

IDEA搭建LUA + OPENRESTY + NGINX开发环境

idea搭建Lua+OpenResty+Nginx开发环境

1. 下载openresty并解压到本地:

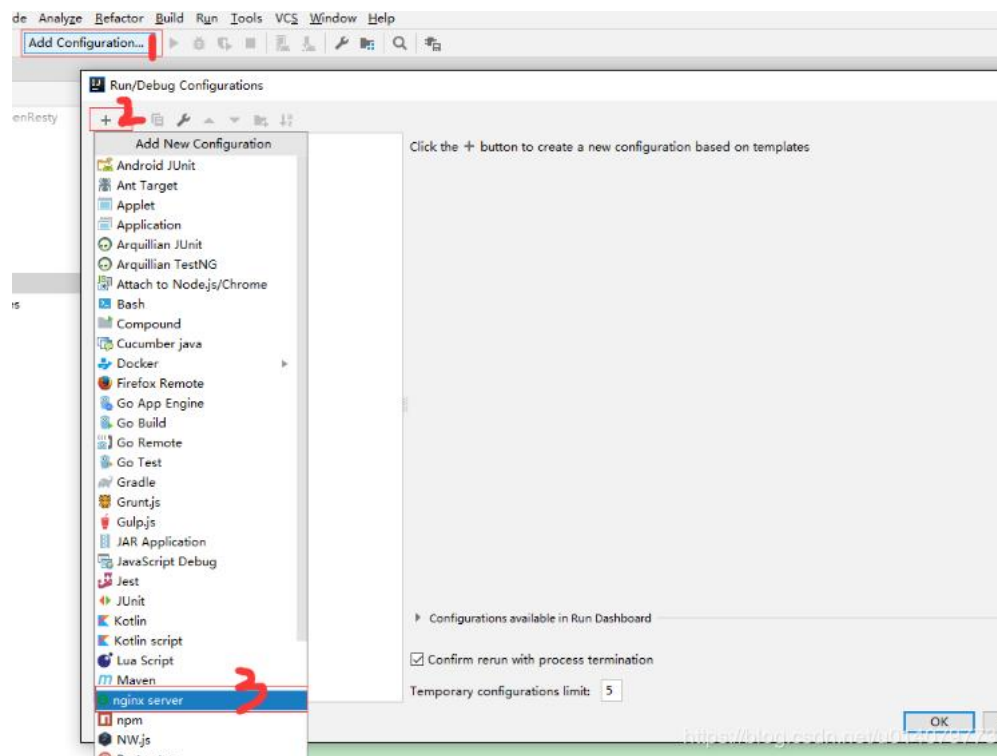
下载地址: <http://openresty.org/cn/download.html>

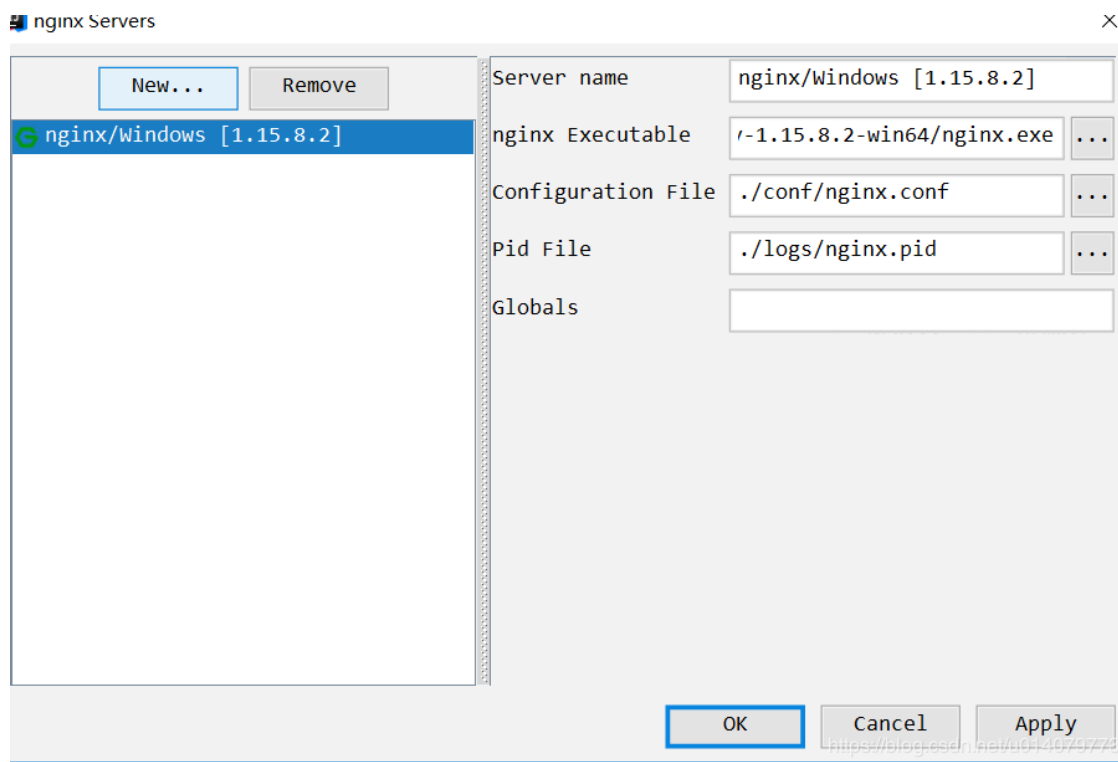
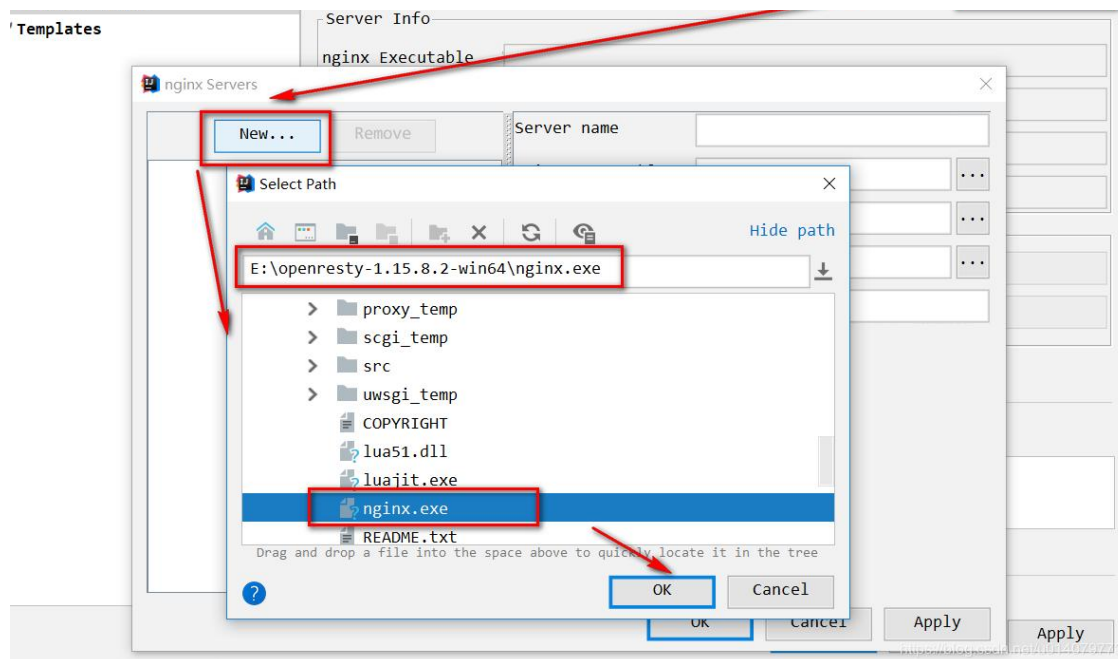
> 此电脑 > SOMNUS (E:) > openresty-1.15.8.2-win64		
名称		修改日期
client_body_temp		2019/9/21 1:
conf		2019/9/23 18
fastcgi_temp		2019/9/21 1:

