



链滴

# geoserver 发布栅格图叠加矢量图的遥感图

作者: [Hawkpool](#)

原文链接: <https://ld246.com/article/1578559025744>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在完成基础的矢量文件图层叠加图层组功能成功发布后,我们还需要使用shp文件与tif文件共同形成遥感地图进行发布

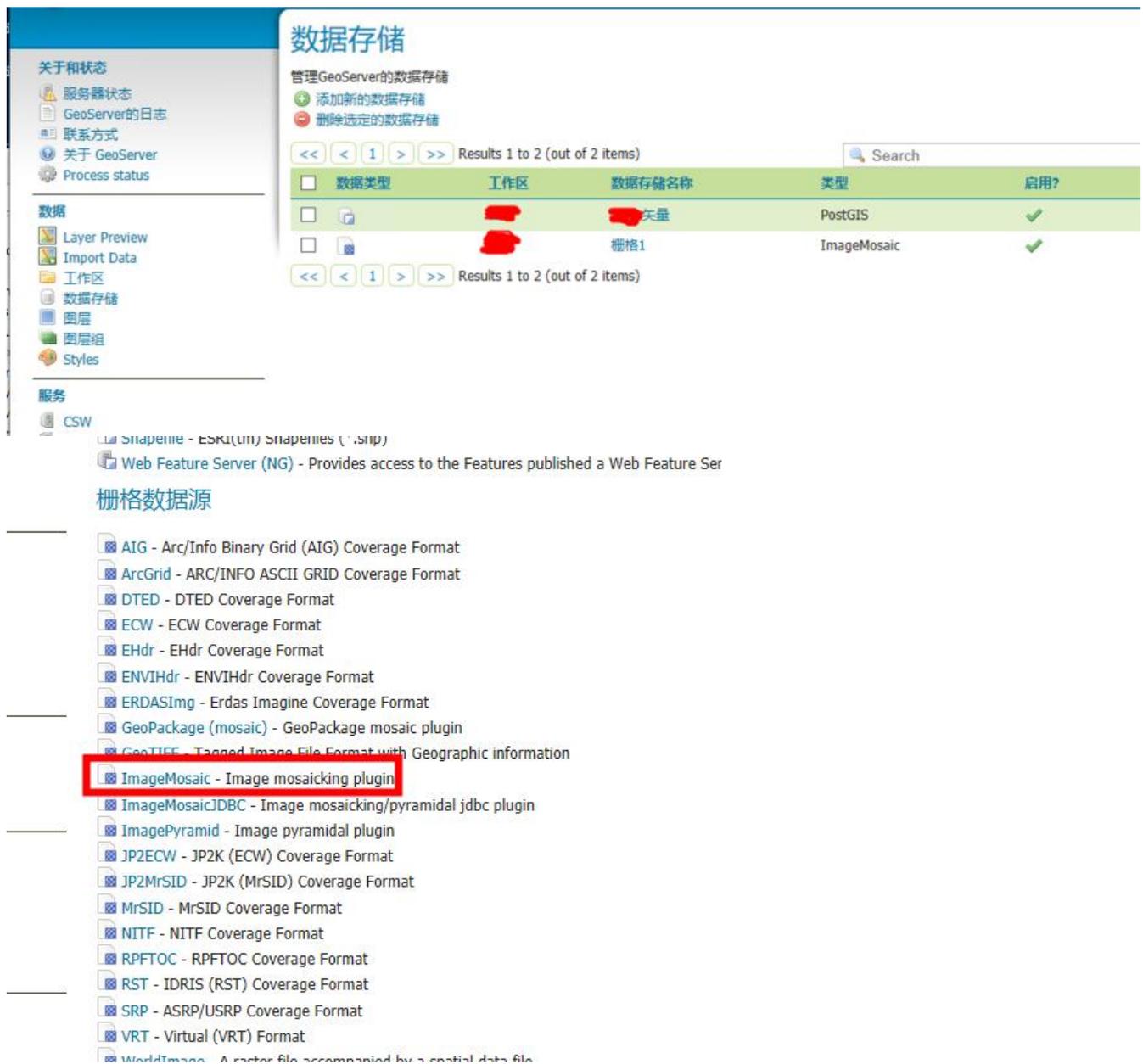
在参考了大量网络教程后,目前还是只实现了shp文件从数据库读取,而tif栅格文件需要从文件目录一个读取

遥感图的国家统一坐标系 是 CSCG2000/EPSCG:4546 ,所以,为了保证shp文件与tif文件能一起展示,必须保证shp文件和tif文件的坐标系是一致的 !!!

