

八、Tomcat中关于Session的实现

标准会话对象——StandardSession

标准会话管理器——StandardManager

标准会话对象——StandardSession

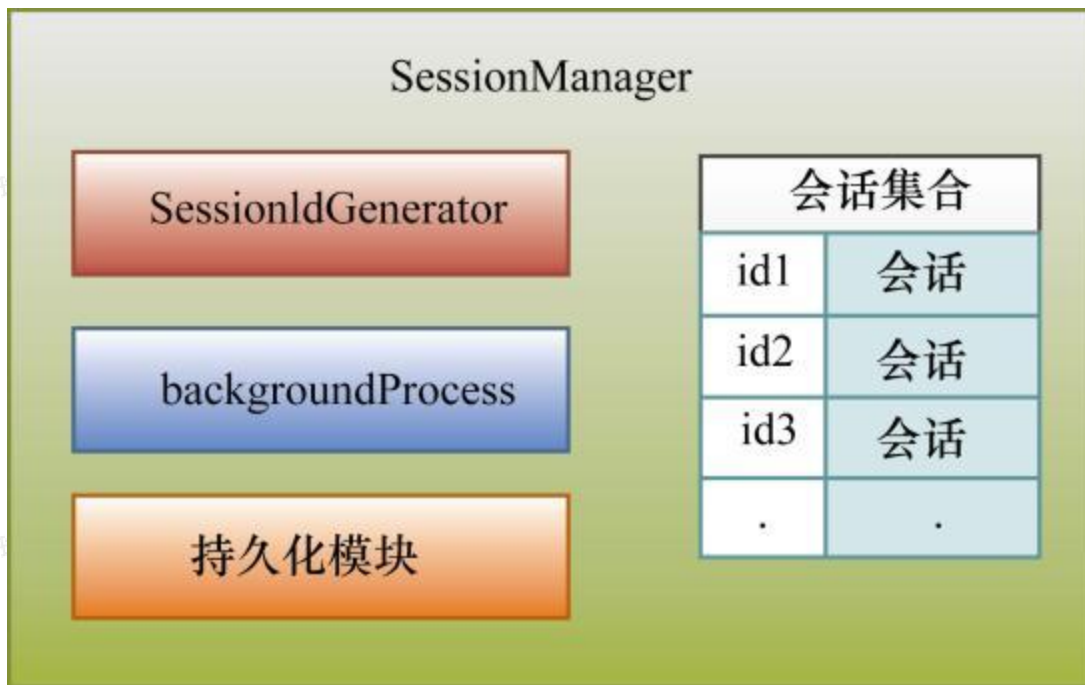
- Session 接口定义了 Tomcat 内部用来操作会话的一些方法。
- Serializable 则是序列化接口，实现它是为了方便传输及持久化。
- HttpSession 是 Servlet 规范中为会话操作而定义的一些方法，作为一个标准 Web 容器，实现它是必然的。

一个最简单的标准会话应该包括 ID 和 Map<String, Object> 结构的属性。其中，ID 用于表示会话编号，它必须是全局唯一的，属性用于存储会话相关信息，以 KV 结构存储。另外，它还应该包括会话创建时间、事件监听器，并且提供 Web 层面访问的外观类等。

标准会话管理器——StandardManager

用于保存状态的会话对象已经有了，现在就需要一个管理器来管理所有会话。例如，会话 ID 生成，根据会话 ID 找出对应的会话，对于过期的会话进行销毁等操作。用一句话描述标准会话管理器：提供一个专门管理某个 Web 应用所有会话的容器，并且会在 Web 应用启动、停止时进行会话重加载和持久化。

会话管理主要提供的功能包括会话 ID 生成器、后台处理（处理过期会话）、持久化模块及会话集的维护。如下图所示，标准会话管理器包含了 SessionIdGenerator 组件、backgroundProcess 模块、持久化模块以及会话集合。



首先看 SessionIdGenerator，它负责为每个会话生成、分配一个唯一标识。例如，最终会生成类似「326257DA6DB76F8D2E38F2C4540D1DEA」字符串的会话标识。具体的默认生成算法主要依靠 JDK 提供的 SHA1PRNG 算法。在集群环境中，为了方便识别会话归属，它最终生成的会话标识类似于「326257DA6DB76F8D2E38F2C4540D1DEA.tomcat1」，后面会加上 Tomcat 的集群标识 jvmRoute 变量的值，这里假设其中一个集群标识配置为「tomcat1」。如果你想置换随机数生成算法，可以通过配置 server.xml 的 Manager 节点 secureRandomAlgorithm 及 secureRandomClass 属性达到修改算法的效果。

然后看如何对过期会话进行处理。负责判断会话是否过期的逻辑主要在 backgroundProcess 模块中，在 Tomcat 容器中会有一条线程专门用于执行后台处理。当然，也包括标准会话管理器的 backgroundProcess，它不断循环判断所有的会话中是否有过期的，一旦过期，则从会话集中删除此会话。

最后是关于持久化模块和会话集的维护。由于标准会话旨在提供一个简单便捷的管理器，因此持久化和重加载操作并不会太灵活且扩展性弱，Tomcat 会在每个 StandardContext (Web 应用) 停止时调用管理器将属于此 Web 应用的所有会话持久化到磁盘中。文件名为 SESSIONS.ser，而目录路径则由 server.xml 的 Manager 节点 pathname 指定或 Javax.servlet. context.tempdir 变量指定，默认存放路径为 %CATALINA_HOME%/work/Catalina/localhost/ WebName/SESSIONS.ser。当 Web 应用启动时，又会加载这些持久化的会话，加载完成后，SESSIONS.ser 文件将会被删除，所以每次启动成功后就不会看到此文件的存在。另外，会话集的维护是指提供创建新会话对象、删除指定会话对象及更新会话对象的功能。