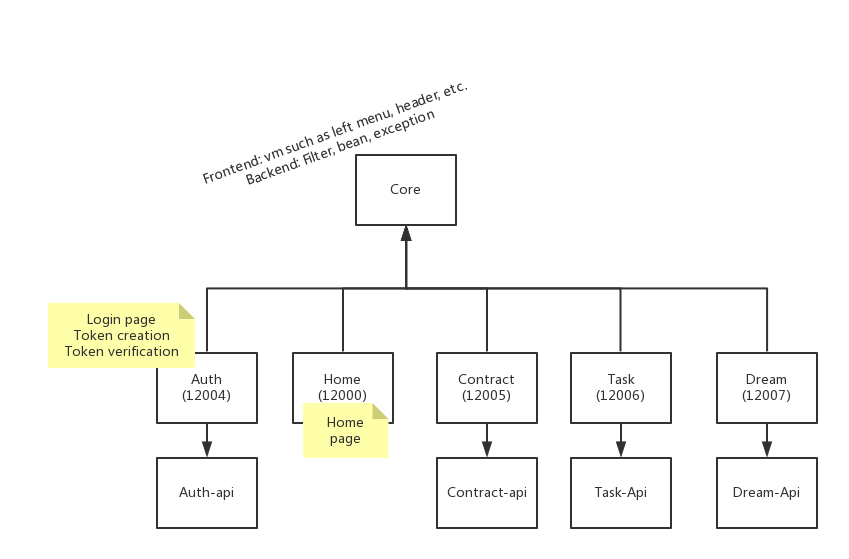
# Turtlebone-Parent配置

## 工程关系图



## 工程GIT

[git@github.com:ExpertsOfTurtle/TurtleboneParent.git](mailto:git@github.com:ExpertsOfTurtle/TurtleboneParent.git)

## 配置方向

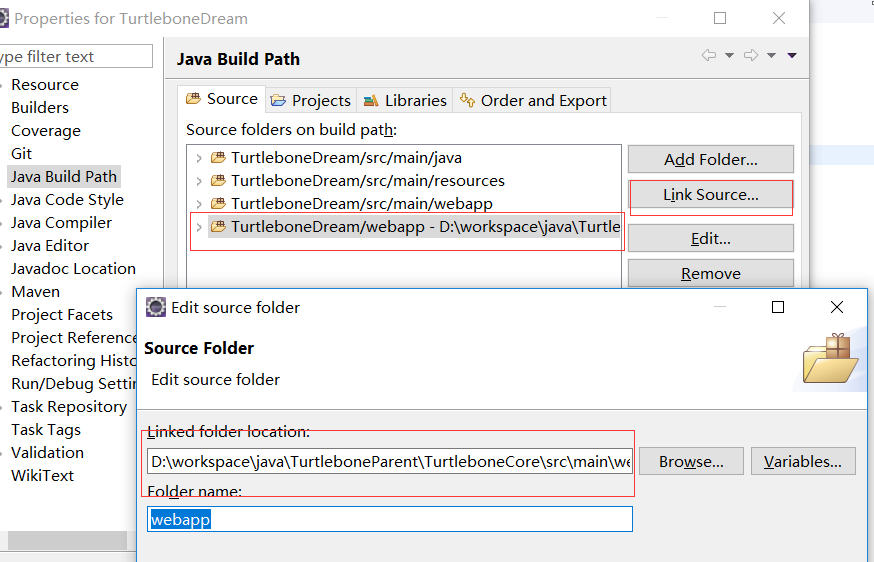
配置pom.xml – 为了在linux中执行mvn package打包成jar

配置工程依赖关系 – 为了在eclipse中编译/执行成功

**配置步骤：**

TurtleboneXXX工程依赖TurtleboneCore和TurtleboneXXX-api

TurtleboneXXX工程的Source设置中，Link Source指向**TurtleboneCore\src\main\webapp**和**TurtleboneCore\src\main\resources**，如下图：