2. 安装插件: Lua, openrestyLuaSupport, idea-nginx-support

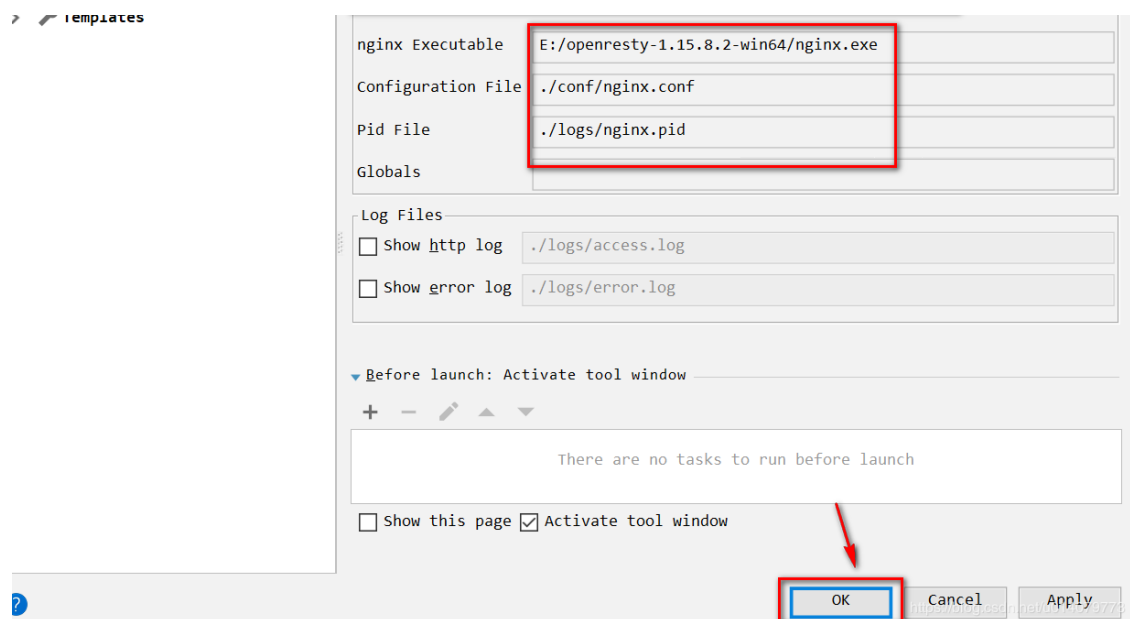
参考地址: <https://blog.csdn.net/u014079773/article/details/101062867>

3. 配置nginx服务:

