

一篇文章让你了解 chaosblade

<https://github.com/chaosblade-io/chaosblade-exec-jvm/tree/v0.1.0>

验证版本：0.1.0

以JVM Sandbox 方式进行验证

找到 jvm.spec.yaml 看查看支持的混沌注入参数

准备资源

```
# 准备示例项目 并自己手动启动
git clone https://github.com/niaoshuai/chaosblade-samples.git
## 启动项目的时候 可增加启动参数（用于远程Debug chaosblade-exec-jvm 源码）
-agentlib:jdwp=transport=dt_socket,address=127.0.0.1:5005,suspend=n,server=y

# 移动chaosblade-java-agent-0.1.0.jar 到 ~/.sandbox-module/
cp chaosblade-java-agent-0.1.0.jar ~/.sandbox-module/
# 启动并查看chaosblade 是否成功被挂载
bash ~/sandbox/bin/sandbox.sh -p 4314 -P 3658 -l
# 日志如下 FROZEN 默认是被冻结
chaosblade  FROZEN      LOADED      0          0          0.1.0 Changjun Xiao
# 找到目标应用的进程号
jps -v
# 使用沙箱attach p参数是进程号 P参数是沙箱对外通信端口 a参数是激活模块
bash ~/sandbox/bin/sandbox.sh -p 4314 -P 3658 -a chaosblade
```

实验（JVM）

1. OOM

```
# 注入堆区满载（请求时间比较短 > 3秒）
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=jvm&action=OutOfMemoryError&area=HEAP&threads=
1&enableSystemGc=false
# 注入非堆区满载
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=jvm&action=OutOfMemoryError&area=NOHEAP&thread
s=1&enableSystemGc=false
# 注入ByteBuffer(OFFHEAP)满载
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=jvm&action=OutOfMemoryError&area=OFFHEAP&threa
ds=1&enableSystemGc=false
```

2. JVM CPU 满载

```
# 注入JVM CPU满载
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=jvm&action=cpufullload
```

实验 (Servlet)

自定义异常

```
# 抛出自定义异常
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=servlet&action=throwCustomException&exception=
java.lang.Exception&servletpath=/index&pathinfo=/rest&method=get
```

延时

```
# 延时
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=servlet&action=delay&time=1000&servletpath=/in
dex&pathinfo=/rest&method=get
```

实验 (Http)

分析源码

<https://github.com/chaosblade-io/chaosblade-exec-jvm/wiki/%E5%BC%80%E5%8F%91%E8%80%85%E6%8C%87%E5%8D%97-%E5%8D%8F%E8%AE%AE%E7%AF%87>

```
# 状态
http://localhost:3658/sandbox/default/module/http/chaosblade/status?
suid=su378dsn137s53bs8adcn
# 创建
## http 之 rest 调用 (org.springframework.web.client.RestTemplate)
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=http&action=delay&time=3000&uri=http://www.bai
du.com&rest=true
## dubbo 调用
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
uid=su378dsn137s53bs8adcn&target=dubbo&action=delay&time=3000&service=com.examp1
e.HelloService&version=1.0.0&consumer=true
# 销毁
http://localhost:3658/sandbox/default/module/http/chaosblade/destroy?
suid=su378dsn137s53bs8adcn
```

1. HTTP REST

```
// 准备一个REST调用
public static void getBaidu(){
    RestTemplate restTemplate = new RestTemplate();
    restTemplate.getForObject("http://www.baidu.com", String.class);
}
```

```
# 开启一个Http Rest 混沌试验
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=http&action=delay&time=3000&uri=http://www.baidu.com&rest=true
# 验证结果 OK 使用完成之后记得销毁
```

2. HTTP CLIENT3

```
// 准备一个httpClient3调用
// [备注 当前版本需要 在没有参数的情况要加上最后面的 / 后面版本会修复这个问题 ]
public static void getBaidu() throws IOException {
    HttpClient client = new HttpClient();
    GetMethod get = new GetMethod("http://www.baidu.com/");
    // execute method and handle any error responses.
    client.executeMethod(get);
    get.getResponseBodyAsString();
    get.releaseConnection();
}
```

```
# 开启一个HTTP CLIENT3 混沌试验 [延时]
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=http&action=delay&time=3000&uri=http://www.baidu.com&httpClient3=true
# 验证结果 OK 使用完成之后记得销毁
```

```
# 开启一个HTTP CLIENT3 混沌试验 [自定义异常] (TODO 这个还没有更好的解决方案)
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=http&&action=throwCustomException&exception=java.lang.Exception&uri=http://www.baidu.com&method=get&httpClient3=true
# 验证结果 OK 使用完成之后记得销毁
```

3. HTTP CLIENT4

```
//准备一个httpClient4调用
public static void getBaidu() throws IOException {
    HttpClient client = HttpClient.createDefault();
    HttpGet get = new HttpGet();
    get.setURI(URI.create("http://www.baidu.com"));
    HttpResponse response = client.execute(get);
    StatusLine statusLine = response.getStatusLine();
    System.out.println(statusLine);
}
```

```
# 开启一个HTTP CLIENT4 混沌试验
http://localhost:3658/sandbox/default/module/http/chaosblade/create?
suid=su378dsn137s53bs8adcn&target=http&action=delay&time=3000&uri=http://www.baidu.com&httpClient4=true
# 验证结果 OK 使用完成之后记得销毁
```

