**Geoserver学习笔记**

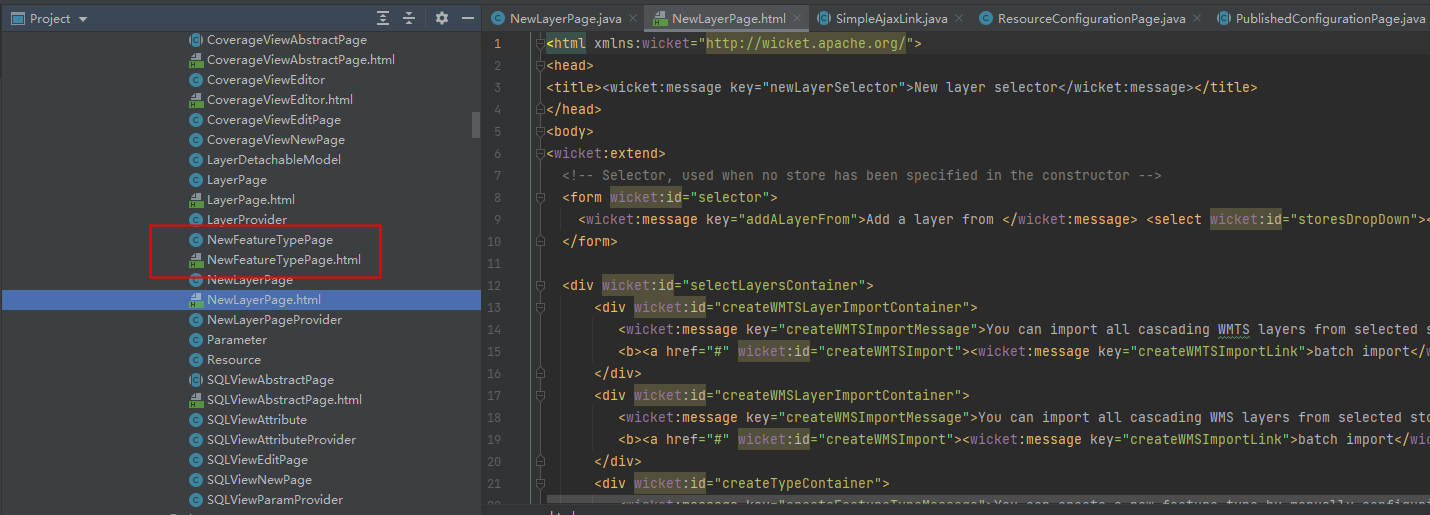
1. **Geoserver简介**

Geoserver开始于2001年，是为了政府民主化的产物，使政府更加透明。Geoserver项目开展过程中又建立了GeoTools工具包。PostGIS以及OpenLayers等的出现增强了Geoserver功能。Geoserver的愿景还是使地理数据让所有人可以触及。

Geoserver提供WMS、WFS以及WCS。WMS在服务器端渲染生成一张图片，返回给前端，可以处理矢量数据以及栅格数据。可以通过sld文件渲染矢量图形，且支持各种ESPG坐标系。WFS为返回图层矢量数据与属性，需要前端进行渲染；WCS为栅格数据的服务。以上三种服务为Geoserver默认自带的服务，Geoserver还提供一些扩展服务如：WPS、CWS等，这些服务默认不在Geoserver中需要另外安装。

**二、Geoserver页面**

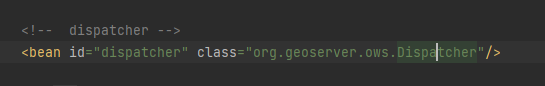
Geoserver页面通过wicket框架搭建的，一个html对应一个java类，在java类中包括具体的方法修该设置html中的数据。



Wicket操作类似于jquery，通过元素id获取元素wicket:id="selector"，然后对其进行操作。

Geoserver使用了spring-web进行开发，在main模块下的applicationContext.xml中定义了不同url对应的处理器。如下图对/geoserver/ows请求会首先被OWSHandlerMapping处理，OWSHandlerMapping继承了SimpleUrlHandlerMapping，而 SimpleUrlHandlerMapping又继承了AbstractUrlHandlerMapping，AbstractUrlHandlerMapping具有注册handler的方法registerHandler，将在applicationContext.xml中定义的url与指定的controller类放在map中保存。OWSHandlerMapping即从map中获取与当前url匹配的controller，如果找到则将请求发给该controller，这里的controller即为org.geoserver.ows.Dispatcher.在Dispatcher中会根据请求的参数服务名称在spring容器中查找Service类型的类。利用spring的applicationContext的getBeanNamesForType方法根据类class获取bean名称。然后在根据请求的服务的version将service进行确认，如果没有指定version，则返回版本最高的。然后通过反射机制，调用service对象的请求指定的方法。调用完serveice方法后通过回调进行返回消息。





