# nexus3 上搭建npm本地服务器

撰写时间：20180509

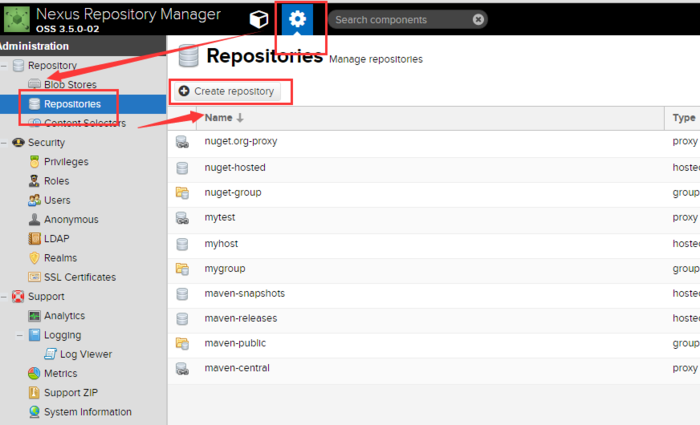
撰写人： 刘鸿飞

## nexus的说明文档

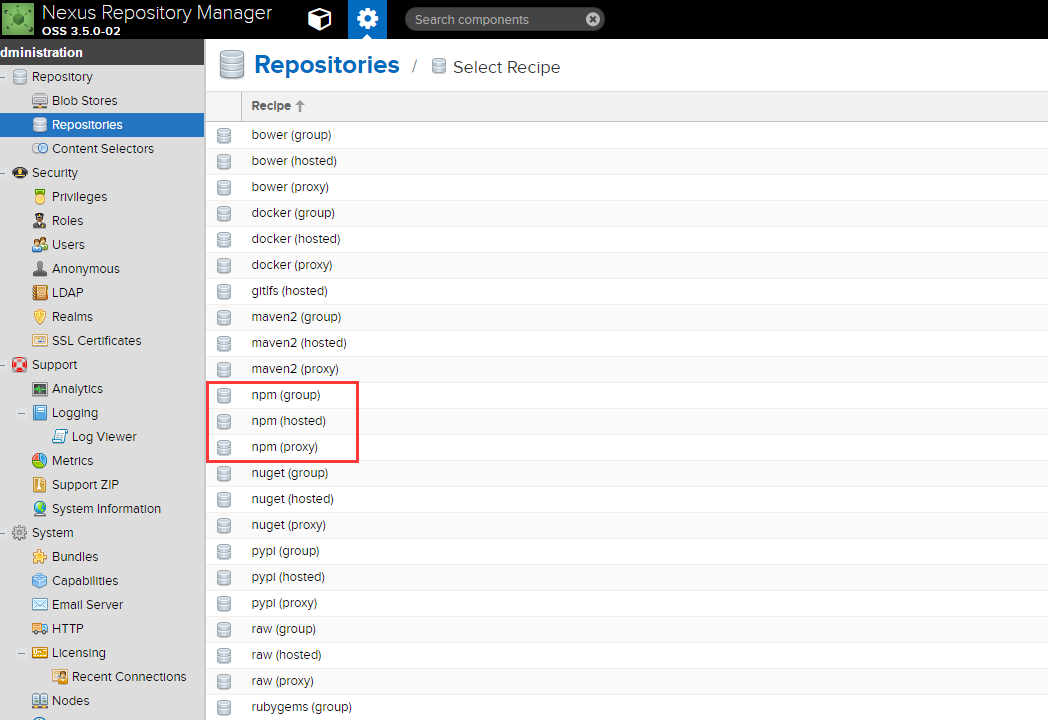
nexus的说明文档网址https://oss.sonatype.org/#Documentation，有什么问题大家可以直接参考官方文档

## Nexus Repository Manager 中添加 npm

登陆后如图所示：



这里有三个类型的npm选项：



这里面一共有3中类型的npm仓库（具体参考开发文档）：

* Proxying npm Registries 代理仓库

To reduce duplicate downloads and improve download speeds for your developers and CI servers, you should proxy the registry hosted at [https://registry.npmjs.org](https://registry.npmjs.org/). By default npm accesses this registry directly. You can also proxy any other registries you require.   
大致的意思就是说，可以访问远程仓库的一些代码，比如angular的代码