## TurtleboneCore工程

### pom.xml

<dependency>

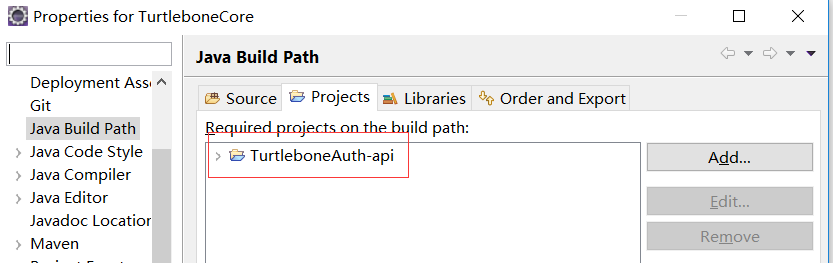
<groupId>com.turtlebone</groupId>

<artifactId>TurtleboneAuth-api</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>

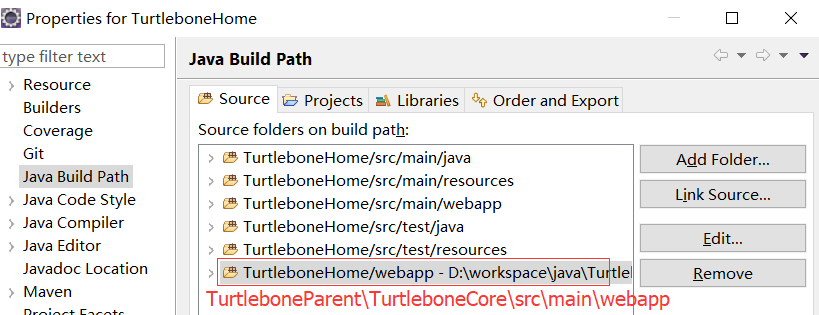
### Eclipse



## TurtleboneHome工程

主页所在的工程，没有实质内容

### Eclipse



## TurtleboneAuth工程

安全相关

提供了token的创建/校验/更新接口给TurtleboneCore的Filter中使用。

登陆页面

### pom.xml

<dependency>

<groupId>com.turtlebone</groupId>

<artifactId>TurtleboneAuth-api</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>

<dependency>

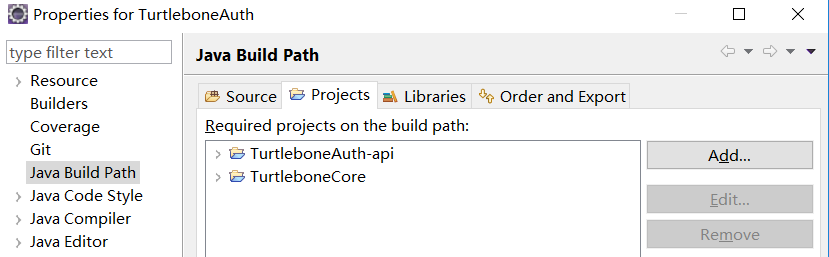
<groupId>com.turtlebone</groupId>

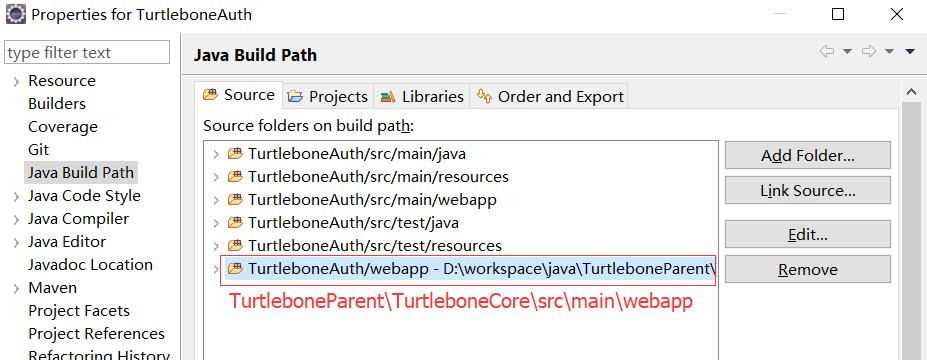
<version>0.0.1-SNAPSHOT</version>

<artifactId>TurtleboneCore</artifactId>

</dependency>

### Eclipse





## TurtleboneContract工程

### pom.xml

<dependency>

<groupId>com.turtlebone</groupId>

<artifactId>TurtleboneContract-api</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>

