EI-PaaS的java web app部署

Date： 09 / 17/ 2017

|  |  |  |
| --- | --- | --- |
| Approved by | Checked by | Prepared by |
|  |  | LinShuotong |

目录

[一、获取EI-PaaS账号 3](#_Toc493519649)

[二、Java开发环境 3](#_Toc493519650)

[(1) 安装Java JDK 3](#_Toc493519651)

[(2) 安装Tomcat 4](#_Toc493519652)

[(3) 安装Java开发IDE 4](#_Toc493519653)

[三、编写自己的Java web service 5](#_Toc493519654)

[四、使用CLI工具进行部署 8](#_Toc493519655)

[(1) 使用CLI上传项目 8](#_Toc493519656)

[(2) Buildpacks 10](#_Toc493519657)

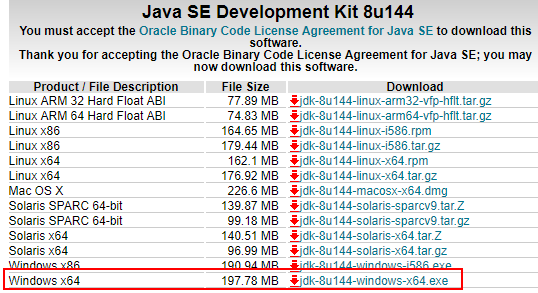
# 获取EI-PaaS账号

# Java开发环境

以下使用的是windows 10 64bit开发环境

## 安装Java JDK

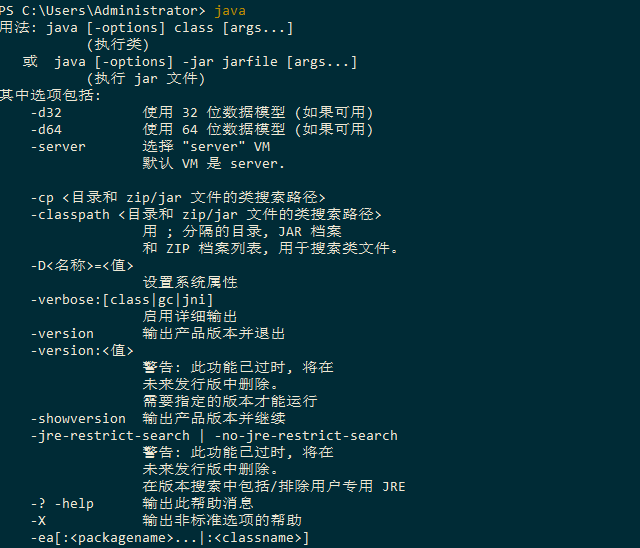
在Oracle官网上下载适配开发者操作系统的Java JDK，下载地址为<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>



下载完成后进行安装，之后配置JDK环境变量，需要设置JAVA\_HOME，CLASSPATH，PATH三个环境变量。

* JAVA\_HOME指明JDK的安装路径，如C:\Program Files\Java\jdk1.8.0\_60；
* PATH使得系统能够在任何路径下识别java命令，在系统原有的PATH后新增 %JAVA\_HOME%bin;%JAVA\_HOME%/jre/bin; 即可
* CLASSPATH为类加载路径，只有类在CLASSPATH路径中，java命令才能识别，设为 .;%JAVA\_HOME%/lib/dt.jar;%JAVA\_HOME%/lib/tools.jar