* Private npm Registries 托管仓库

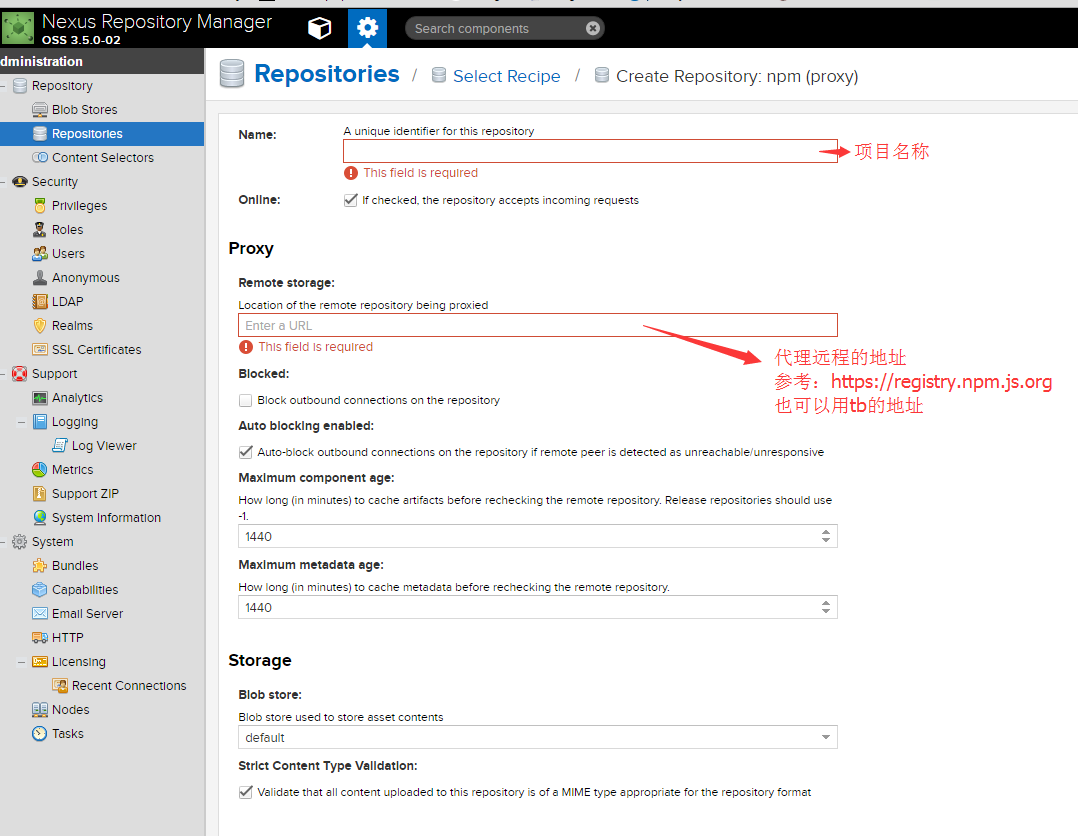
A private npm registry can be used to upload your own packages as well as third-party packages. You can create a private npm registry by setting up a hosted repository with the npm format in the repository manager. It is good practice to create two separate hosted repositories for these purposes.   
大致的意思是说，这是你的私服，可以放你要放到自己的npm私服上的代码

* Grouping npm Registries 存储库组

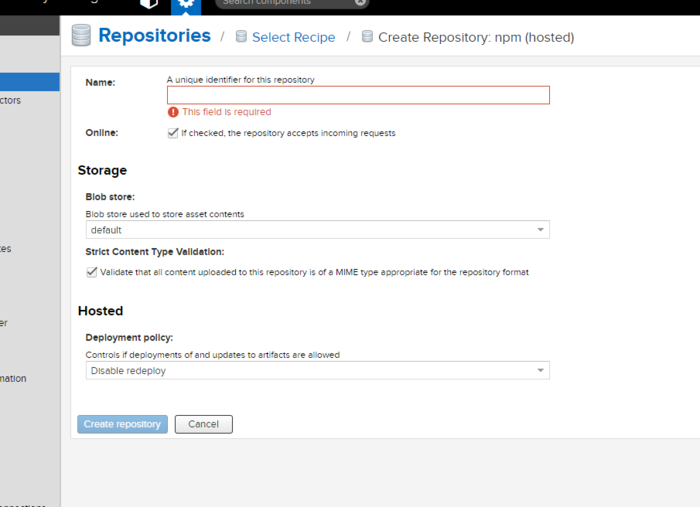
A repository group is the recommended way to expose all your npm registries repositories from the repository manager to your users, without needing any further client side configuration. A repository group allows you to expose the aggregated content of multiple proxy and hosted repositories with one URL to npm and other tools.   
这是一个综合性的库组，可以将请两个综合起来，下面有详细的介绍。