geoserver从Postgis中读取shp文件可以参考[上一篇文章](#)

geoserver发布tif文件教程如下

geoserver默认支持读取tif栅格文件,但是文件必须在geoserver服务器本地目录下



## 添加栅格数据源

说明

ImageMosaic  
Image mosaicking plugin

**存储库的基本信息**

工作区 \*

数据源名称 \*

栅格2

说明

栅格2

启用

**连接参数**

URL \*

file:///opt/geoserver/geoserver-config/3116.0-37513.0 [浏览...](#)

此处的路径是我在docker-compose文件中共享的文件目录,所以docker中的服务可以访问

## 新建图层

添加一个新图层

On stores you can also create a new coverage view by merging different coverages as a multibands coverage. [Configure new Coverage view ...](#)  
Here is a list of resources contained in the store '栅格2'. 点击你要配置的图层

发布的	图层名称	操作
	3116.0-37513.0	发布

至今还没有数据链接

**坐标参考系统**

本机SRS  
EPSG:4546

定义SRS  
EPSG:4546  EPSG:CGCS2000 / 3-degree Gauss-Kruger CM 111E...

SRS处理  
Reproject native to declared

**边框**

Native Bounding Box

最小 X	最小 Y	最大 X	最大 Y
341,226.58039832	2,003,156.846705	658,773.41960167	4,998,263.826050

