**DiOne应用统一开发环境指南（Python）**

**一：背景**

DiOne应用开发需要依赖django，celery，mysql-client等第三方库，而像celery又依赖其他多个库，新手在搭建本地开发环境时，过程繁琐，问题百出。

不同开发者，开发环境不同， 有window，有mac。window版本也不完全一致，开发者需要一个统一开发环境，方便调试。 DiOne提供的统一开发环境和线上使用的系统、库版本一致。

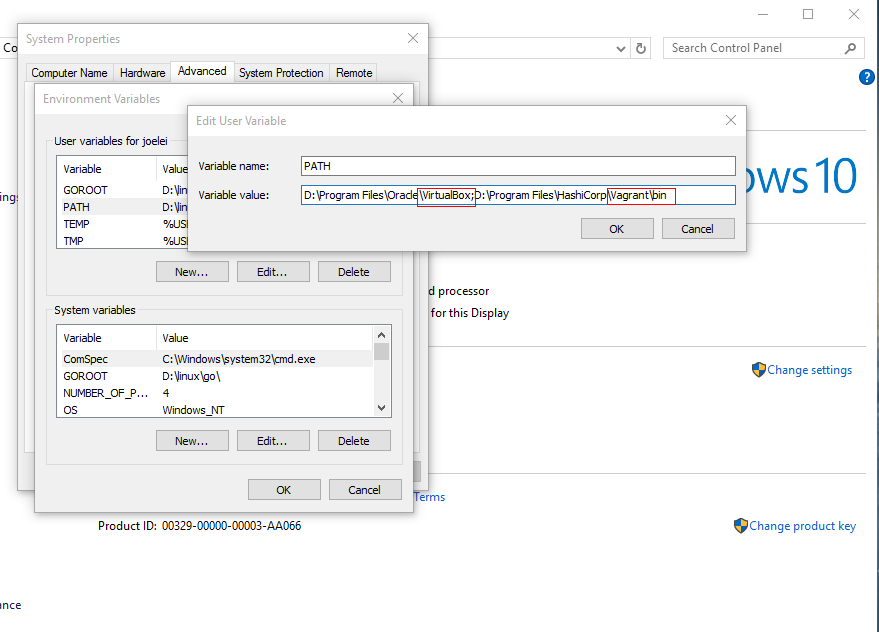
想要在多平台下开发，必须依赖于虚拟环境，DiOne提供统一开发环境使用virtualbox做hypervisor，使用vagrant做管理工具，下面对使用方法做详细介绍。

**二：vagrant安装配置**

下面以window版本为例子，假设安装目录为D:\Program Files。

1，安装vagrant，添加***D:\Program Files\Vagrant\bin***到PATH环境变量

       外网下载地址：[https://www.vagrantup.com/downloads.html](https://www.vagrantup.com/downloads.html" \t "_blank)



2，安装virtualbox ，根据提示一步一步安装完成后，添加***D:\Program Files\VirtualBox***到PATH环境变量

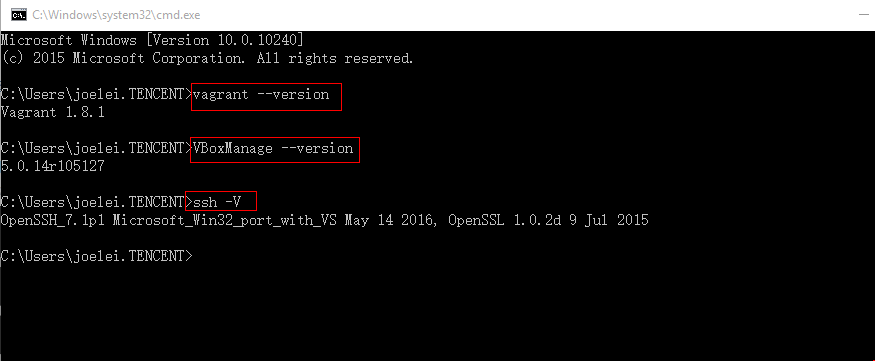
       外网下载地址：[https://www.virtualbox.org/wiki/Downloads](https://www.virtualbox.org/wiki/Downloads" \t "_blank)

3，安装ssh客户端（如果本地已经有，请跳过这一步）

vagrant使用ssh命令需要本地装ssh命令行客户端（注意xshell, SecureCRT等GUI等不能被使用）

下载完成后，解压到任意目录，如***D:\Program Files\OpenSSH-Win32***，添加到PATH环境变量

       外网下载地址：https://github.com/PowerShell/Win32-OpenSSH/releases/download/v0.0.12.0/OpenSSH-Win32.zip

 运行命令vagrant，VBoxManage(Virtualbox管理工具) 和ssh，能正常返回版本号，说明安装成功。  


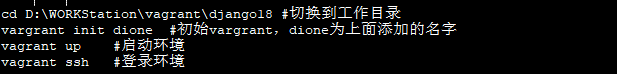
**三：添加DiOne镜像到vagrant**

vagrant需要基础镜像才能使用，DiOne已经制作好了应用开发镜像，里面包含了My-Sql, Rabbit-MQ, django等应用开发所需的环境，即开即用。