## 创建代理仓库

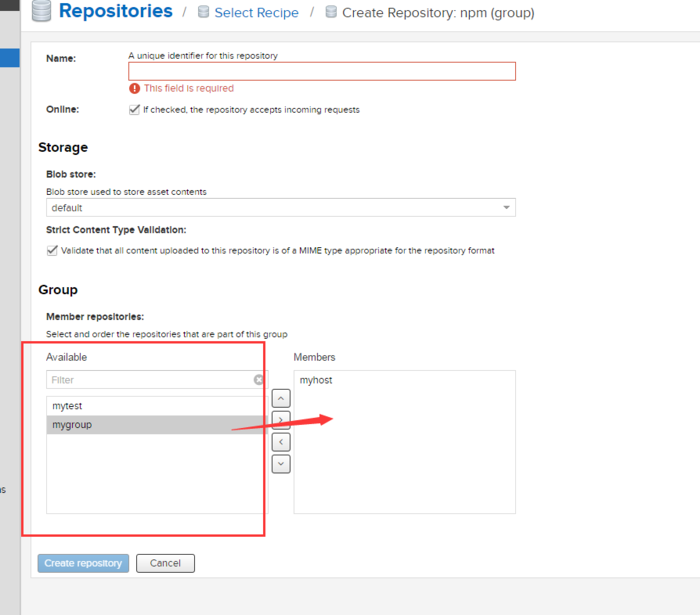
1. 创建 Proxying npm Registries 代理仓库



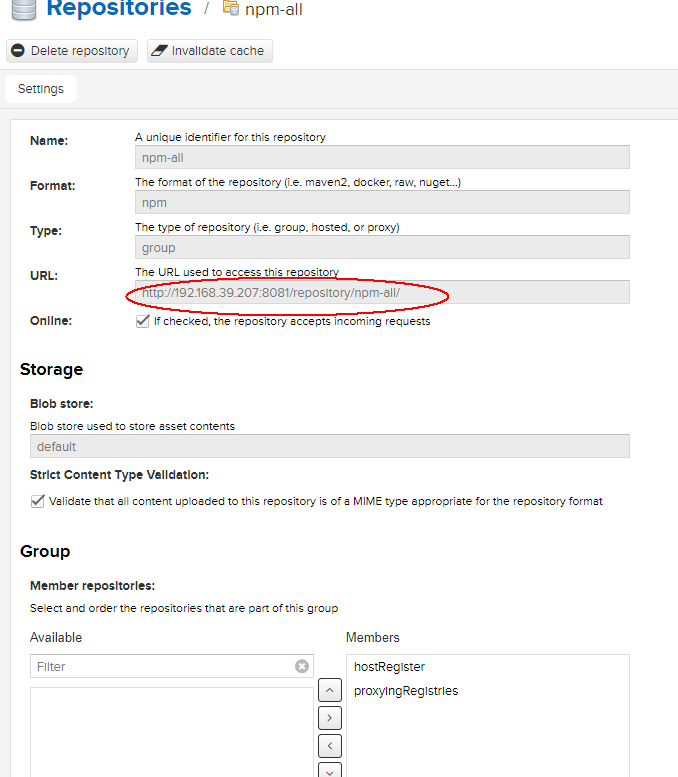
1. 创建 Hosted npm Registries 代理仓库



1. 创建 Grouping npm Registries 仓库  
   需要注意的是组仓库要把之前的 proxy 和 hosted 都添加上。



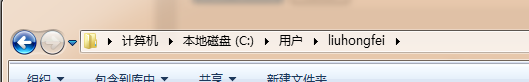
Grouping npm Registries 仓库地址如下图所示



**地址配置**

在home目录下，的.npmrc文件中添加 registry = http://192.168.39.207:8081/repository/npm-all/ 。这个路径要和自己的路径一致，并且最后的rep名要和group相同。

**win下 home路径**



如果该路径下没有.npmrc这个文件，那么就自己创建即可。

linux环境下

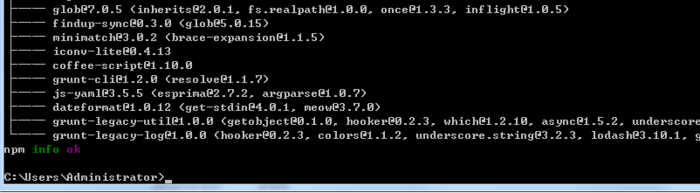
npm config set registry http://192.168.39.207:8081/repository/npm-all/