从数据中计算

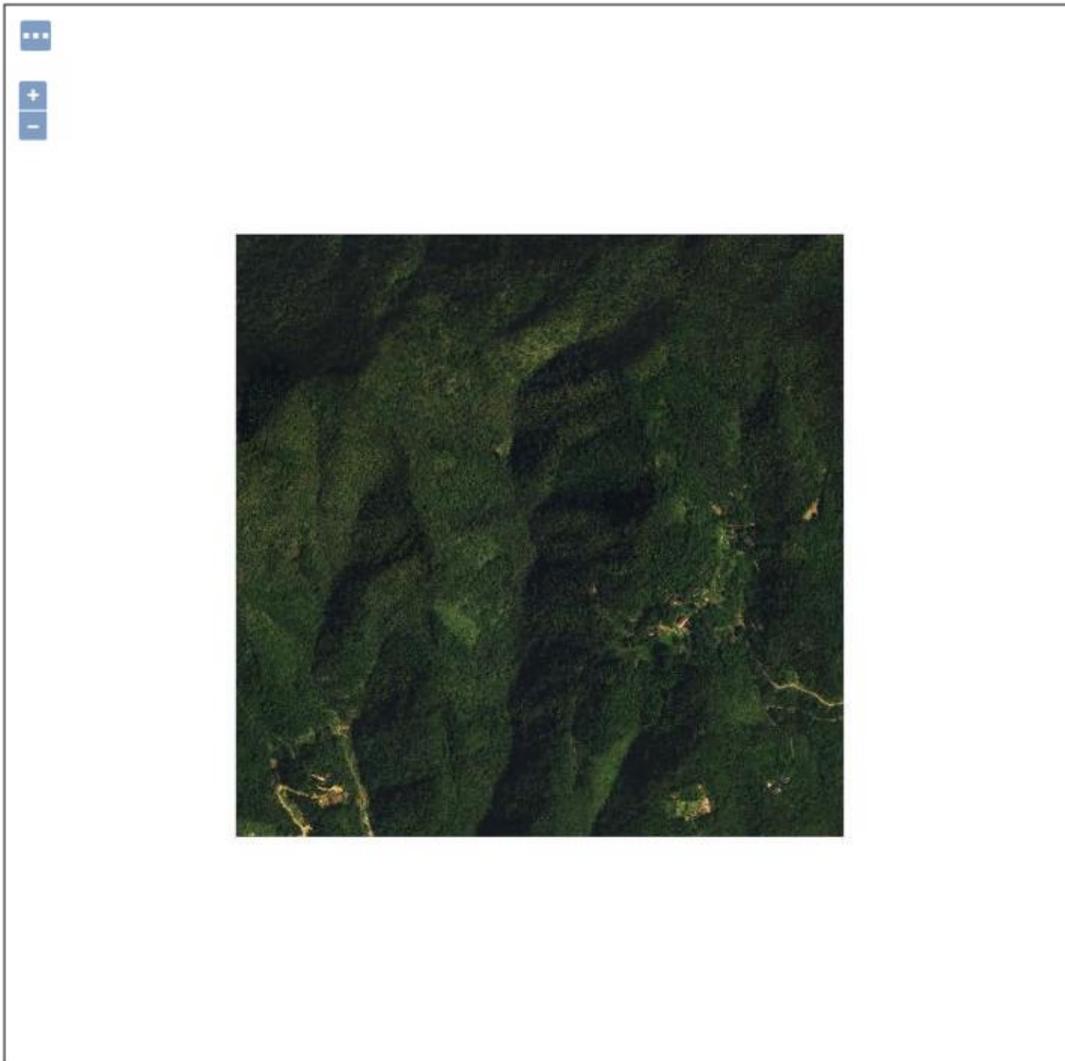
纬度/经度边框

最小 X	最小 Y	最大 X	最大 Y
108.98272135427	18.104164178231	113.017278645721	45.1198514785251