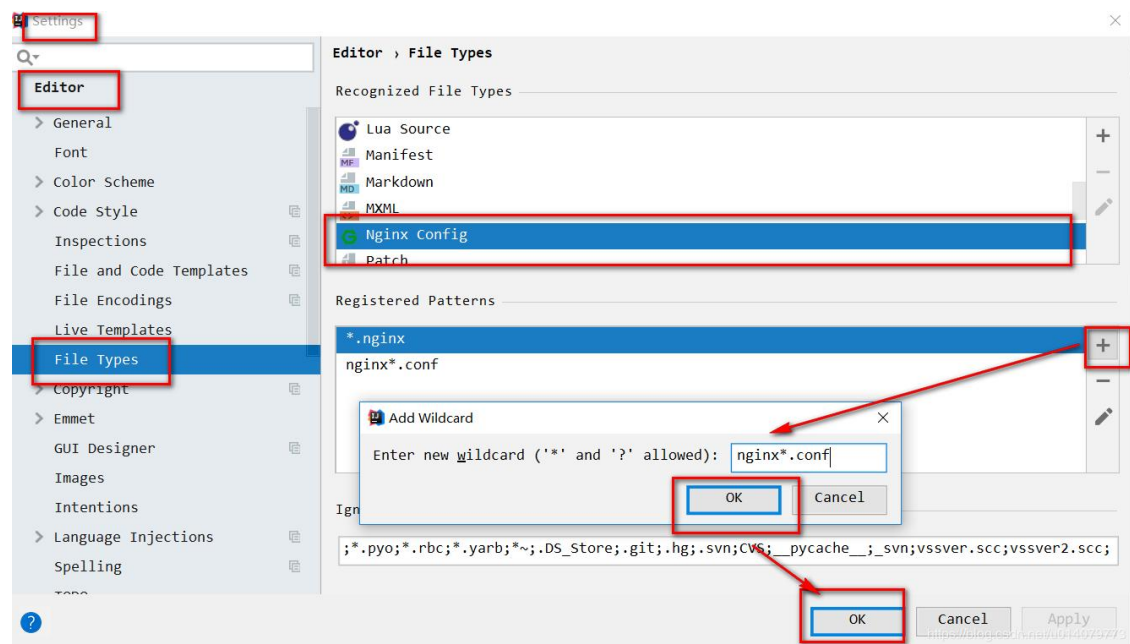




点击"OK", 选择已添加的nginx

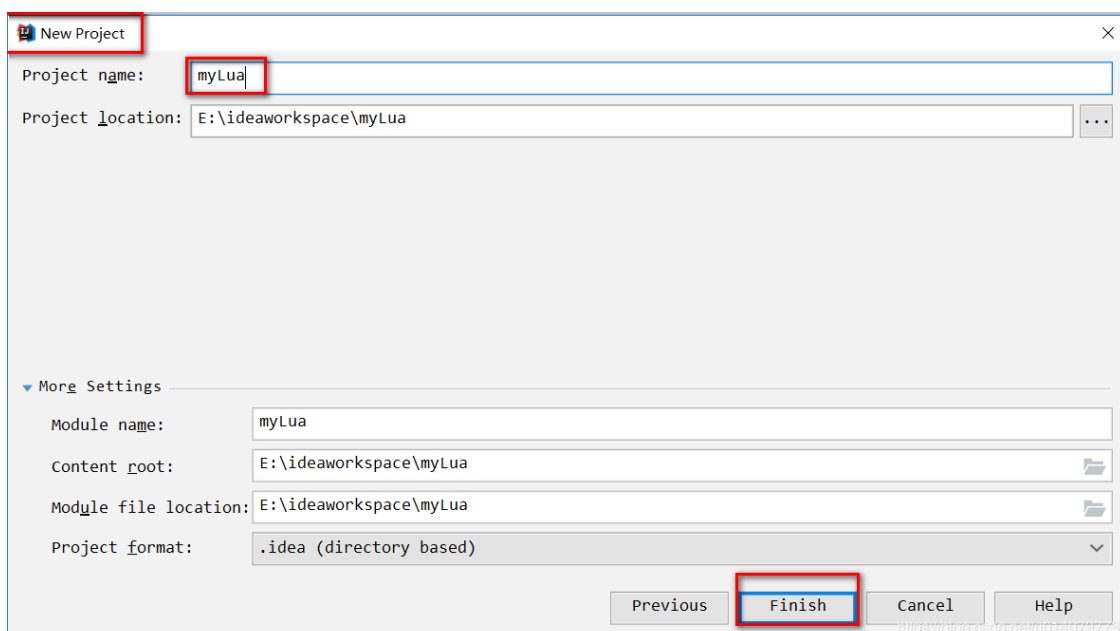
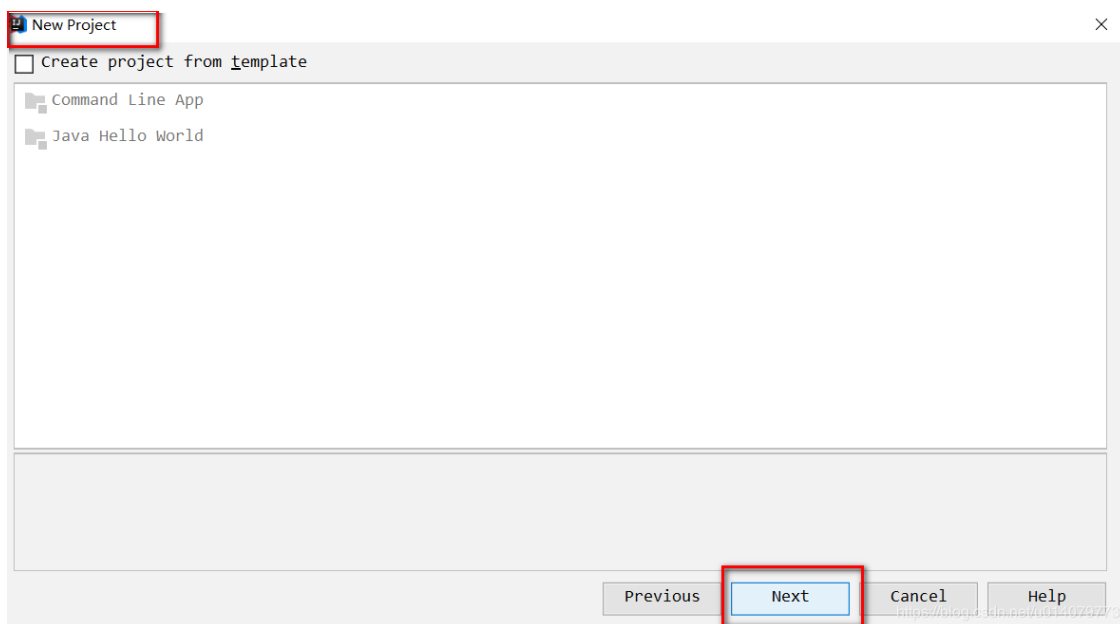
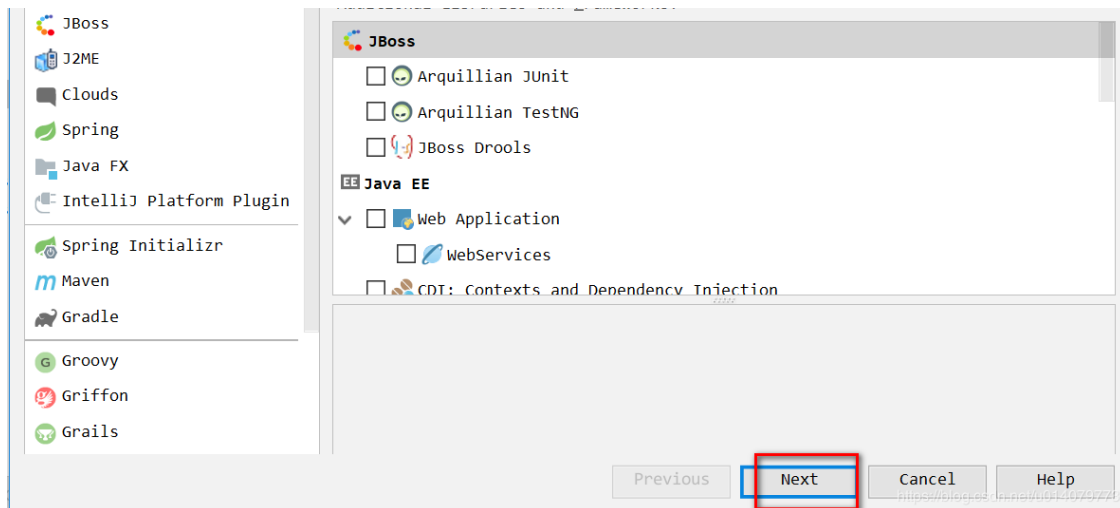


