

基于OSGi搭建动态化的系统 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E5_9F_BA_E4_BA_8EOSGi_c104_145584.htm 搭建动态化的系统是作为java开发人员一直就非常追求的目标，一个系统能够动态化就意味着：
添加新功能时不需要重启系统；
修改已存在的功能时不需要重启系统；
删除一些不需要的功能时不需要重启系统；
修改系统中的配置时不需要重启系统即刻生效；
系统的业务行为可动态的改变。也许习惯了传统java开发方式的人而言，没有这些动态化也没什么，但不可否认，这些动态化的特征还是非常吸引人的，尤其是如果能很容易就获得这些好处，那么自然就不会错过这些好处了，基于OSGi可以很容易的让我们获取到这些好处，对于上面所描述的系统的动态化的特征可以归纳为这么三点：
模块的动态化：新增、修改、删除以及扩展；
配置的动态化；
行为的动态化。下面就来看看基于OSGi是怎么样来构成这样的动态化的系统的。
模块的动态化：新增、修改、删除以及扩展；
新增 对于模块的新增，在OSGi可以通过在控制台中使用install命令去安装，或者调用BundleContext的installBundle方法去安装新的模块；
修改 当修改了模块后，只需在控制台中使用update命令去更新模块，或者通过调用BundleContext.getBundle(Bundle的id).update()；
删除 当需要删除不需要的了的模块时，只需在控制台中使用uninstall命令去删除，或者通过调用BundleContext.getBundle(Bundle的id).uninstall()。
扩展 对于模块的扩展，可以使用OSGi所支持的调用多服务的方式或者使用Equinox提供的扩展点的方式

来实现。 配置的动态化 对于配置的动态化方面，使用OSGi的Configuration Admin Service(简称CM)可以很容易的实现，使用CM使得系统中的配置可以采用统一的形式去管理、动态的通知变化以及使用。 行为的动态化 系统在运行中要根据业务的需求选择相应的服务是经常会碰到的需求，例如一份文件在某种情况下可能需要进行存储，而在另一种情况下可能是需要进行提取信息并存储为其他文件，由于OSGi支持服务选择时的过滤(如context.getAllServiceReferences(服务接口完整类名,"(key=db)")),使得系统可以在运行时根据需求来选择或配置使用相应的服务，而OSGi同时又支持了服务的动态化的管理，当服务可用时会自动的通知，当其不可用时同样会自动的通知，这就保证了开发人员在使用服务时可做出相应的处理。如果觉得你所设计的系统需要这些动态化的特征的话，那么建议使用下OSGi(推荐的OSGi框架：Equinox)试试，或者可以参考下OSGi的实现方法自己去实现所需要的动态化的特征。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com