**三、WMS服务**

WMS提供http请求获取地理图片服务。WMS服务的优点为，可以从多个服务器获取数据，然后再客户端将这些数据拼接起来。参考地址[WMS reference — GeoServer 2.21.x User Manual](https://docs.geoserver.org/stable/en/user/services/wms/reference.html)

WMS主要提供一下内容

|  |  |  |  |
| --- | --- | --- | --- |
| 操作 | 含义 |  |  |
| GetCapabilities | 关于服务信息，以及图层列表等 |  |  |
| GetMap | 返回指定区域的图像 |  |  |
| GetFeatureInfo | 通过像素坐标获取当前位置的图形数据 |  |  |
| DescribeLayer | 对图层描述 |  |  |
| GetLegendGraphic | 获取图例 |  |  |

WMS服务主要业务逻辑代码存在package org.geoserver.wms文件中的，

DefaultWebMapService其实现了WebMapServie定义的接口，GetMap、GetFeatureInfo、GetCapabilities等。

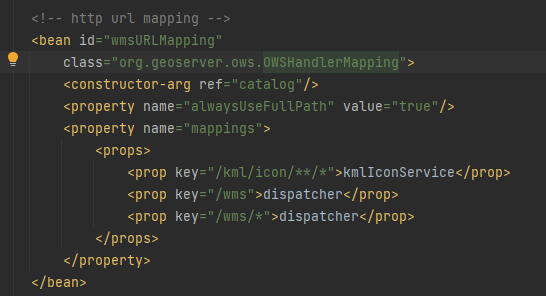
**3.1 GetCapabilities**

GetCapabilites方法获取WMS服务的元数据，包括支持的操作、参数以及可用的图层等。示例：

<http://localhost:8080/geoserver/wms?service=wms&version=1.1.1&request=GetCapabilities>

Geoserver服务请求主要包括三个参数，service指定请求服务类型，version指定服务的版本，同一个服务具有多个版本，request为该服务具体的操作。这里使用的是WMS服务因此指定该服务的类型service=wms，指定版本为1.1.1，操作为GetCapabilites。如果没有指定版本号，默认为最新版本。

WMS具体操作在项目wms模块中。在wms模块中的applicationContext.xml文件中，定义了wms的hanlder通过OWSHandlerMapping进行处理，对应的controller为dispatcher，而dispatcher在main模块中的applicationContext.xml中定义。



请求首先进入OWSHandlerMapping中，找到对应的controller，如果在handler中没有找到会在DispatcherServlet中寻找。找到controller即为dispatcher。在dispatcher中handleRequestInternal方法中处理请求。

1. 首先在init方法中解析请求参数信息；
2. 然后在service方法中根据请求中服务名称获取对应的service，查找service具体方法为findService。加载所有的service然后根据当前请求的服务名称以及版本号进行查找符合要求的service。加载所有service操作在loadServices方法中，主要用到spring的ApplicationContext对象getBeanNamesForType方法获取所有Service类型的bean名称。然后在利用bean名称从ApplicationContext中获取对应的bean。每次请求都会进行以上操作，该过程会通过ConcurrentHashMap进行缓存以提高效率。这里找到的service是org.geoserver.wms.DefaultWebMapService。
3. 获取请求操作对应的service中的方法，在dispatch方法中利用反射机制获取请求中request对应的方法这里即为GetCapabilites，获取到方法之后在组织方法所需的参数。Dispatch方法返回Operation对象，其中包含了service，method以及method所需的参数等信息。
4. 执行service，通过以上的准备，在execute中通过反射机制执行service中的具体方法。
5. 返回请求，执行response方法，将结果返回。

