## 依赖

<dependencies>

<!--xdocreport必备的依赖-->

<dependency>

<groupId>fr.opensagres.xdocreport</groupId>

<artifactId>fr.opensagres.xdocreport.document</artifactId>

<version>2.0.1</version>

</dependency>

<dependency>

<groupId>fr.opensagres.xdocreport</groupId>

<artifactId>fr.opensagres.xdocreport.core</artifactId>

<version>2.0.1</version>

</dependency>

<dependency>

<groupId>fr.opensagres.xdocreport</groupId>

<artifactId>fr.opensagres.xdocreport.template</artifactId>

<version>2.0.1</version>

</dependency>

<dependency>

<groupId>fr.opensagres.xdocreport</groupId>

<artifactId>fr.opensagres.xdocreport.converter</artifactId>

<version>2.0.1</version>

</dependency>

<!--模板引擎相关-->

<dependency>

<groupId>fr.opensagres.xdocreport</groupId>

<artifactId>fr.opensagres.xdocreport.template.freemarker</artifactId>

<version>2.0.1</version>

</dependency>

<!--文档格式相关-->

<dependency>

<groupId>fr.opensagres.xdocreport</groupId>

<artifactId>fr.opensagres.xdocreport.document.docx</artifactId>

<version>2.0.1</version>

</dependency>

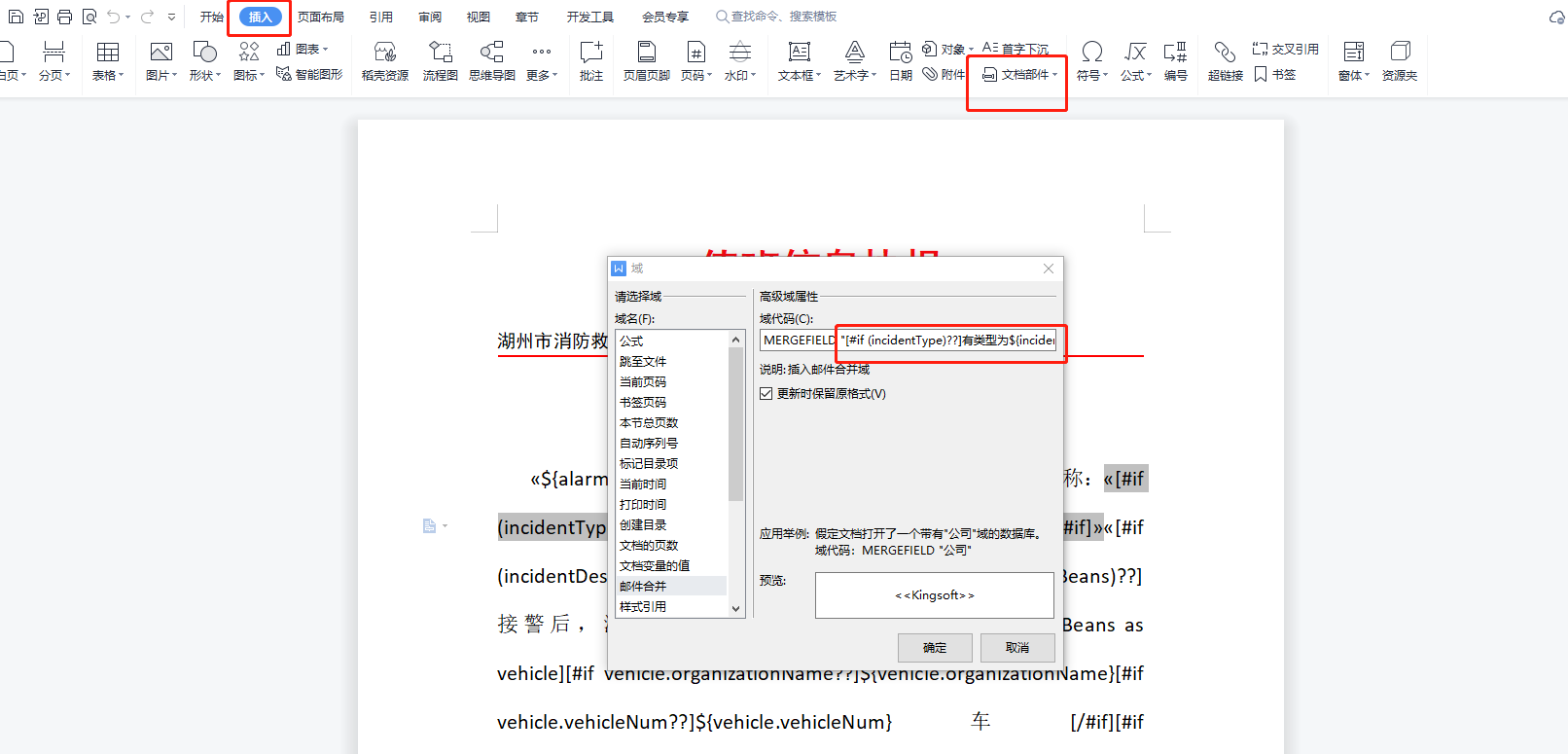
</dependencies>

## 模板

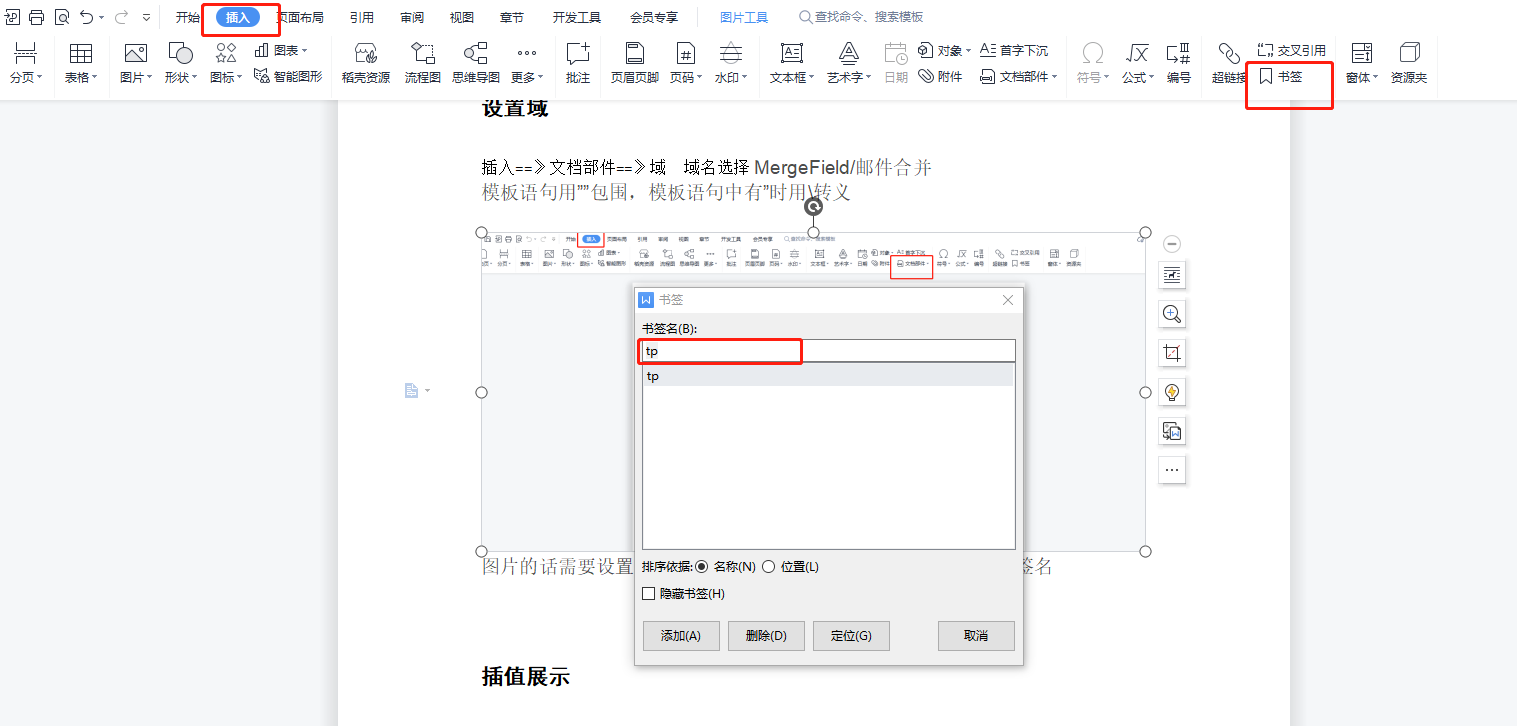
#### 设置域

插入==》文档部件==》域 域名选择MergeField/邮件合并

模板语句用””包围，模板语句中有”时用\转义



图片的话需要设置书签,选中图片==》插入==》书签==》设置书签名



#### 插值展示

${XXX}、普通变量

${object.XXX}、对象属性

${object[XXX]}、map

#### 列表展示

[#list list as item]

${item.a}

[/#list]

#### if判断

[#if xxx]