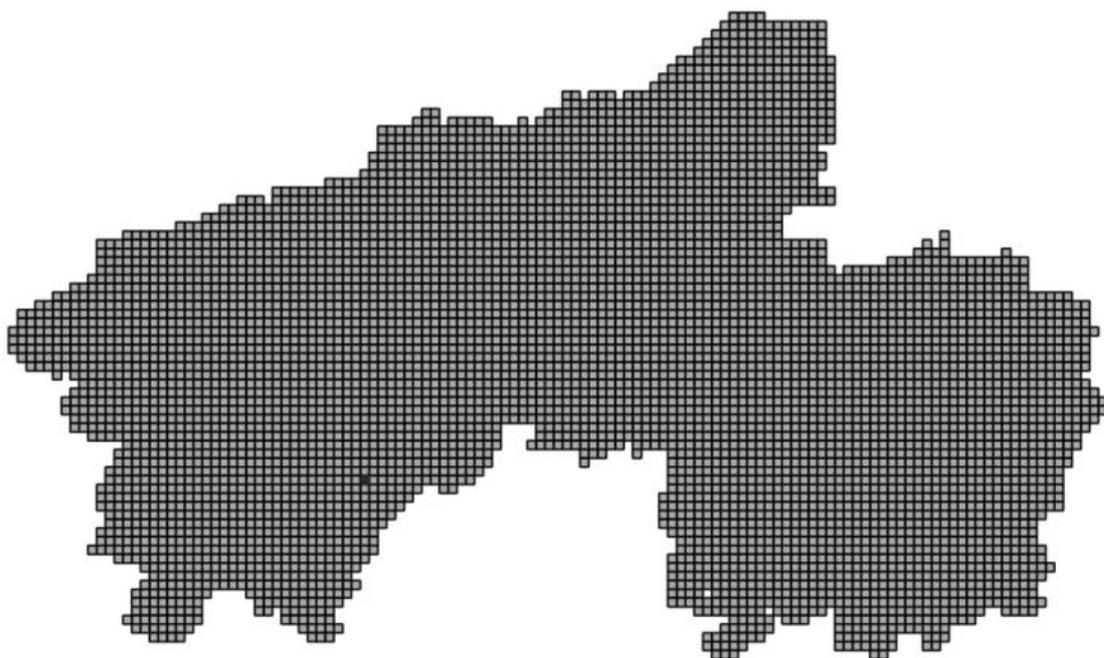
**覆盖参数**

Accurate Resolution Computation

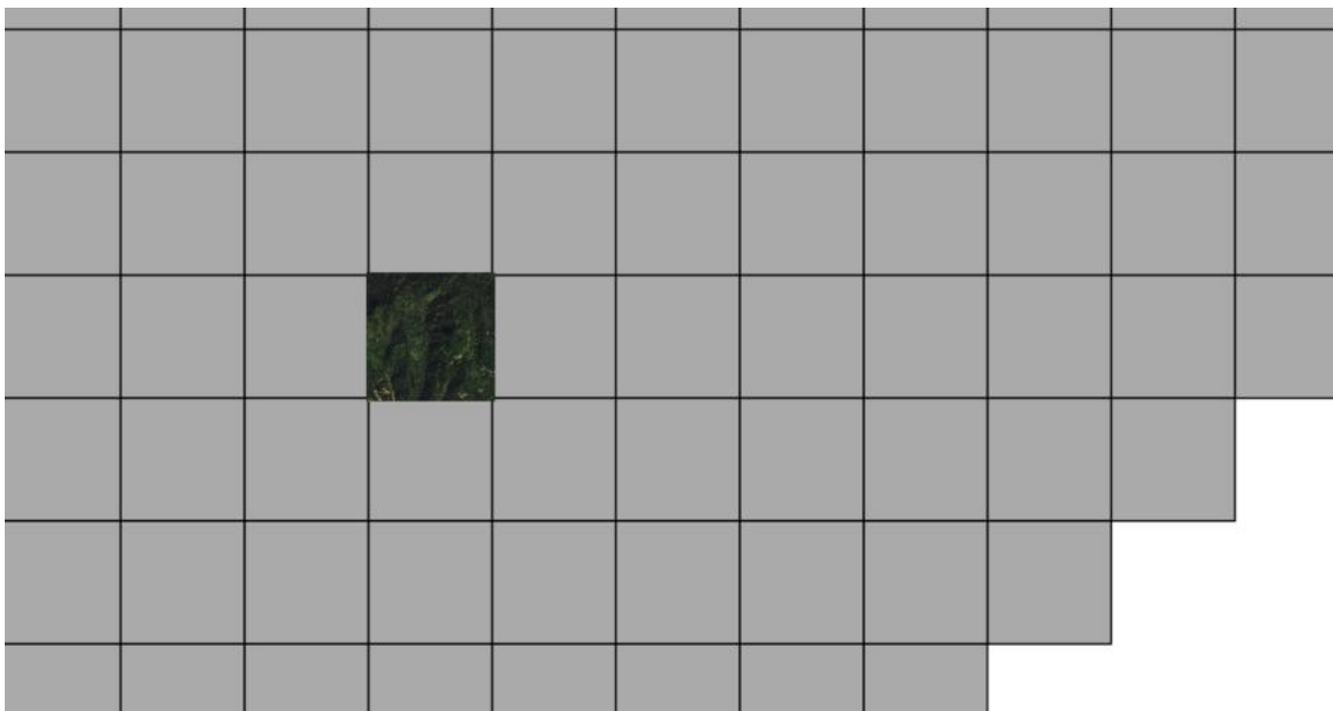
配置完成后即可在Layer Preview中查看到显示效果



再配置图层组与矢量网格图叠加



看到那个小黑点了吗,那就是我们的栅格图...



至此,基本的图层叠加已经完成,但是....看这密密麻麻的格子,总得有个批量导入栅格图的途径吧

原本是想统一数据源,让shp和tif数据都从数据库读取,降低服务器存储空间的压力

在参考文档[GeoServer发布PostGIS数据库中的栅格数据](#)后

经过大量尝试,虽然tif文件入库成功,geoserver的imageMosaicJdbc插件也成功安装,但是在使用imageMosaicJdbc读取数据库数据的时候却一直无法连接,并一直报错

连接参数

URL \*

file:/opt/geoserver/geoserver-config/test.pgraster.xml

```
Could not list layers for this store, an error occurred retrieving them: Failed to create reader from file:/opt/geoserver/geoserver-config/test.pgraster.xml and hints Hints: REPOSITORY = org.geoserver.catalog.CatalogRepository@72a194bf System defaults: FORCE_LONGITUDE_FIRST_AXIS_ORDER = true FEATURE_FACTORY = org.geotools.feature.LenientFeatureFactoryImpl@3a7eade2 FILTER_FACTORY = FilterFactoryImpl FORCE_AXIS_ORDER_HONORING = http COMPARISON_TOLERANCE = 1.0E-8 EXECUTOR_SERVICE = java.util.concurrent.ThreadPoolExecutor@42fab46b[Running, pool size = 0, active threads = 0, queued tasks = 0, completed tasks = 0] LENIENT_DATUM_SHIFT = true STYLE_FACTORY = StyleFactoryImpl GRID_COVERAGE_FACTORY = GridCoverageFactory TILE_ENCODING = null
```

保存 取消

经过问题排查和参数检查后,确认不是配置的问题,在百度和google之后终于在一个官方错误报告邮件录上找到类似的错误

[Image Mosaic failing to create reader again](#)

结果发现

Hi Paul,

rasters failing to load do not report any error to GeoServer, just a null. **This is a 10+ years old design mistake that's expensive to fix today** (idea, add a flag to make them throw an exception, retrofit each and every one of them to do so).

As Daniele suggested, currently the way is to switch to gt Dev logging and hunt into the many messages (some of which are red herrings).

The mosaic is especially hard because during init it creates tables, config files and the like, won't revert them on failure, and won't proceed further in the next setup if it finds them... So you have to know what to look for and delete manually.

