长春站实时榜说明文档

winguse

引言

2012 年 10 月 3 号回到长春，开始榜的开发，中途净月校区大停电，停工两天。其余时间基本是有空就写，13 号热身赛发现若干 Bug，14 号早上彻底完工，现场过程中一行代码没修改，大屏幕显示器鼠标都没动过，全自动执行到比赛结束。

外网服务器被拒绝服务攻击，造成榜无法同步，深表歉意。

版权声明

系统结构

<table>
<tr><td colspan="5">Live Board</td></tr>
<tr><td colspan="4">board.css (CSS3 + div)</td><td rowspan="2">system_admin.jsp</td></tr>
<tr><td colspan="4">board.js</td></tr>
<tr><td colspan="2">AJAX</td><td colspan="2">index.jsp</td><td>AJAX</td></tr>
<tr><td colspan="2" rowspan="3">GetStatus
StatusCacheUnion
DBStatusFetcher</td><td rowspan="3">ClearPedding</td><td rowspan="3">ProblemCache</td><td rowspan="3">TeamCache</td><td rowspan="3">SystemAdmin</td></tr>
<tr><td colspan="5">Board.java</td></tr>
<tr><td colspan="5">MySQL Database</td></tr>
<tr><td colspan="5">RunsListener</td></tr>
<tr><td colspan="5">PC^2 API</td></tr>
<tr><td colspan="5">PC^2</td></tr>
</table>

系统要求

PC^{2}9.2, JDK1.7, MySQL5.2（UTF-8 编码，记得修改）, Tomcat7, Eclipse for J2EE，现代浏览器（Chrome 首选）

Windows 和 Linux 通用。为了截图方便，我直接在 Windows 下写文档的。

队伍数据

如果你同时使用我们的赛事系统，ContestService/BoardInfo 这个连接即可获得相关的数据。

否则，请参考 http://acm.nenu.edu.cn/ContestService/BoardInfo 的格式设置你相关的数据内容。

注意，文件要保存到 js/boardinfo.js 里面。

其中那个 json 的主键是 PC2 里面队伍名的 DisplayName，请注意对应修改。我们的方案是手工修改。因为本身的队伍名是差不多对应好的，你打开榜的时候，可以在 Chrome 的开发者工具里面的 Console 里面看到那些没有对应的队伍的警告，稍微修改一下就好，不用很久。

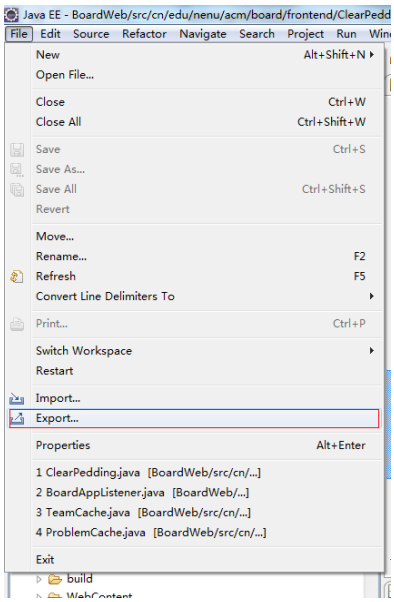
如果你使用我们的赛事系统，并且是用导入 ICPC 数据的方法，又使用的是 ICPC 提供的 PC^2 数据文件，那么几乎没有什么地方要修改的。除了酱油队。