下载地址：http://dpass-1251144231.cossh.myqcloud.com/dione-django1.8-u2.boxdione-django1.8.box是下载的镜像名称，dione是添加的镜像名称，注意这个后面会需要用到。

**四：初始化环境**

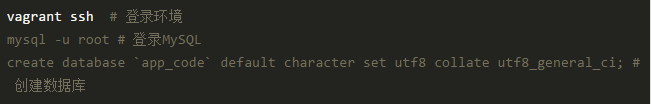
vagrant第一次使用需要初始化环境，如开发环境为***D:\WorkStation\vagrant\django18***，切换到工作目录，下文所有命令都是在工作目录下执行。

vagrant会在当前目录新建.vagrant目录并生成Vagrantfile配置文件，修改Vagrantfile，可以定制化vagrant，如端口映射，同步目录等   
注意：

1. 默认**vagrant会把当前目录做共享文件夹**，挂载到/vagrant下
2. DiOne镜像root和vagrant用户的密码都是vagrant

**1.创建数据库**

DiOne统一开发环境已经安装mariadb 5.5(兼容mysql5.5)，新的应用使用需要先创建数据库



**2.APP代码获取，切换工作目录**

  vagrant默认会当前目录挂着到/vagrant目录下，把应用的代码拷贝到***D:\WorkStation\vagrant\django18（初始化工作目录)，***在虚拟环境中即可看到应用的代码，假设应用为test-bk1， 切换到应用的工作目录***/vagrant/test-bk1/***

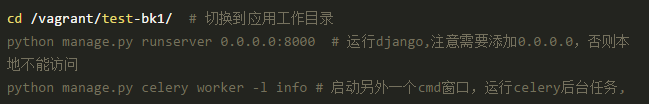


**3.创建表结构&升级表结构（默认仅支持django1.8）**

新的APP或者APP有数据库变更时，需要migrate到数据库中才能使用（django migrate请查看[https://docs.djangoproject.com/en/1.8/topics/migrations/](https://docs.djangoproject.com/en/1.8/topics/migrations/" \t "_blank)）

**五：运行环境，应用开发**

切换到应用的工作目录***/vagrant/test-bk1/***



在本地浏览器即可，访问8000端口即可

**六：Vagrant常用命令**



**七：Vagrantfile配置文件详解**

在开发目录下有一个文件Vagrantfile，里面包含有大量的配置信息，主要包括三个方面的配置，虚拟机的配置、SSH配置、Vagrant的一些基础配置。Vagrant是使用Ruby开发的，所以它的配置语法也是Ruby的，但是我们没有学过Ruby的人还是可以跟着它的注释知道怎么配置一些基本项的配置。

**网络设置**

Vagrant有两种方式来进行网络连接，一种是host-only(主机模式)，意思是主机和虚拟机之间的网络互访，而不是虚拟机访问internet的技术，也就是只有你一個人自High，其他人访问不到你的虚拟机。另一种是Bridge(桥接模式)，该模式下的VM就像是局域网中的一台独立的主机，也就是说需要VM到你的路由器要IP，这样的话局域网里面其他机器就可以访问它了，一般我们设置虚拟机都是自high为主，所以我们的设置一般如下：

config.vm.network :private\_network, ip: “11.11.11.11”   
这里我们虚拟机设置为hostonly，并且指定了一个IP，IP的话建议最好不要用192.168..这个网段，因为很有可能和你局域网里面的其它机器IP冲突，所以最好使用类似11.11..这样的IP地址。

**同步目录**

我们上面介绍过/vagrant目录默认就是当前的开发目录，这是在虚拟机开启的时候默认挂载同步的。我们还可以通过配置来设置额外的同步目录：

config.vm.synced\_folder “/Users/astaxie/data”, “/vagrant\_data”   
上面这个设定，第一个参数是主机的目录，第二个参数是虚拟机挂载的目录

**端口转发**

config.vm.network :forwarded\_port, guest: 8000, host: 8000   
这一行的意思是把对host机器上8000端口的访问请求forward到虚拟机的8000端口的服务上，例如你在你的虚拟机上使用nginx跑了一个Go应用，那么你在host机器上的浏览器中打开[http://localhost:8000](http://localhost:8080/" \t "_blank)时，Vagrant就会把这个请求转发到VM里面跑在8000端口的nginx服务上，因此我们可以通过这个设置来帮助我们去设定host和VM之间，或是VM和VM之间的信息交互。

修改完Vagrantfile的配置后，记得要用vagrant reload命令来重启VM之后才能使用VM更新后的配置

更多内容请查阅官方文档 [https://www.vagrantup.com/docs/cli/index.html](https://www.vagrantup.com/docs/cli/index.html" \t "_blank)

**注意：**

**Virtualbox虚拟化需要BIOS开启虚拟化支持，[http://jingyan.baidu.com/article/8ebacdf0df465b49f65cd5d5.html](http://jingyan.baidu.com/article/8ebacdf0df465b49f65cd5d5.html" \t "_blank)**