**3.2 GetMap**

GetMap通过http请求获取具有投影的图像：

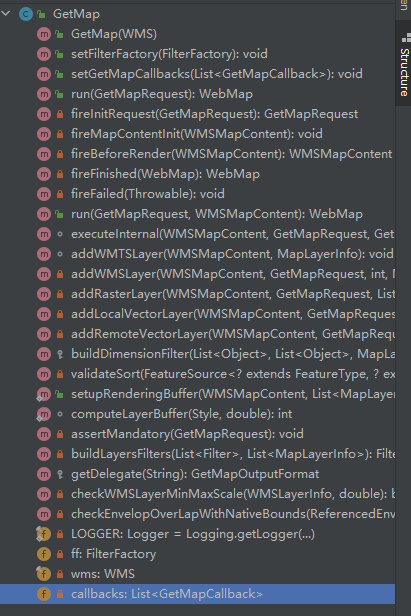
示例：

<http://localhost:8092/geoserver/webgis/wms?service=WMS&version=1.1.0&request=GetMap&layers=webgis%3Astates&bbox=-124.73142200000001%2C24.955967%2C-66.969849%2C49.371735&width=768&height=330&srs=EPSG%3A4326&styles=&format=image%2Fjpeg>

GetMap接口需要提供参数为，layers为图层名称 + workspace；bbox为图像的显示范围；width为输出图像的宽度；height为输出图像的高度，srs为坐标系；format为图像格式。概略步骤为，利用geotools创建mapContent，然后根据图层名称读取图层数据，将其添加到mapContent中，最后根据图像尺寸以及格式生成图像返回给前端。

Geoserver采用wicket架构进行开发，需要继承wicket下的WebApplication，然后在类中重写goHomePage,跳转到主页面。Wicket一个类与一个html相对应。在html中定义的wicket:message需要在项目的properties文件中找对应的文本，不同语言具有不同的properties文件，因此可以实现页面不同语言的切换。 wicket:id需要对应的java文件中定义的元素进行替换，并定义操作事件。

GetMap主要业务逻辑定义在package org.geoserver.wms中的GetMap类中。其主要方法如下：



org.geoserver.ows为open web service，其中定义了请求参数，请求解析，以及响应等类。其中Dispatcher继承了AbstractController，会对所有的请求进行拦截处理。GetMap请求会首先被Dispatcher拦截处理，处理请求参数将不同类型的功能处理为对应类型的请求。然后GetMap请求到达DefaultWebMapService类，DefaultWebMapService类判断请求类型，然后调用GetMap类中的run方法。

Geotools加载图层具体流程为首先创建DataStore，然后根据DataStore创建SimpleFeatureSoure，然后利用featureSource创建featureLayer，最后将layer添加到map中。在Geoserver中Dispatcher根据请求的具体类型选择KvpRequestReader进行具体图层读取，GetMap对应的KvpRequestReader为GetMapKvpRequestReader

3.2

**四、WFS服务**

WFS提供http请求获取、修改以及删除地理矢量数据服务。

WFS主要提供一下内容

|  |  |  |  |
| --- | --- | --- | --- |
| 操作 | 含义 |  |  |
| GetCapabilities | 关于服务信息，以及图层列表等 |  |  |
| DescribeFeatureType | 获取关于要素类型信息描述 |  |  |
| GetFeature | 获取要素，几何信息以及属性信息 |  |  |
| LockFeature | 通过给要素添加锁防止被修改 |  |  |
| Transaction | 编辑要素 |  |  |

示例：

<http://example.com/geoserver/wfs?service=wfs&version=1.1.0&request=GetCapabilities>

**五、WCS服务**

WCS提供http请求处理遥感数据，WMS只是获取图像数据，而WCS会在原始数据上进行一些处理，如图像裁剪、波段运算等。

WCS主要提供一下内容

|  |  |  |  |
| --- | --- | --- | --- |
| 操作 | 含义 |  |  |
| GetCapabilities | 关于服务信息，以及图层列表等 |  |  |
| DescribeCoverage | 返回xml格式关于数据详细描述 |  |  |
| GetCoverage | 返回栅格数据 |  |  |

示例：

<http://www.example.com/wcs?service=wcs&AcceptVersions=1.1.0&request=GetCapabilities>

以上三种服务为Geoserver默认自带的服务，Geoserver还具有WPS、CWS以及WMTS等需要安装的扩展服务。

WPS为地图计算服务，

<http://localhost:8080/geoserver/ows?service=WPS&version=1.0.0&request=GetCapabilities>

CSW为获取Geoserver中catlog中的数据。