配置nginx.conf文件打开方式，设置完成之后，我们看到的nginx.conf文件中语法就高亮显示了。

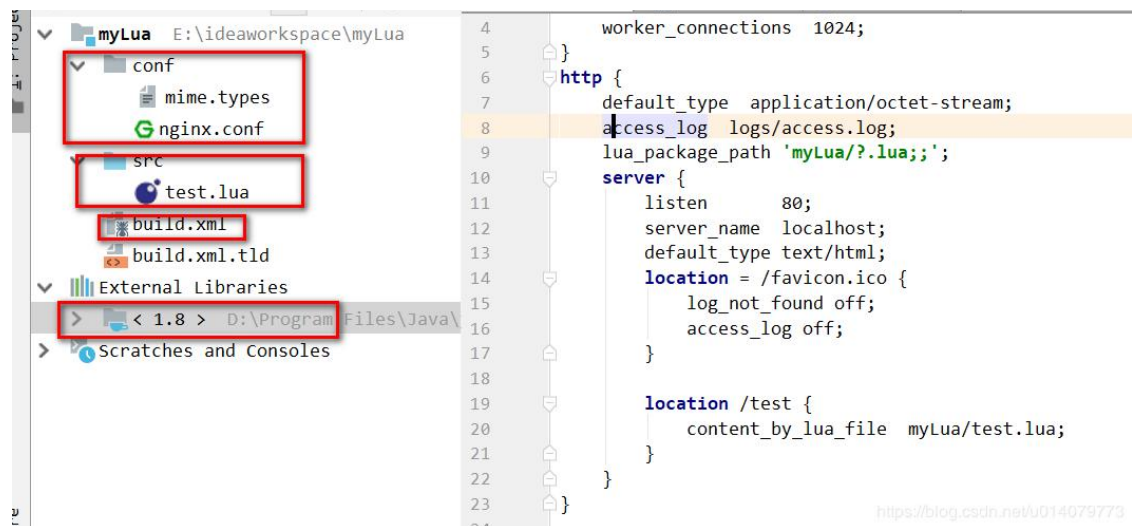


到此环境配置结束，接下来测试案例：

创建一个Lua工程：



创建好后，在根目录下创建conf文件夹， build.xml文件，在conf文件夹下创建nginx.conf文件，在src目录下创建test.lua文件。整个项目结构如下：



配置文件nginx.conf

```

1 worker_processes 2;
2 error_log logs/error.log info;
3 events {
4     worker_connections 1024;
5 }
6 http {
7     default_type application/octet-stream;
8     access_log logs/access.log;
9     lua_package_path 'myLua/?.lua;;';
10    server {
11        listen 80;
12        server_name localhost;
13        default_type text/html;
14        location = /favicon.ico {
15            log_not_found off;
16            access_log off;
17        }
18
19        location /test {
20            content_by_lua_file myLua/test.lua;
21        }
22    }
23 }

```

文件build.xml:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project name="myLua" default="dist" basedir=".">
3     <description>
4         run myLua
5     </description>
6     <!-- set global properties for this build -->
7     <property name="openresty-home" location="E:\openresty-1.15.8.2-win64"/>
8     <property name="conf" location="${basedir}/conf"/>
9     <property name="src" location="${basedir}/src"/>
10    <property name="target-conf" location="${openresty-home}/conf"/>
11    <property name="target-src" location="${openresty-home}/${ant.project.name}"/>
12
13    <echo>#####开发版本的ant配置#####</echo>
14    <target name="clean" depends="">
15        <echo>清理openresty目录 ${dist}下的conf,logs,janus,januslib</echo>
16        <delete dir="${target-conf}"/>
17        <delete dir="${target-src}"/>
18        <delete>

```