配置之后在cmd中运行java，显示结果如下即成功



## 安装Tomcat

在Tomcat官网上下载适配开发者操作系统的Tomcat版本，下载地址为<http://tomcat.apache.org/download-80.cgi>

下载后解压如下图



## 安装Java开发IDE

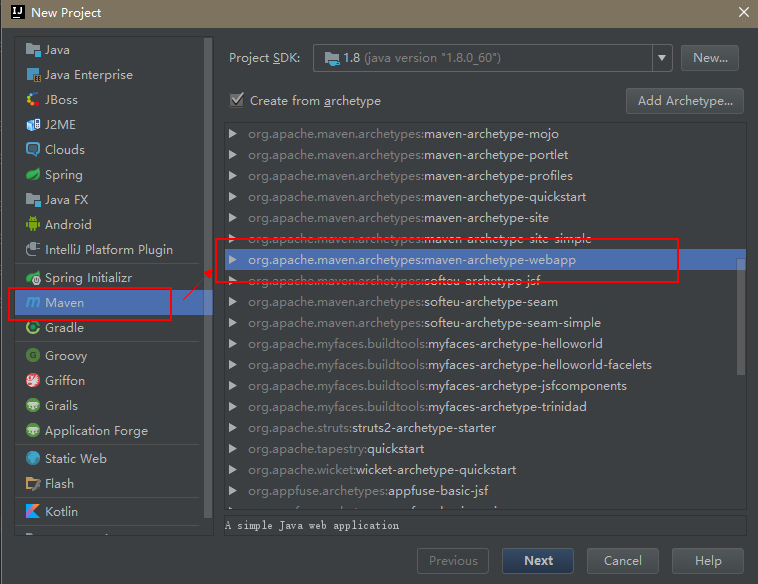
下载JetBrains的IntelliJ IDEA，IntelliJ IDEA是java语言开发的集成环境，IntelliJ IDEA的下载地址为<https://www.jetbrains.com/idea/download>

# 编写自己的Java web service

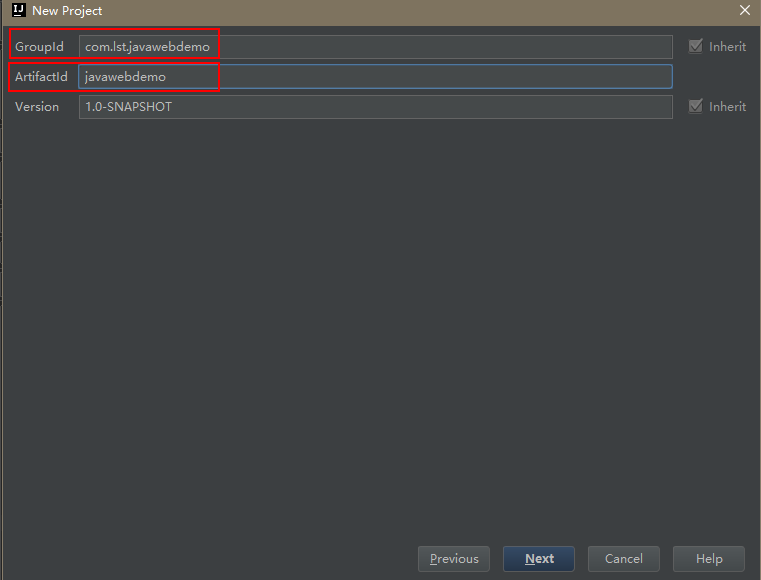
下边的示例是在windows 10 64bit系统下使用JetBrains的IntelliJ IDEA进行开发，

创建一个新的project，选择maven工具来创建webapp项目。maven是一个项目构建工具，多用于java项目。具体见maven官网<http://maven.apache.org/>

选择如下选项后点击Next



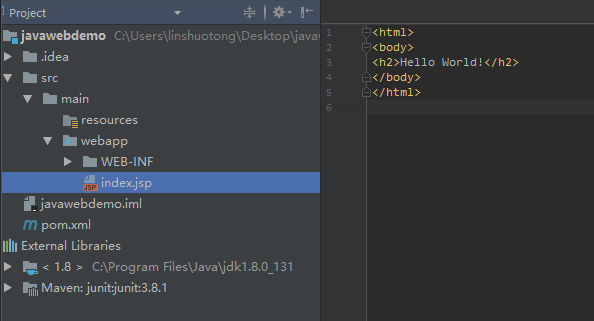
填写GroupID和ArtifactID，GroupID指公司名或组织名，一般由三部分组成，每部分使用”.”隔开，第一部分为项目类型(如商业为com或者非营利性组织为org)，第二部是公司名，第三部分为项目名；ArtifactID为项目名。之后点击Next，填写project location，最后点击finish。