<dependency>

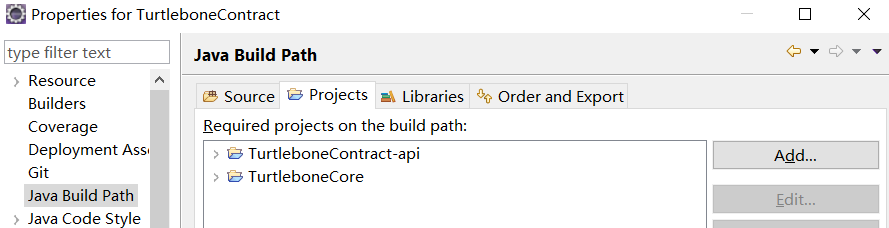
<groupId>com.turtlebone</groupId>

<version>0.0.1-SNAPSHOT</version>

<artifactId>TurtleboneCore</artifactId>

</dependency>

### Eclipse



## TurtleboneTask工程

### pom.xml

<dependency>

<groupId>com.turtlebone</groupId>

<artifactId>TurtleboneCore</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>

<dependency>

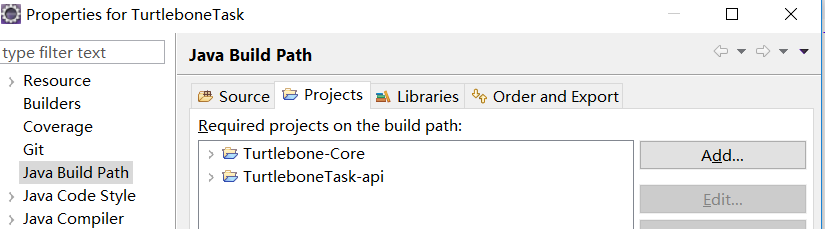
<groupId>com.turtlebone</groupId>

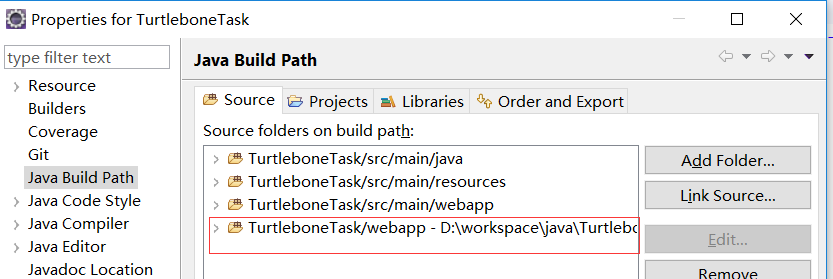
<artifactId>TurtleboneTask-api</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>

### Eclipse





## TurtleboneDream工程

### pom.xml

<dependency>

<groupId>com.turtlebone</groupId>

<artifactId>TurtleboneCore</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>

<dependency>

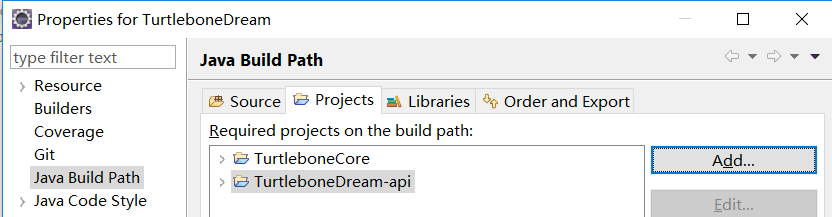
<groupId>com.turtlebone</groupId>

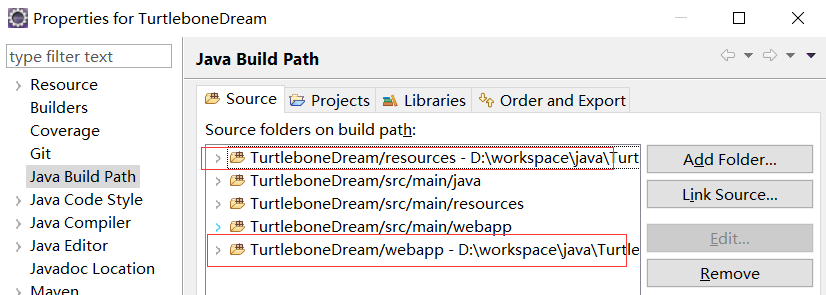
<artifactId>TurtleboneTask-api</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>

