1. 新建个类

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.Properties;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.env.EnvironmentPostProcessor;

**import** org.springframework.core.env.ConfigurableEnvironment;

**import** org.springframework.core.env.MutablePropertySources;

**import** org.springframework.core.env.PropertiesPropertySource;

**import** org.springframework.core.io.FileSystemResource;

**import** org.springframework.core.io.support.PropertiesLoaderUtils;

**public** **class** MyEnvironmentPostProcessor **implements** EnvironmentPostProcessor {

@Override

**public** **void** postProcessEnvironment(ConfigurableEnvironment configurableEnvironment, SpringApplication application) {

//tomcat路径

String property = System.*getProperty*("catalina.home");

//System.out.println("catalinahome:"+property);

String path =property+File.***separator***+"conf"+File.***separator***+"my.properties";

File file = **new** File(path);

//System.out.println("Loading local settings from : "+path);

**if** (file.exists()) {

MutablePropertySources propertySources = configurableEnvironment.getPropertySources();

Properties properties = loadProperties(file);

//System.out.println(properties.toString());

propertySources.addFirst(**new** PropertiesPropertySource("Config", properties));

}

}

**private** Properties loadProperties(File f) {

FileSystemResource resource = **new** FileSystemResource(f);

**try** {

**return** PropertiesLoaderUtils.*loadProperties*(resource);

} **catch** (IOException ex) {

**throw** **new** IllegalStateException("Failed to load local settings from " + f.getAbsolutePath(), ex);

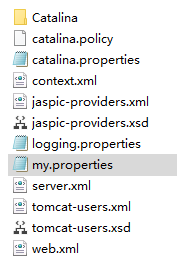
}

}

}

addFirst表示当外部配置文件与内部配置文件的内容一致时，以外部为准；addLast则相反

1. 根据新建的类，我这里的配置文件名称为：my.properties，将my.properties放置在所要部署的tomcat下的conf下



1. 在src/main/resources下新建META-INF，然后再在META-INF新建spring.factories，内容为：

org.springframework.boot.env.EnvironmentPostProcessor=com.cpic.handler.MyEnvironmentPostProcessor