```

23     <target name="init" depends="clean">
24         <echo>创建安装目录</echo>
25         <mkdir dir="${target-conf}"/>
26         <mkdir dir="${target-src}"/>
27     </target>
28
29     <target name="dist" depends="init" description="generate the distribution" >
30         <echo>复制安装文件</echo>
31         <copy todir="${target-conf}">
32             <fileset dir="${conf}"></fileset>
33         </copy>
34         <copy todir="${target-src}">
35             <fileset dir="${src}"></fileset>
36         </copy>
37     </target>
38
39 </project>

```

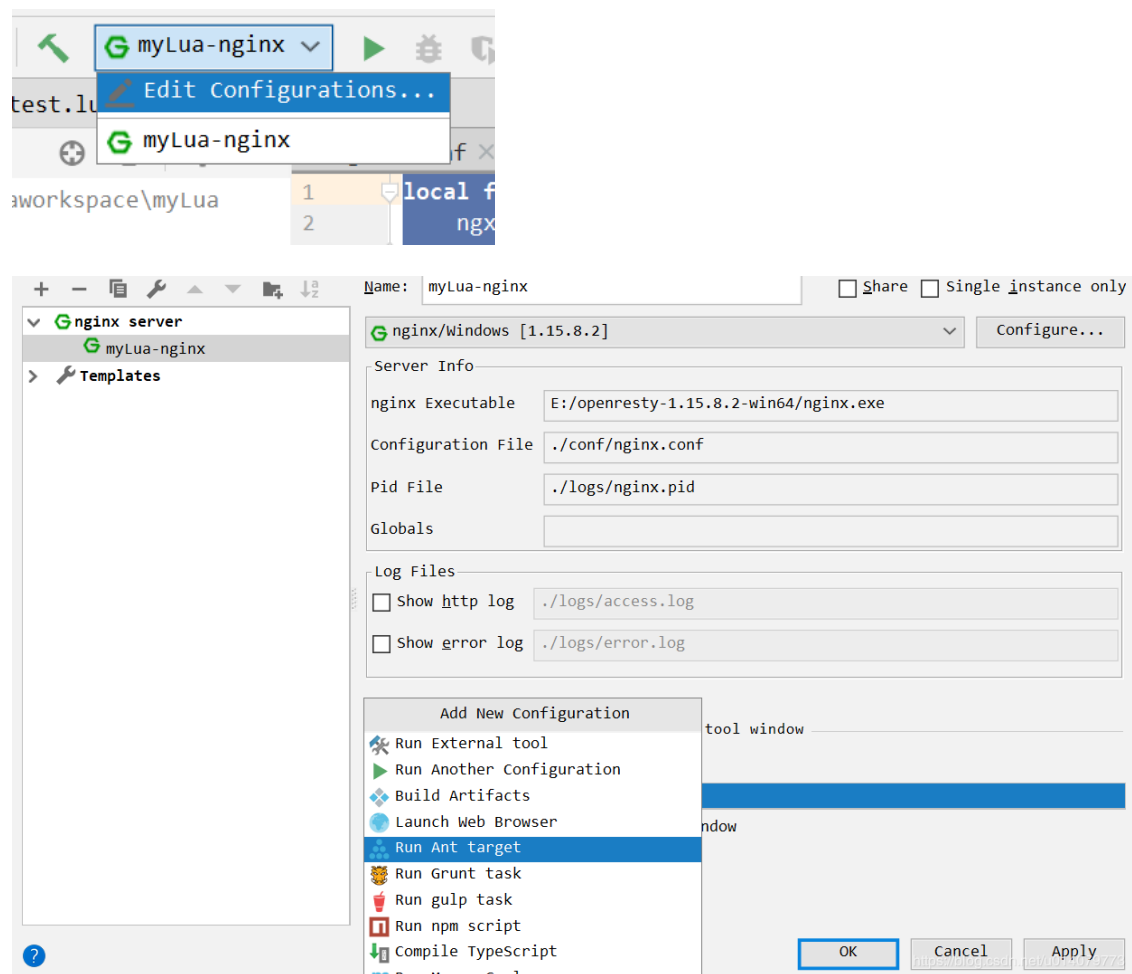
文件test.lua

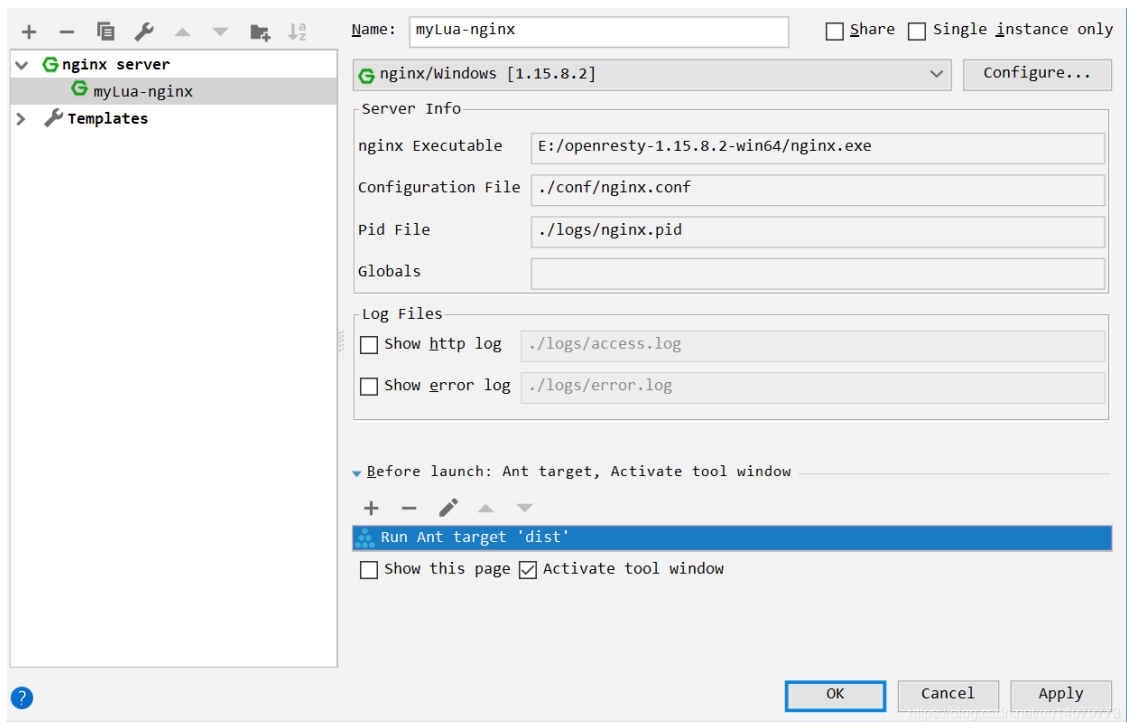
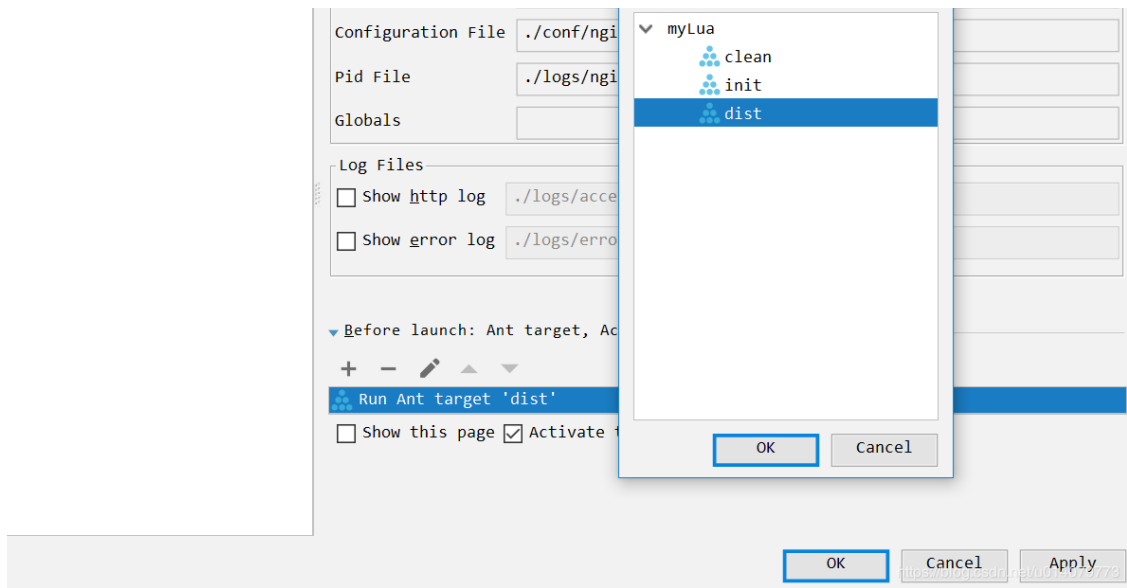
```