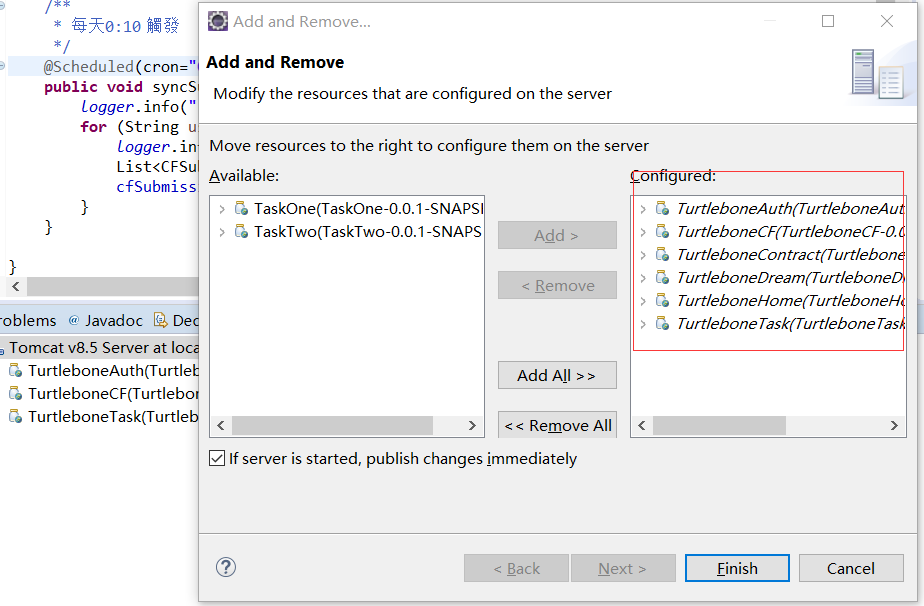
### Eclipse



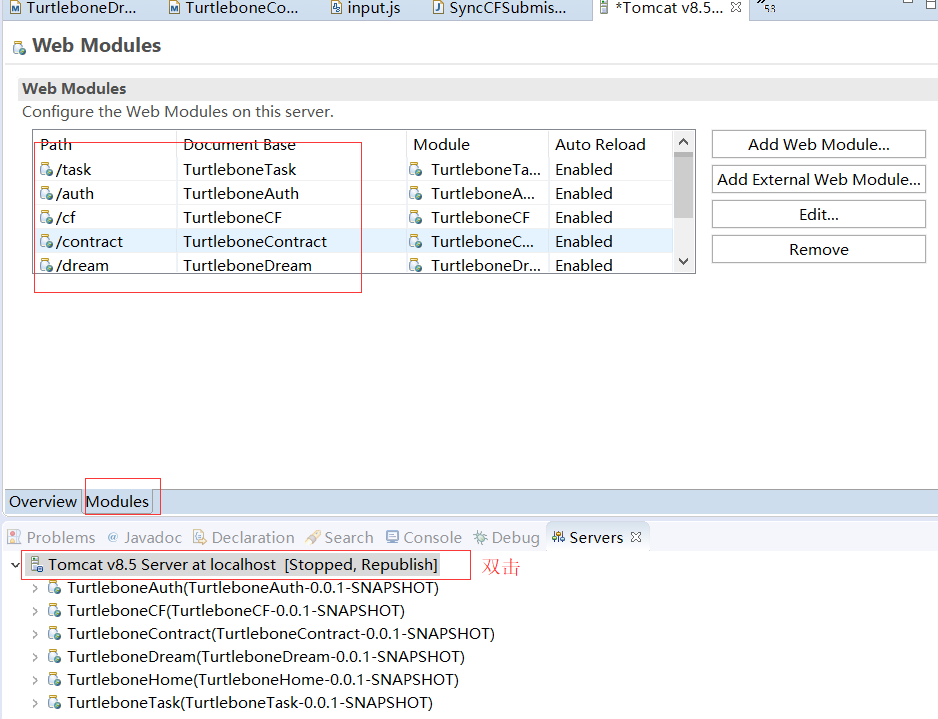


## 配置服務器

新建tomcat服务器，把工程添加到服务器中，如图：



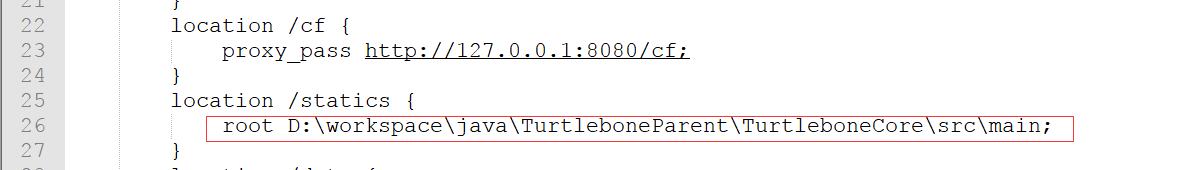
双击服务器，如下图：



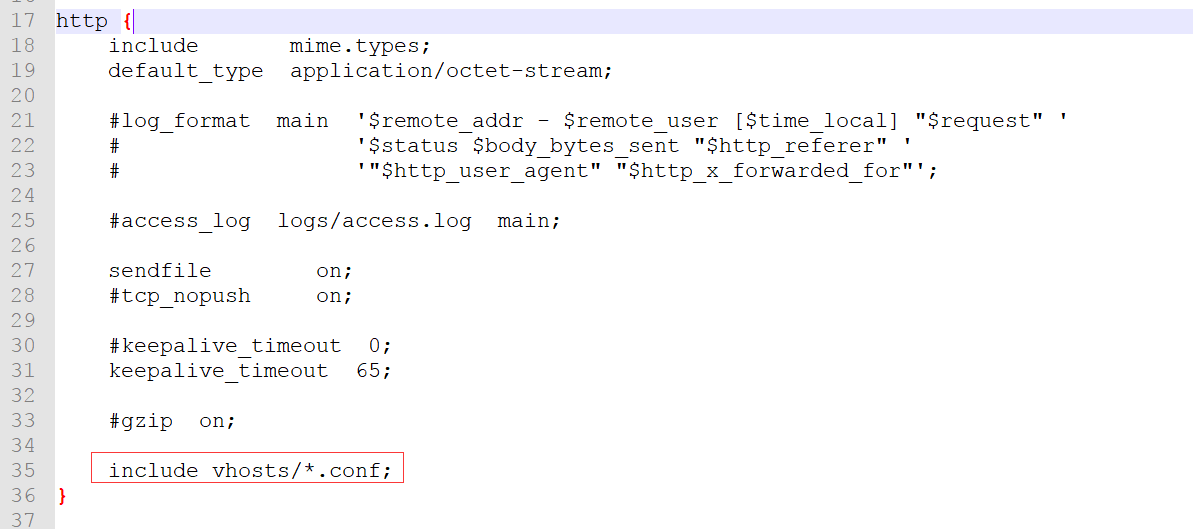