到这里，npm的私服创建完毕，接下来，我们要验证一下了。

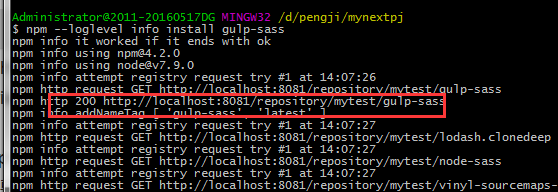
## 验证npm私服是否创建成功

随便在一个目录下，进入cmd，然后执行npm –loglevel info install grunt，看看是够能够下载成功。如果下载失败，可多试几次。

只是下载成功是不够，还要看下，是不是从你的npm私服中进行下载的。



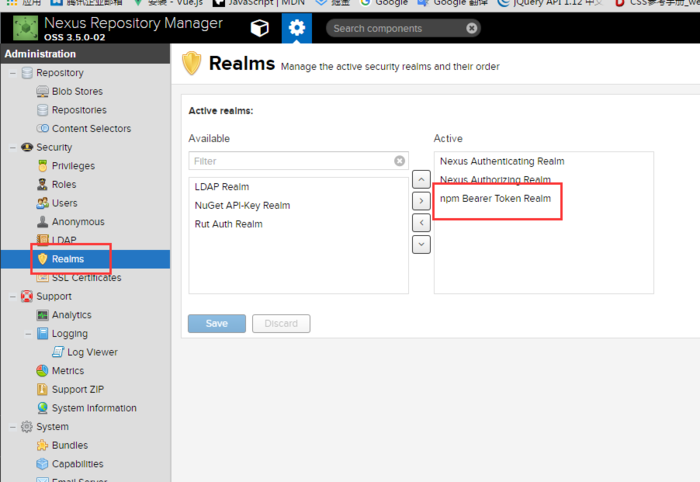
此时可以验证一下 是否依赖走了代理：



## 设置权限并登陆

没有设置权限认证则不能向私服上传代码。

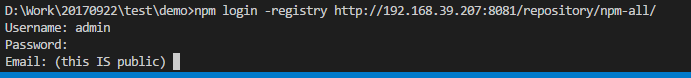
添加权限认证很好添加的，在Nexus上的realms中添加上npm Bearer Token Realms即可。



// 使用

$ npm login –registry http://192.168.39.207:8081/repository/npm-all/

// 进行登陆，需要填写账号、密码以及邮箱。



## 将模块或组件发布到私服

要发布的模块，必须保证在根目录下有package.json文件，否则会报错。

pachage.json文件可以用 npm init 直接自动生成。

{

"name": "test",

"version": "1.0.1"

"publishConfig" : {

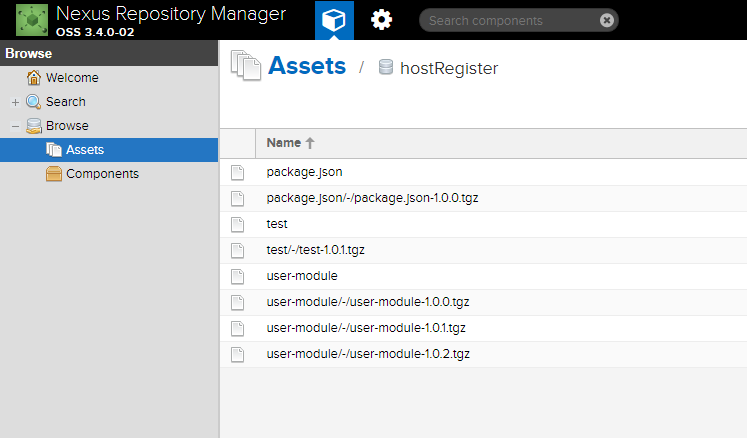
"registry" : " http://192.168.39.207:8081/repository/hostRegister/"

} // 发布路径的配置

}

发布直接用 npm publish就可以发布了。

发布之后可以在 系统中的 左侧browse 中的assets中看到上传的资源



group 中可以同时看到 proxy 和 hosted 的资源， proxy会存下第一次下载过的目录，之后再下载时走的是代理，hosted存的是本地上传的包资源。

然后可以在本地下载一下刚才上传到hosted的包执行命令：



下载成功

