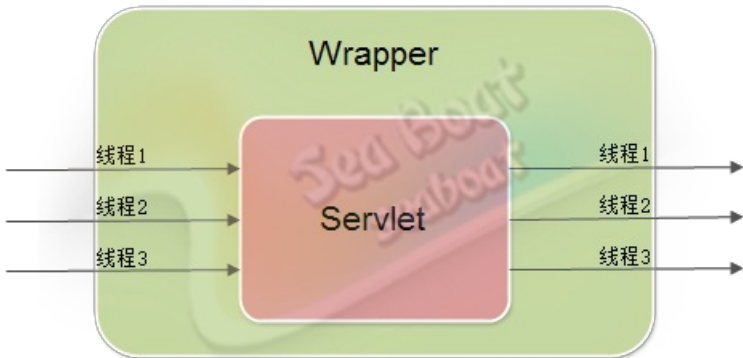


tomcat中Servlet对象池介绍及如何使用

tomcat中Servlet对象池

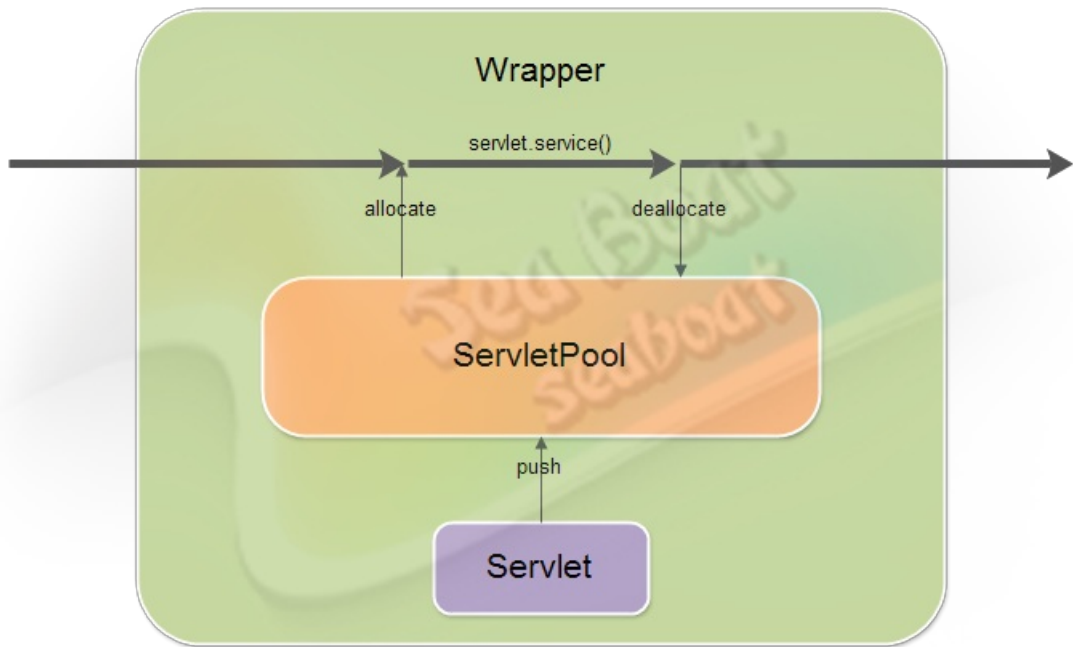
Servlet在不实现SingleThreadModel的情况下运行时是以单个实例模式，如下图，这种情况下，Wrapper容器只会通过反射实例化一个Servlet对象，对应此Servlet的所有客户端请求都会共用此Servlet对象，而对于多个客户端请求tomcat会使用多线程处理，所以应该保证此Servlet对象的线程安全，多个线程不管执行顺序如何都能保证执行结果的正确性。例如刚做web应用开发时可能会犯的一个错误：在某个Servlet中使用成员变量累加去统计访问次数，这就存在线程安全问题。



为了支持一个Servlet对象对应一个线程，Servlet规范提出了一个SingleThreadModel接口，tomcat容器必须要完成的机制是：如果某个Servlet类实现了SingleThreadModel接口则要保证一个线程独占一个Servlet对象。假如线程1正在使用Servlet对象1，则线程2只能用Servlet对象2。

针对SingleThreadModel模式，tomcat的Wrapper容器使用了对象池策略，Wrapper容器会有一个Servlet堆保存若干个该Servlet对象，当需要该Servlet对象时从堆中pop一个对象，而当用完后则push回堆中。Wrapper容器中最多可以有20个该Servlet对象，例如xxxServlet类的对象池，已经有20个线程占用了20个对象，那么第21个线程执行时就会阻塞等待，直到对象池中有可用的对象才继续执行。

整个流程如下图所示，某个线程处理客户端请求，它首先尝试从Servlet对象池中获取Servlet对象，此时如果对象池有可用对象则直接返回一个对象，如果不够使用则继续实例化Servlet对象并push进对象池，但Servlet对象的总数量必须保证在20个以内，如果20个Servlet对象都被其他线程使用了，那么就必须要等到其他线程用完后放回后才能获取，此时该线程会一直阻塞等待。从对象池中获取到Servlet对象后则调用Servlet对象的service方法对客户端请求进行处理，处理完后再将Servlet对象放回对象池中。



本节介绍了Servlet对象池，它是为了支持Servlet规范SingleThreadModel接口而引入的，它就是一个栈结构，需要时就pop一个对象，使用完就push回去。

感谢阅读，希望能帮助到大家，谢谢大家对本站的支持！

您可能感兴趣的文章:Tomcat 检测内存泄漏实例详解如何通过tomcat的ManagerServlet远程部署项目servlet和tomcat_动力节点Java学院整理tomcat中Servlet的工作机制详细介绍Tomcat报错：HTTP Status 500 (Wrapper cannot find servlet class)解决办法tomcat报错：Wrapper cannot find servlet class ...问题解决Spring关闭Tomcat Servlet容器时内存泄漏问题解决方案