then
```

```
        echo $SERVER is running
    else
        echo $SERVER is not running
    fi
}

function stop()
{
    if [ "`pgrep $SERVER -u $UID`" != "" ];then
        kill -9 `pgrep $SERVER -u $UID`
    fi

    echo "sleeping..." && sleep $INTERVAL

    if [ "`pgrep $SERVER -u $UID`" != "" ];then
        echo "$SERVER stop failed"
        exit 1
    fi
}

case "$1" in
    'start')
        start
        ;;
    'stop')
        stop
        ;;
    'status')
        status
        ;;
    'restart')
        stop && start
        ;;

```

```
*)  
echo "usage: $0 {start|stop|restart|status}"  
exit 1  
;;  
esac
```

看 shell 源码发现在 start 和 stop 时会 sleep 几秒，这是因为 API 服务器的启动需要时间去做准备工作，停止也需要时间去做清理工作。

编译并测试

1. 下载 apiserver_demos 源码包（如前面已经下载过，请忽略此步骤）

```
$ git clone  
https://github.com/lexkong/apiserver\_demos
```

2. 将 apiserver_demos/demo13 复制为
\$GOPATH/src/apiserver

```
$ cp -a apiserver_demos/demo13/  
$GOPATH/src/apiserver
```

3. 在 apiserver 目录下编译源码

```
$ cd $GOPATH/src/apiserver  
$ make
```

查看 admin.sh 用法

```
$ ./admin.sh -h
usage: ./admin.sh {start|stop|restart|status}
```

查看 apiserver 运行状态

```
$ ./admin.sh status
apiserver is not running
```

启动 apiserver

```
$ ./admin.sh start
sleeping...
```

查看 apiserver 状态

```
$ ./admin.sh status
apiserver is running
```

重启 apiserver

```
$ ./admin.sh restart
sleeping...
sleeping...
```

停止 apiserver

```
$ ./admin.sh stop
sleeping...
```

小结

本小结展示了如何用 `admin.sh` 去管理 API 服务器：启动、重启、停止和查看状态。该 `admin.sh` 命令在进行 `start`、`stop`、`restart` 和 `status` 操作时做了很多检查工作，以确保运行结果是正确的。