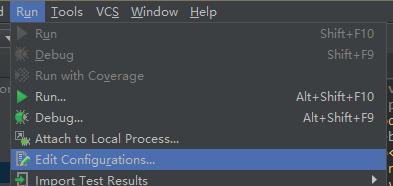
创建项目后，右下角会弹出maven项目的提示框，点击Enable Auto-Import，之后maven的配置文件pom.xml发生改变时，maven会自动下载所需的依赖包。



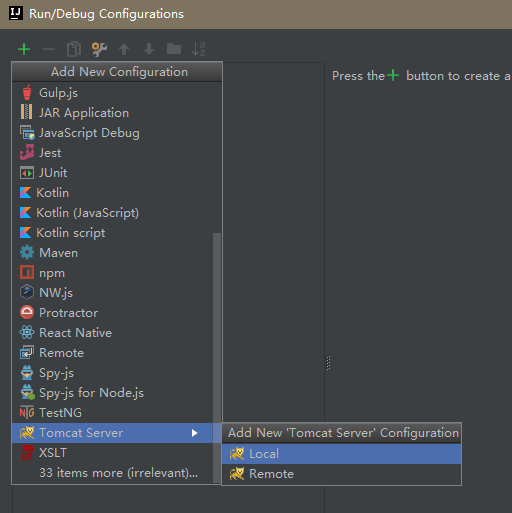
等待maven下载完所需的包后，此时目录结构如下图所示，默认下有index.jsp文件



接着进行Tomcat的配置，点击上方导航栏的Run-> Edit Configurations

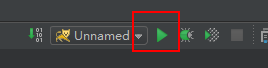


之后点击左上”+”, Tomcat Server->Local

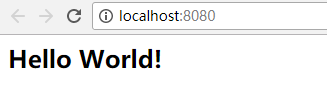


之后点击右下角Fix，选择javawebdemo:war exploded后，点击Apply, OK。此时环境已经全部搭建好了。

接着运行demo，点击右上角运行项目。



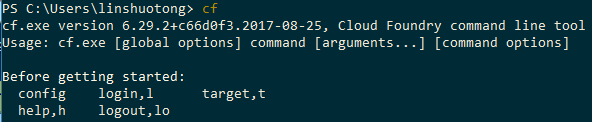
运行后浏览器会弹出localhost:8080如图所示，此时运行成功



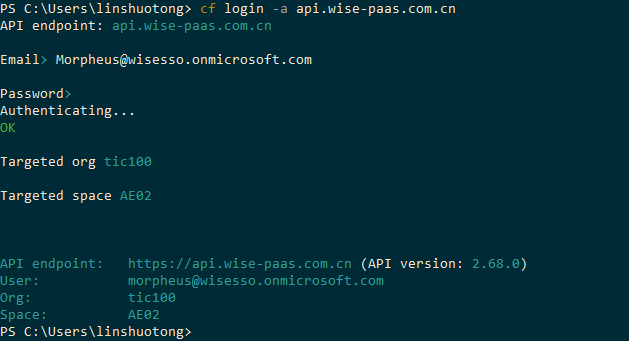
# 使用CLI工具进行部署

## 使用CLI上传项目

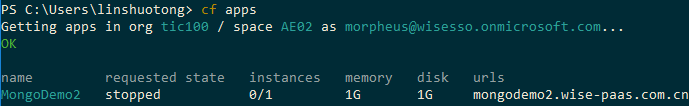
在Cloud Foundry 官网文档中https://docs.cloudfoundry.org/cf-cli/install-go-cli.html，找到对应操作系统的CLI安装方式。此处以winodows 10 64bit系统上安装CLI为例。下载安装包安装完之后，打开cmd中运行cf命令，出现提示则说明安装成功。



