链滴

数据库设计之权限设计

作者：alex18595752445

原文链接：https://ld246.com/article/1583375448624

来源网站：链滴

<h2 id="RBAC-介绍--权限-">RBC 介绍 (权限)</h2>
RBAC 是什么？
RBAC 是基于角色的访问控制（<ode>Role-Based Access Control</code> ）在 RBAC 中，权限与角色相关联，用户通过成为适当角色的成员而得到这些角色的权限。这极大地简化了权限的管理。这样管理都是层级相互依赖的，权限赋予给角色，而把角色又赋予用户，样的权限设计很清楚，管理起来很方便。
<h2 id="RBAC-介绍-">RBAC 介绍。</h2>
RBAC 认为授权实际上是 <code>Who</code> 、<code>What</code> 、<code>How</code> 三元组之间的关系，也就是 <coe>Who</code> 对 <code>What</code> 进行 <code>How</code> 的操作，也就是“主体”“客体”的操作。
Who：是权限的拥有者或主体（如：User，Role）。
What：是操作或对象（operation，object）。
How：具体的权限（Privilege,正向授权与负向授权）。
然后 RBAC 又分为 <code>RBA0、RBAC1、RBAC2、RBAC3</code> ，如果你不知道他们有什么区别，你可以百度百科：<a href"https://ld246.com/forward?goto=http%3A%2F%2Fbaike.baidu.com%2Flink%3Furl%3DTg3nxjvD2QVLLkjKa_4XaQoOWSPAVpR1FgHAG_gANcamtN2cYIm1r1irNw9VZ816FBrMEvdoYqwzixdHd5e_" target="_blank" rel="nofollow ugc">百度百科-RBAC 估计你看不懂。还是看看我简单介绍。
我这里结合我的见解，简单的描述下（去掉那么多的废话）。
<h2 id="RBAC0-RBAC1-RBAC2-RBAC3-简单介绍-">RBAC0、RBAC1、RBAC2、RBAC3 简单介。</h2>

RBAC0：是 RBAC 的核心思想。
RBAC1：是把 RBAC 的角色分层模型。
RBAC2：增加了 RBAC 的约束模型。
RBAC3：其实是 RBAC2 + RBAC1。

<h2 id="RBAC0-RBAC-的核心-">RBAC0，RBAC 的核心。</h2>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2020/03/image-c1528049.png?imageView2/2/interlace/1/format/jpg"></>
<h2 id="RBAC1-基于角色的分层模型">RBAC1，基于角色的分层模型</h2>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2020/03/image-1f31da56.png?imageView2/2/interlace/1/format/jpg"></>
<h2 id="RBAC2-是-RBAC-的约束模型-">RBAC2、是 RBAC 的约束模型。</h2>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2020/03/image-e5bf974a.png?imageView2/2/interlace/1/format/jpg"></>
<h2 id="RBAC3-就是-RBAC1-RBAC2">RBAC3、就是 RBAC1+RBAC2</h2>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2020/03/image-45959b78.png?imageView2/2/interlace/1/format/jpg"></>
估计看完图后，应该稍微清楚一点。
下面来看个 Demo。员工权限设计的模型图，以及对应关系。

<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2020/03/image-01f94ef6.png?imageView2/2/interlace/1/format/jpg">
关系图，以及实体设计。
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2020/03/image-868c545d.png?imageView2/2/interlace/1/format/jpg">
表设计
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2020/03/image-80a45e7b.png?imageView2/2/interlace/1/format/jpg">
在我们平常的权限系统中，想完全遵循 <a href="https://ld246.com/forward?goto=http%3%2F%2Fwww.sojson.com%2Ftag_rbac.html" title="RBAC" target="_blank" rel="nofollow ugc"RBAC 模型是很难的，因为难免系统业务上有一些差异化的业务考量，所以在设计之初，不要理想，太追求严格的 RBAC 模型设，因为这样会使得你的系统处处鸡肋，无法拓展。
所以在这里要说明一下， RBAC 是一种模型，是一种思想，是一种核心思想，但是就思想而言，不是要你完全参照，而是你在这个基之上，融入你自己的思想，赋予你的业务之上，达到适用你的业务。所以要批评一下那些说：“<cod>RBAC</code> 模型是垃圾，按照它思路去执行，结果无法拓展。”之类话语的人。那是你自己不变通。
言归正传。
背景需求：
需要在 <code>“权限” =>“角色” =>“用户” </code> 之间，在赋予一个特殊的角“客服”，这个需求比较常见，我一个用户想把我的权限分配到“客服”角色上，然后由几个“客服去操作对应的业务流程。比如我们的天猫，淘宝商家后天就是如此，当店铺开到一定的规模，那么就有分工。
A 客服：负责打单填写发货单。
B~E 客服：负责每天对我们说“亲，您好。祝亲生活愉快！”，也就是和我们沟通交流的客服。/p>
F~H：负责售后。
... ...
那么这些客服也是归属到这个商家下面去。而且每个商家可能都有类似的客服，分工完全靠商家己去分配管理。
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2020/03/image-c6e7df11.png?imageView2/2/interlace/1/format/jpg">
这样的系统，融合我们的权限控制，关键要看“客服”用户的添加是在哪添加，如果是由客服直添加，不走我们的统一注册流程，那建议不要融合到上面这一套 权限、角色、用户之间去，而是给用再多一个绑定，把多个用户绑定到客服下，并且给客服赋予对应的权限。
<h3 id="1-权限赋予-">1、权限赋予：</h3>
权限赋予是把当前用户的权限拉出来，然后分配的客服可以小于等于当前用户的权限。
<h3 id="2-权限加载-">2、权限加载：</h3>
正常的加载权限，当用户登录后，并且第一次使用权限判断的时候， Shiro 会去加载权限。
<h3 id="3-权限判断-">3、权限判断：</h3>
走正常用户权限判断，但是数据操作需要判断是不是当前归属的用户的数据，其实这个是属于业层，就算你不是客服，也是需要判断。
<h3 id="4-禁用---启用-">4、禁用 | 启用：</h3>
禁用启用，也是正常的用户流程，添加到禁用列表里，如果被禁用，就无法操作任何内容。
总之：不要让框框架架来限制你的业务，也不要让你的业务局限于框框架架。但是也不推荐你去

动框框架架，而是基于框框架架做业务封装。</p>