[#else][#elseif xxx]

[/#if]

非空判断 [#if (a)??&&a!=””][/#if]

## Java代码

**模板加载**

//1.通过freemarker模板引擎加载文档，并缓存到registry中

InputStream in = new FileInputStream(

"E:\\project\\xdocreport\\file\\test.docx");

IXDocReport report = XDocReportRegistry

.getRegistry()

.loadReport(in, TemplateEngineKind.Freemarker);

**设置域**

//2.设置填充字段、填充类以及是否为list。

//设置的域的名字要和模板里的名字保持一致

FieldsMetadata fieldsMetadata = report.createFieldsMetadata();

fieldsMetadata.load("project", Project.class);

fieldsMetadata.load("employees", Employee.class, true);//employees是个list

fieldsMetadata.addFieldAsImage("logo");//图片类型

report.setFieldsMetadata(fieldsMetadata);

**准备数据**

//3.模拟填充数据

String title = "项目开发报告";

Project project = new Project("网站开发", "截至2018年年底");

List<Employee> employees = new ArrayList<Employee>();

employees.add(new Employee("张三", "产品", "任务完成"));

employees.add(new Employee("李四", "开发", "任务完成"));

//图片数据的准备

IImageProvider logo = new FileImageProvider(

new File("E:\\project\\xdocreport\\file\\1.jpg"),

true);

logo.setSize(200f, 100f);

**填充数据**

//4.匹配填充字段和填充数据，进行填充

IContext context = report.createContext();

context.put("title", title);

context.put("project", project);

context.put("employees", employees);

context.put("logo", logo);

//准备输出流

OutputStream out = new FileOutputStream(

new File("E:\\project\\xdocreport\\file\\test-out.docx"));

//组装文档

report.process(context, out);