## 配置nginx

目的是把80端口转发到8080，这样所有请求都可以不填写端口（默认80端口）

复制local.conf文件到本地nginx的vhosts/里面，并修改如图：



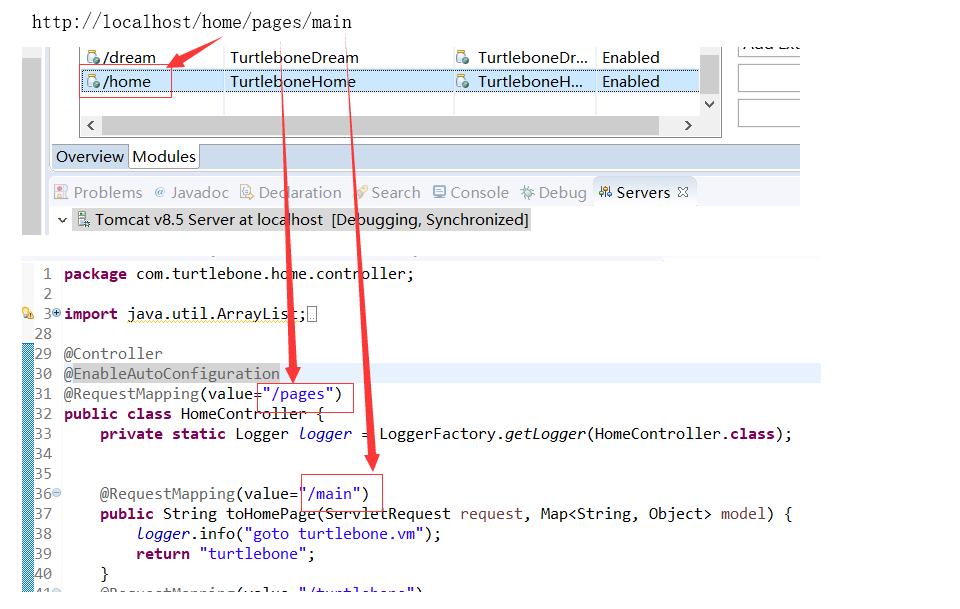
修改nginx的conf/nginx.conf，添加一句：include vhosts/\*.conf，如图：