I hope one day we can setup a code Sprint with a few devs, for a few days, to actually start returning meaningful info out if failures.

Cheers  
Andrea

大致意思是...有10多年历史的设计错误,改起来很麻烦..所以就没改了.....

总而言之,使用插件的方式基本是宣告失败

但是栅格图的发布总不能真的一个一个的通过UI界面去操作吧

经过大量的搜索以及资料翻阅,终于在csdn上找到了一个号称可以操作geoserver批量导入栅格图的Java项目代码

该项目下载下来后,能直接在idea中打开,但是却报了一个依赖错误

总的来说就是找不到这个依赖的jar

```
16 </dependency>
17 <!-- https://mvnrepository.com/artifact/it.geosolutions/geoserver-manager -->
18 <dependency>
19     groupId:it.geosolutions:groupId
20     artifactId:geoserver-manager:artifactId
21     version:1.8-SNAPSHOT:version
22 </dependency>
```

经过修改Maven版本,重新导包,重新下包等一系列操作后发现问题根深蒂固,依旧存在,于是我决定去Maven仓库里面一探究竟

果然,依赖的jar包没有下载下来,只有几个lastupdated文件

名称	修改日期	类型	大小
geoserver-manager-1.8-SNAPSHOT.jar.lastUpdated	2020/1/10 13:59	LASTUPDATED ...	1 KB
geoserver-manager-1.8-SNAPSHOT.pom.lastUpdated	2020/1/10 13:59	LASTUPDATED ...	1 KB
resolver-status.properties	2020/1/10 13:59	Properties 源文件	1 KB

这就好办了,去maven官网找爹去

Version	Repository	Usages	Date
1.7.x 1.7.0	Boundless	1	Sep, 2016
1.6.x 1.6.0	Boundless	2	Sep, 2016
1.5.x 1.5.2	Boundless	0	Sep, 2016
1.5.x 1.5.1	Boundless	2	Sep, 2016

结果一找,又懵逼了,人家maven只有1.7的版本,还是16年发布的,现在都2020年了啊!!!

我抱着死马当活马医的态度试着测试修改版本号,依旧不行

好吧,那我找geoserver-manager的仓库去!!

Name	Last modified	Size
../		
<a href="#">1.0/</a>	10-Feb-2019 13:49	-
<a href="#">1.0-SNAPSHOT/</a>	17-Apr-2019 04:10	-
<a href="#">1.1-SNAPSHOT/</a>	16-Apr-2019 23:13	-
<a href="#">1.2-SNAPSHOT/</a>	17-Apr-2019 04:24	-
<a href="#">1.2.1/</a>	16-Mar-2019 18:22	-
<a href="#">1.2.2/</a>	04-Feb-2019 22:47	-
<a href="#">1.2.2-SNAPSHOT/</a>	17-Apr-2019 08:24	-
<a href="#">1.2.3/</a>	15-Mar-2019 10:40	-
<a href="#">1.2.3-SNAPSHOT/</a>	16-Apr-2019 11:43	-
<a href="#">1.2.4-SNAPSHOT/</a>	17-Apr-2019 03:28	-
<a href="#">1.3-SNAPSHOT/</a>	17-Apr-2019 06:26	-
<a href="#">1.3.0/</a>	17-Mar-2019 17:04	-
<a href="#">1.3.1/</a>	18-Mar-2019 07:21	-
<a href="#">1.3.1-SNAPSHOT/</a>	16-Apr-2019 16:50	-
<a href="#">1.3.2/</a>	18-Feb-2019 08:37	-
<a href="#">1.3.2-SNAPSHOT/</a>	16-Apr-2019 21:44	-
<a href="#">1.3.3/</a>	17-Mar-2019 20:52	-
<a href="#">1.3.3-SNAPSHOT/</a>	17-Apr-2019 02:44	-
<a href="#">1.4-SNAPSHOT/</a>	16-Apr-2019 15:06	-
<a href="#">1.4.0/</a>	10-Feb-2019 10:31	-
<a href="#">1.4.1/</a>	10-Mar-2019 10:32	-
<a href="#">1.4.1-SNAPSHOT/</a>	17-Apr-2019 07:31	-
<a href="#">1.4.2/</a>	27-Jan-2019 08:49	-
<a href="#">1.4.3/</a>	19-Mar-2019 04:49	-
<a href="#">1.4.4/</a>	06-Feb-2019 18:15	-
<a href="#">1.5/</a>	18-Feb-2019 00:38	-
<a href="#">1.5-SNAPSHOT/</a>	16-Apr-2019 17:29	-
<a href="#">1.5.1/</a>	27-Jan-2019 12:42	-
<a href="#">1.5.2/</a>	14-Jan-2019 10:18	-
<a href="#">1.6-SNAPSHOT/</a>	16-Apr-2019 22:35	-
<a href="#">1.6.0/</a>	14-Jan-2019 14:32	-
<a href="#">1.7-SNAPSHOT/</a>	16-Apr-2019 13:13	-
<a href="#">1.7.0/</a>	15-Jan-2019 03:17	-
<a href="#">1.8-SNAPSHOT/</a>	17-Apr-2019 00:29	-
<a href="#">naven-metadata.xml</a>	03-May-2019 09:03	1.08 KB