1 local function main()
2     ngx.say("Hello World,This is my first lua!");
3 end;
4
5 main()

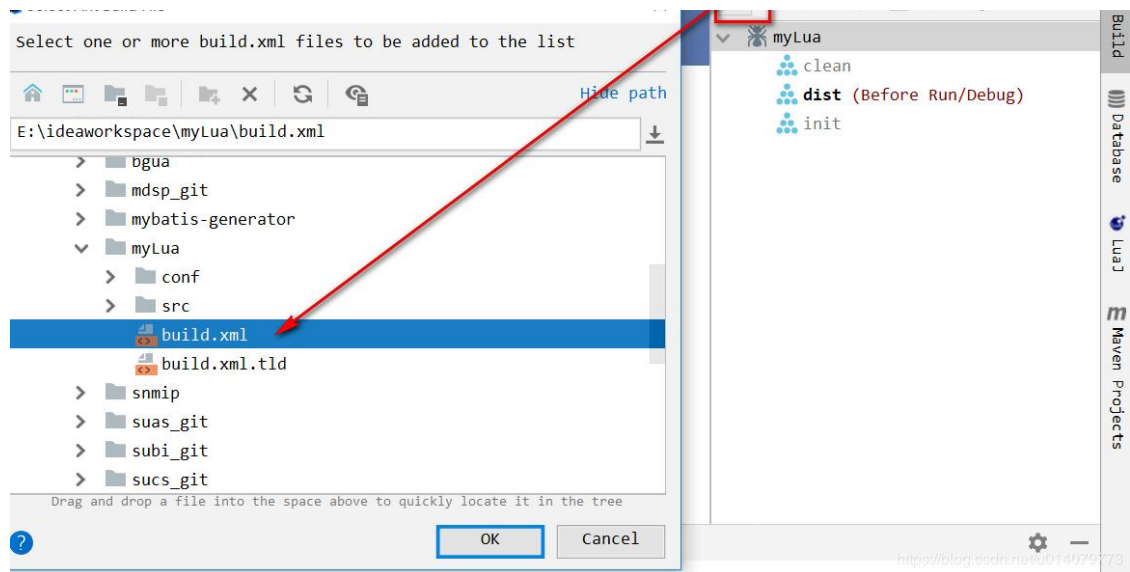
```

使用idea nginx support插件自动化构建项目，idea启动nginx

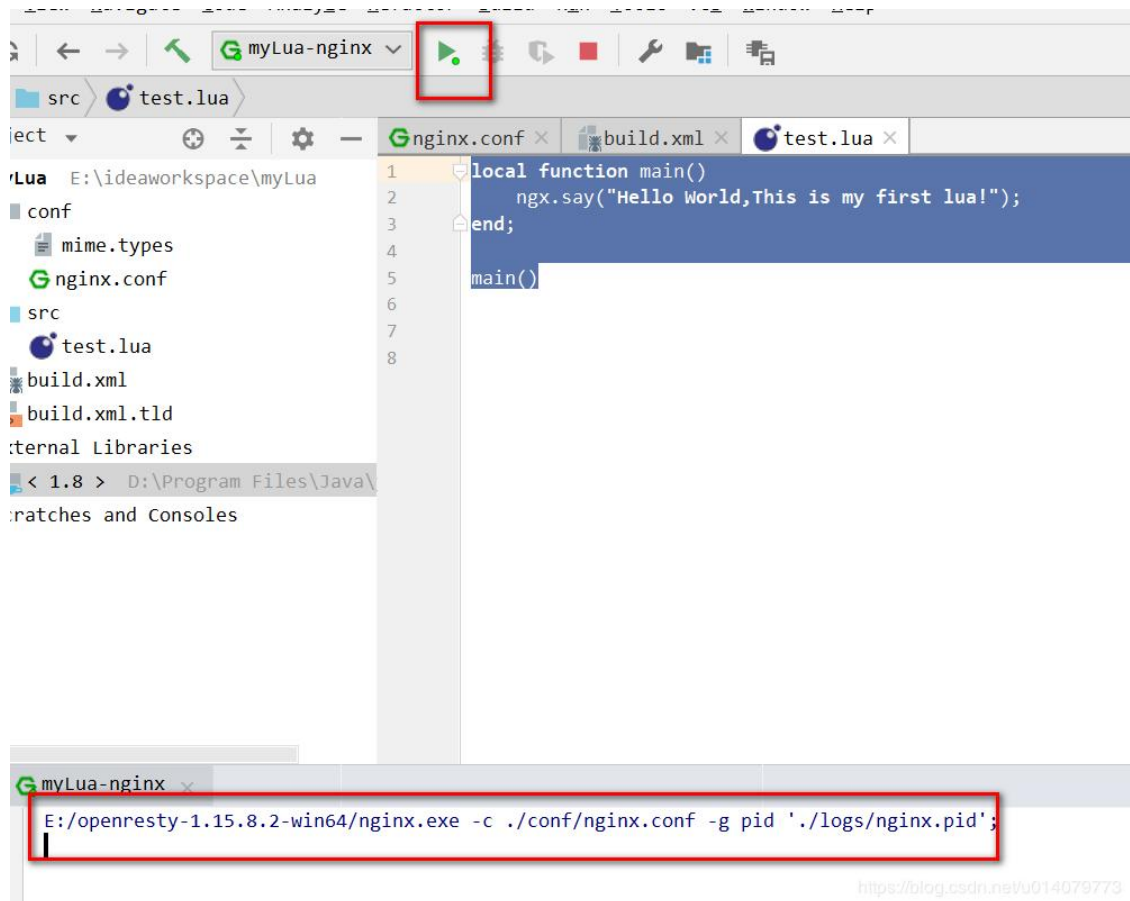




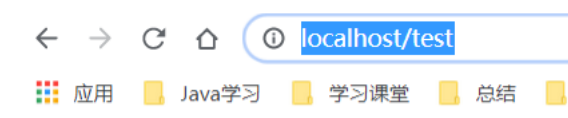
ant build添加ant target



idea启动nginx:

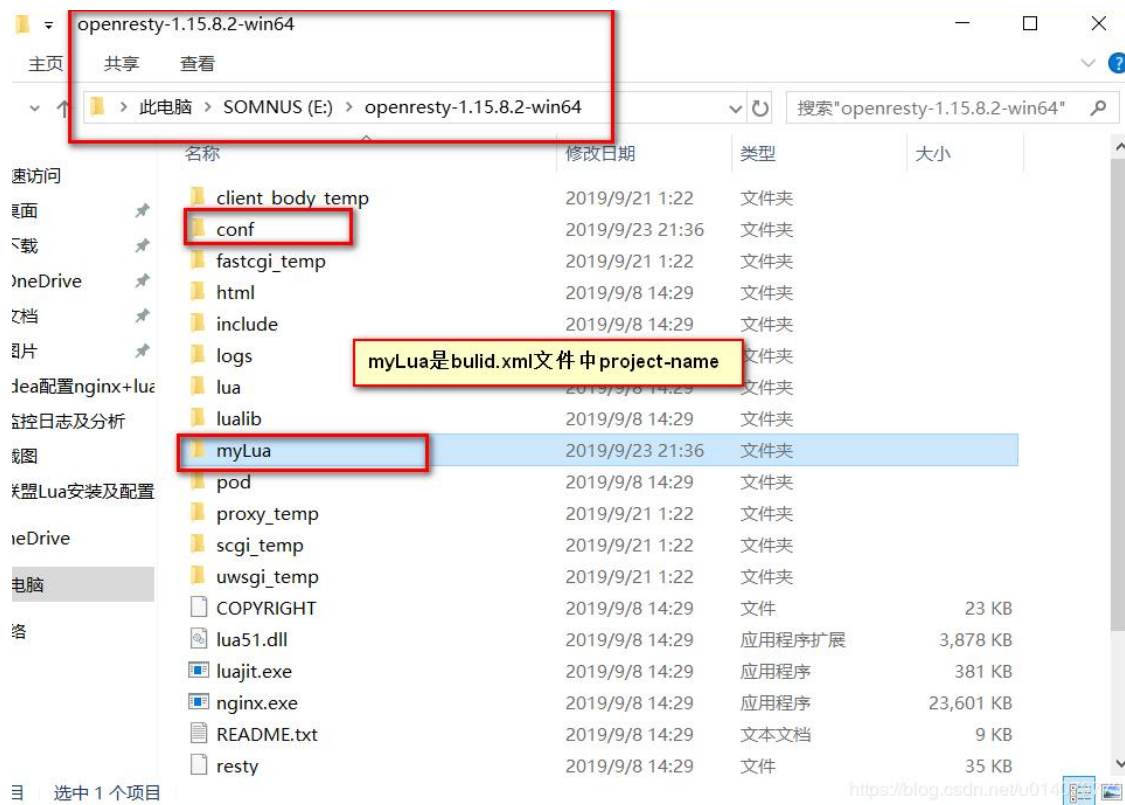


本地启动成功, 访问test.lua:<http://localhost/test>



Hello World, This is my first lua!

总结:



版权声明：本文为u014079773原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：<https://blog.csdn.net/u014079773/article/details/101220208>

智能推荐



VUE用AXIOS发送JSON数据 400 BAD REQUEST的解决

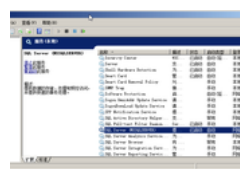
问题描述 刚开始学SpringBoot + vue.js前后端分离开模式。写了个demo，前端以JSON格式将参数数据传到后端，后端用==@RequestBody==注解接收。vue代码如下 F12中看到请求的== Content-Type==没有设置成功，所以传到后端的不是JSON数据导致报错。解决方案 在网上收集资料后发现。原来是因为data为空，content-length为0，此时...

时...

	CPU tensor	GPU tensor
ing point	torch.FloatTensor	torch.cuda.FloatTensor
ing point	torch.DoubleTensor	torch.cuda.DoubleTensor
ing point	torch.LongTensor	torch.cuda.LongTensor
r (signed)	torch.FloatTensor	torch.cuda.FloatTensor
r (signed)	torch.DoubleTensor	torch.cuda.DoubleTensor
er (signed)	torch.FloatTensor	torch.cuda.FloatTensor
er (signed)	torch.DoubleTensor	torch.cuda.DoubleTensor
er (signed)	torch.FloatTensor	torch.cuda.FloatTensor

YOSHUA BENGIO实验室MILA开放面向初学者的PYTORCH教程

MILA 实验室近日在 GitHub 上开启了一个初学者入门项目，旨在帮助 MILA 新生快速掌握机器学习相关的实践基础。目前该项目已经提供了一系列的 PyTorch 入门资料，并从张量、自动微分、图像识别、神经机器翻译和生成对抗网络等方面详细阐述。项目地址：https://github.com/mila-udem/welcome_tutorials PyTorch 是 Torch 在 Pyth...



SQL SERVER安装详细讲解