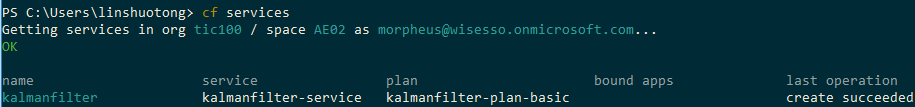
接着使用cf login –a api.wise-paas.com.cn 命令后输入账号密码后登录。



使用cf apps可以列出当前所有的应用

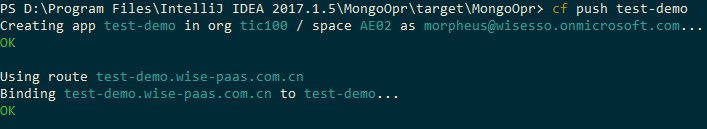


使用cf services可以列出当前所有的服务



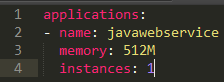
cmd进入要上传的web项目的根目录中，上传项目有两种方式：

1. 使用cf push APPNAME命令上传，此时cf会将该目录下的所有文件打包后上传到EI-PaaS上，新上传的应用名即刚才指定的APPNAME。



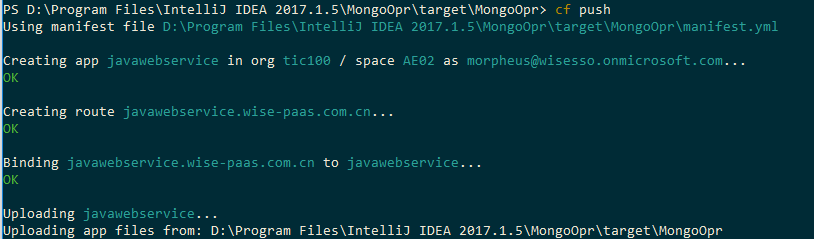
1. 使用manifest.yml文件进行参数配置，可以在manifest.yml中配置上传应用的相关参数，如app命名，占用内存及创建的实例数目等，cf上传项目时能够读取该配置文件进行相关配置。

在上传项目根目录中创建manifest.yml文件，编辑如下内容：

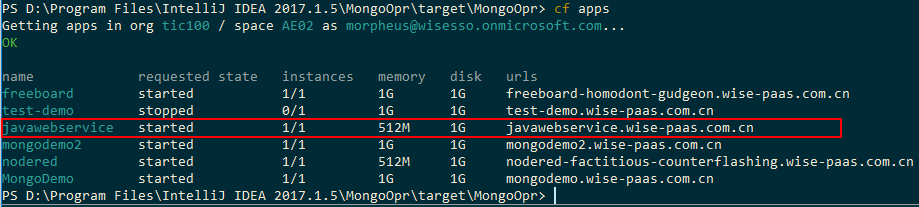


name表示上传后该app的命名，memory表示该app占用的内存空间大小，instances表示创建该app的实例数。

之后在根目录中cmd运行cf push，cf push命令默认情况下会在当前目录查找命名为manifest.yml的文件，并读取其中的参数将项目进行上传并配置。



使用cf apps可以看到我们新上传的项目



打开浏览器输出javawebservice.wise-paas.com.cn，可以看到项目的运行



## Buildpacks

由于我们上传的项目可能是java也可能是其他语言python，Cloud Foundry需要区别不同的语言并配置相关的运行环境才能运行上传的项目，此时会用到Cloud Foundry上的Buildpacks。Buildpacks为应用程序提供框架和运行时支持。Buildpacks会检查上传的应用程序，以确定哪些依赖关系并下载相关依赖包，这样我们的项目就能运行起来了。