## 如何访问

测试1，只配置home工程，那么可以通过访问<http://localhost/home/pages/main>

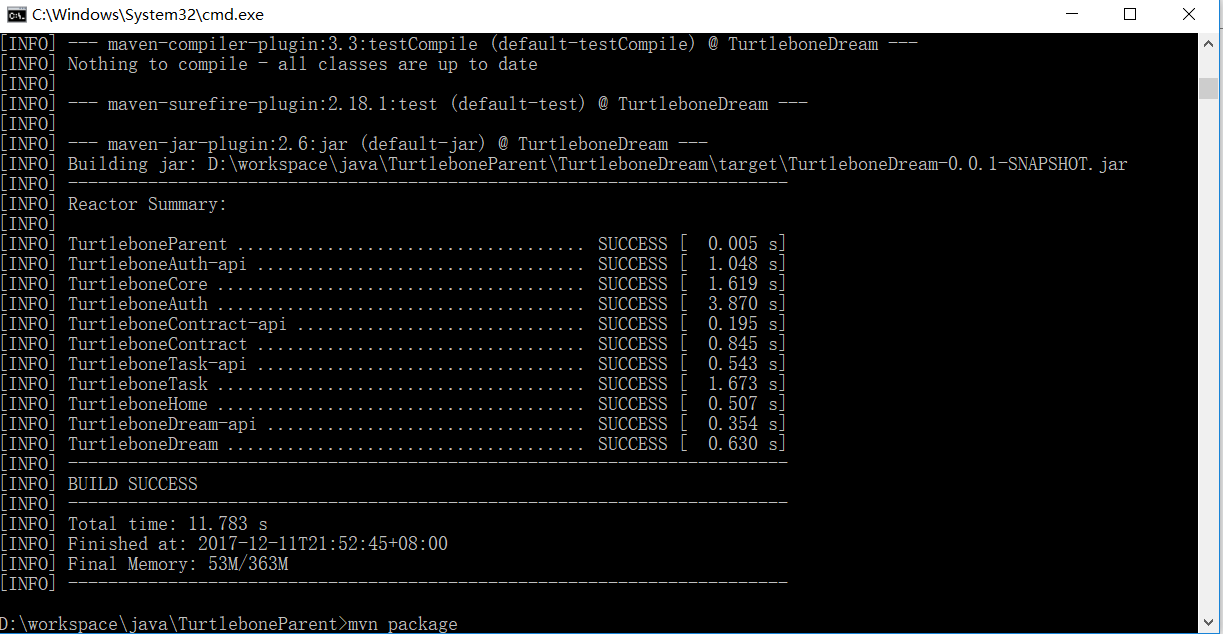
测试2，只配置cf工程，那么可以用postman或者浏览器访问<http://localhost/cf/cf/syncSubmission>



## 在linux上打包（windows中可以忽略）

进入TurtleboneParent目录，执行命令mvn package

成功后每一个TurtleboneXXX工程都会生成一个war，如图：



## 在linux中执行（windows中可以忽略）

把war包放到tomcat的webapps文件夾里面，并且文件命名如下：

TurtleboneAuth – auth.war

TurtleboneCF – cf.war

TurtleboneContract – contract.war

TurtleboneDream – dream.war

TurtleboneHome – home.war

TurtleboneTask – task.war