了解SQL SERVER 2008的安装过程中的关键问题；掌握服务管理器的启动和停止方法；掌握注册服务器的步骤。步骤分析 1. 安装SQL Server 2008。SQL server下载地址 2 登录SQL Server 2008服务器 完成启动、暂停、停止和重新启动服务器和登录SQL Server Management Studio等操作。（1）启动、暂停、停止SQLSERVER服务；...



SQ程序 程序分析：此程序为输出某个数的平方，过大如50000会发生负溢出 运行负数 仍然适用，nice...

GodEye

显示在一行_使用基于SWIFT的一行代码自动显示LOG,NETWORK,CPU, RAM, FPS等

GodEye 使用基于Swift的一行代码自动显示Log, Crash, Network, ANR, Leak, CPU, RAM, FPS, NetFlow, Folder等。就像上帝睁开眼睛 功能 监控日志并区分日志类型. 监控崩溃, 包括Uncatched Exception和Signal崩溃. 监控网络, 包括请求和响应的所有信息. 监控卡顿, 并告诉你卡顿时候的所有线程的堆栈. 监控内存泄露, 告诉你泄露...

猜你喜欢



用C语言写一个自动售货机

自动售货机 如图所示的简易自动售货机，物品架1、2上共有10样商品，按顺序进行编号，分别为1-10。同时标有价格与名称，-一个编号对应一个可操作按钮，供选择商品使用。如果物架上的商品被用户买走，储物柜中会自动取出商品送到物架上，保证物品架上一定会有商品.用户可以一次投入较多钱币，并可以选择多样商品，售货机可以一次性将商品输出并找零钱。用户购买方法：(1)放入钱币，用户可以以此放入多个纸币，面...

0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

关联度分析法-灰色关联分析

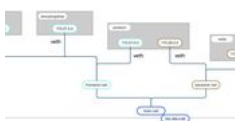
目录 灰色关联分析介绍 灰色关联分析的步骤 实例 灰色关联分析介绍 对于两个系统之间的因素，其随时间或不同对象而变化的关联性大小的量度，称为关联度。在系统发展过程中，若两个因素变化的趋势具有一致性，即同步变化程度较高，即可谓二者关联程度较高；反之，则较低。因此，灰色关联分析方法，是根据因素之间发展趋势的相似或相异程度，亦即“灰色关联度”，作为衡量因素间关联程度的一种方法。 ...



C FUN函数求N个整数的平均值_PYTHON语法示例——函数

12_Python语法示例(函数) 1.写一个函数mysum,此函数带有两个参数x,y打印出两个参数x,y的和 2.写一个函数print_even,传入一个数参n代表终值整数(不包含n),打印0-n之间所有偶数 3.写一个函数mymax,返回两个数的最大值 4.写一个函数input_number用于读取输入多个整数,输入负数时结束将输入的数形成列表返回给调用者 5.写一个函数mysum,可以传入...

DOCKER COMPOSE



Docker Compose WHAT Docker Compose VS Docker Stack WHY 发布服务 HOW 服务拆分 YML定义 构建或拉取镜像 配置环境变量 配置网络 配置数据卷 构建运行 参考 WHAT Docker Compose是用Python编写的项目，并不包含在Docker Engine安装包中。Docker Compose用于解析YML文件并运行YML中定义的容器...



会保证保持线性可分性。目的是使用SVM公式找到一个最佳的线性可分平面。分类器。在这片博文中将会包括一些核心代码。所有代码整理成Jupyter notebook，发布在我的github上。Jupyter Notebook教程中最后分类器实现效果的可视化 1 什么是硬边界SVM? 简单来说，在二维情况下，就是对于线性可分的一些点（包含正例和负例），找一条线能将他们分开。同时，能保证线离点距...