乍一看好像来对地方了,但事实却是十分残酷的

## Index of main/it/geosolutions/geoserver-manager/1.8-SNAPSHOT

Name	Last modified	Size
../		
<a href="#">geoserver-manager-1.8-SNAPSHOT-sources.jar-&gt;</a>	-	-
<a href="#">geoserver-manager-1.8-SNAPSHOT.jar-&gt;</a>	-	-
<a href="#">geoserver-manager-1.8-SNAPSHOT.pom-&gt;</a>	-	-
<a href="#">naven-metadata.xml</a>	03-May-2019 09:03	322 bytes

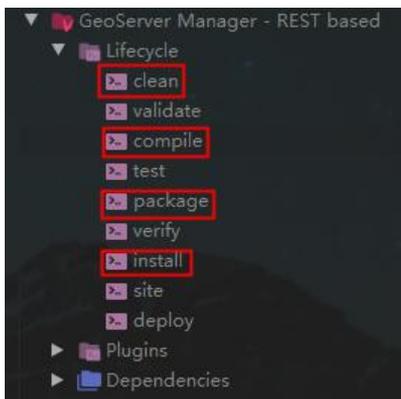
Artifactory/6.10.0 Server at localhost Port 8081



Maven仓库里的文件夹仍在,文件也能看到,但就是下载不了!!!

但我不信,一定还有其他解决方案,google一番后,发现github上好像有一个同名的开源项目

下载下来后,经过一键Maven构建打包四连后



果然在仓库里出现了想要的画面

名称	修改日期	类型	大小
_remote.repositories	2020/1/10 14:04	REPOSITORIES ...	1 KB
geoserver-manager-1.8-SNAPSHOT.jar	2020/1/10 14:04	Executable Jar File	209 KB
geoserver-manager-1.8-SNAPSHOT.jar.lastUpdated	2020/1/10 13:59	LASTUPDATED ...	1 KB
geoserver-manager-1.8-SNAPSHOT.pom	2020/1/10 9:19	POM 文件	13 KB
geoserver-manager-1.8-SNAPSHOT.pom.lastUpdated	2020/1/10 13:59	LASTUPDATED ...	1 KB
geoserver-manager-1.8-SNAPSHOT-sources.jar	2020/1/10 14:04	Executable Jar File	197 KB
maven-metadata-local.xml	2020/1/10 14:04	XML 源文件	1 KB
resolver-status.properties	2020/1/10 14:06	Properties 源文件	1 KB

回去一看批量导入项目的依赖问题也解决了

仔细一看这套批量导入的项目代码,其实就是调用了geoserver的rest接口

通过代码实现了UI界面上的一套操作

创建工作区---轮询文件目录,创建数据源---发布图层

通过修改项目中的相关参数,成功实现了轮询本地目录读取tif文件,发布栅格图的功能

<a href="#">3116.0-37512.0</a>	tuli1_1: <a href="#">3116.0-37512.0</a>	OpenLayers	KML	Select one
<a href="#">3116.0-37513.0</a>	tuli1_1: <a href="#">3116.0-37513.0</a>	OpenLayers	KML	Select one
<a href="#">3116.0-37514.0</a>	tuli1_1: <a href="#">3116.0-37514.0</a>	OpenLayers	KML	Select one

其中存在的问题:

- 该项目读取的目录与geoserver服务器需要是同一目录, 因此导致无法远程操作linux服务器下的geoserver批量发布
- 该项目没有一套成型的web操作界面,只能通过开发工具右键run java运行

目前来讲较好的解决方案:

- 使用带有GUI操作界面的Linux服务器,并安装IDEA开发环境,方便操作栅格批量发布,及文件传输
- 使用windows server服务器发布geoserver,并安装IDEA开发环境,方便操作栅格批量发布,及文件传输