智慧城市

文档版本记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 作者 | 日期 | 版本 | 纪要 |
|  | 李坡 | 2019年7月20日 | V1.0.0 | 文档内容形式 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[目录 3](#_Toc29905682)

[1 编写目的 4](#_Toc29905683)

[2 简要代码 4](#_Toc29905684)

# 编写目的

采用mapbox-vector-tile-3.0.0实现矢量切片mvt。

# 简要代码

|  |
| --- |
| HashMap<String, List<Geometry>> stringListHashMap = styleGroupSObject();  VectorTile.Tile.Builder tileBuilder = VectorTile.Tile.*newBuilder*();  if (stringListHashMap.size() > 0) {  for (Map.Entry<String, List<Geometry>> item : stringListHashMap.entrySet()) {  List<Geometry> geometryList = item.getValue();  VectorTile.Tile.Layer vectorLayer = createVectorLayer(geometryList, tileEnvelope, item.getKey());  if (vectorLayer == null) {  VectorTile.Tile.Layer tileLayer = createVectorLayer(new ArrayList<>(), tileEnvelope, "defaultLayer");  tileBuilder.addLayers(tileLayer);  } else {  tileBuilder.addLayers(vectorLayer);  }  }  } else {  VectorTile.Tile.Layer vectorLayer = createVectorLayer(new ArrayList<>(), tileEnvelope, "defaultLayer");  tileBuilder.addLayers(vectorLayer);  }   byte[] tileData = tileBuilder.build().toByteArray();  VectorTileCache.*getInstance*().addTile(key, CompressUtils.*gZip*(tileData));  stream.write(tileData); }  */\*\*  \* 创建图层  \*  \** ***@param*** *geometries  \** ***@param*** *tileEnvelope  \** ***@param*** *layerName  \** ***@return*** *\*/* private VectorTile.Tile.Layer createVectorLayer(List<Geometry> geometries, Envelope tileEnvelope, String layerName) {  IGeometryFilter acceptAllGeomFilter = geometry -> true;  MvtLayerParams layerParams = new MvtLayerParams(512, 4096);   try {   Envelope cilpEnvelope = new Envelope(tileEnvelope);  double cilpWidth = cilpEnvelope.getWidth();  double distance = cilpWidth / layerParams.tileSize;  cilpEnvelope.expandBy(distance \* 2);   TileGeomResult tileGeom = JtsAdapter.*createTileGeom*(geometries, tileEnvelope, cilpEnvelope, *GEOMETRY\_FACTORY*, layerParams,  acceptAllGeomFilter);   MvtLayerProps layerProps = new MvtLayerProps();  IUserDataConverter userDataConverter = new UserDataKeyValueMapConverter();   List<VectorTile.Tile.Feature> features = JtsAdapter.*toFeatures*(tileGeom.mvtGeoms, layerProps,  userDataConverter);   VectorTile.Tile.Layer.Builder layerBuilder = MvtLayerBuild.*newLayerBuilder*(layerName, layerParams);  layerBuilder.addAllFeatures(features);  MvtLayerBuild.*writeProps*(layerBuilder, layerProps);   return layerBuilder.build();  } catch (Exception e) {  return null;  } } |