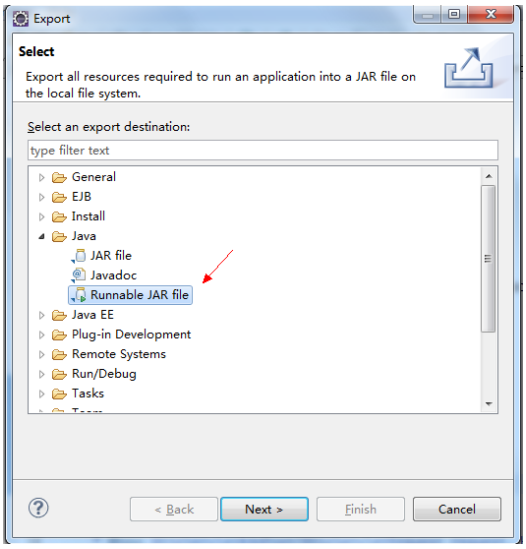
工作步骤

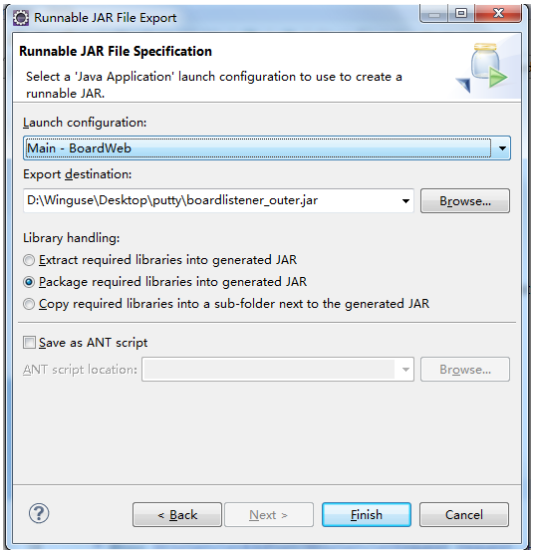
修改 Board.java 配置数据连接池参数，密码之类。

后端 API

1. 修改 RunsListener.java 修改 scoreboard 帐号密码。
2. 导出 API Listener







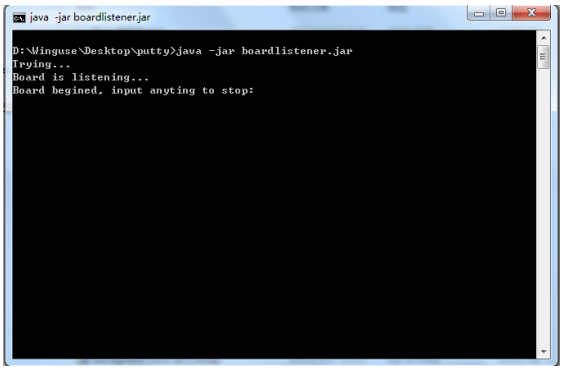
3. 运行 API

   准备正确的 pc2.ini 放在 API jar 包的目录下。

   打开命令窗口，切换到 API 目录。

   运行 java -jar XXX.jar

   运行正确截图如下：



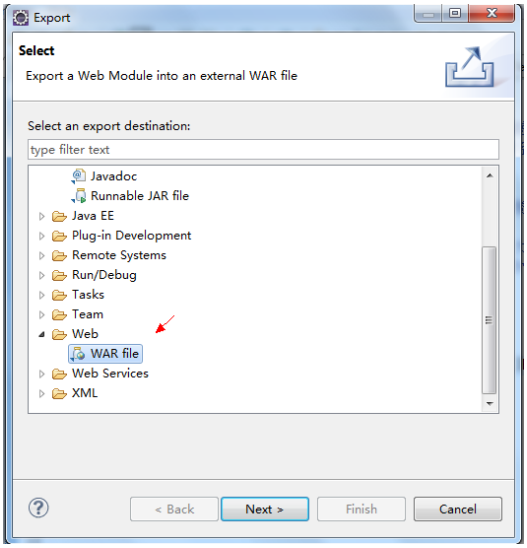
   有新 status 的状态改变，会显示成日志。

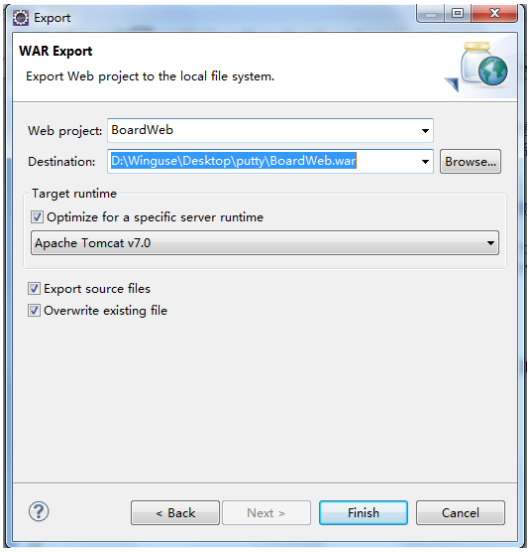
   要终止程序，随便输入点东西即可。

注意：每次运行 API 都会重置数据库，同时，存在这样的情况，你正在启动 API，却有新的提交，这些新的是不会被更新的——这个没有任何办法解决（看队伍提交和你启动的速度啦），所以尽可能不要动 API，并且保证 API 先运行，比赛才开始。你可以试试先加监听，后同步数据的办法，不过真心写得比较麻烦。

前端

1. 点击菜单，选择导出，这次导出 war 文件：



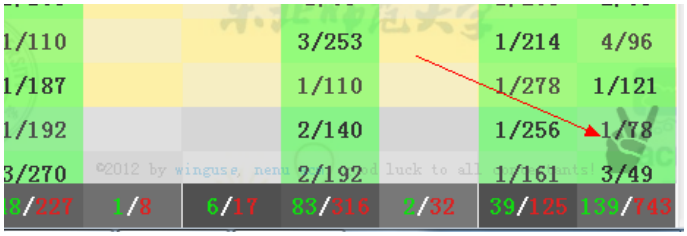


2. 将 war 文件上传或复制到服务器 Tomcat App 的相关目录，默认配置会自动解压。
3. 直接访问 http://Tomcat 服务器/BoardWeb 即可。

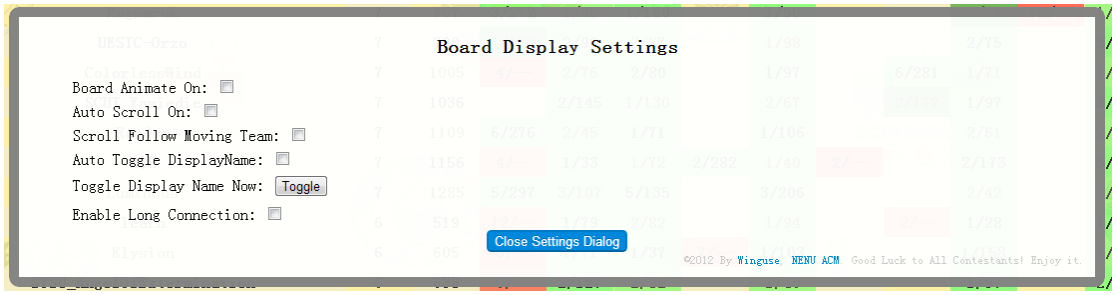
基本参数说明

显示调整

1. 点击队伍名、或学校名，将显示队伍的详细信息。
2. 点击榜的右下角，有个 **胜利的手势** 可以唤出设置对话框。（PS：我是想找个兔子的图标的，可好像 UTF-8 字符集里面好像没有）



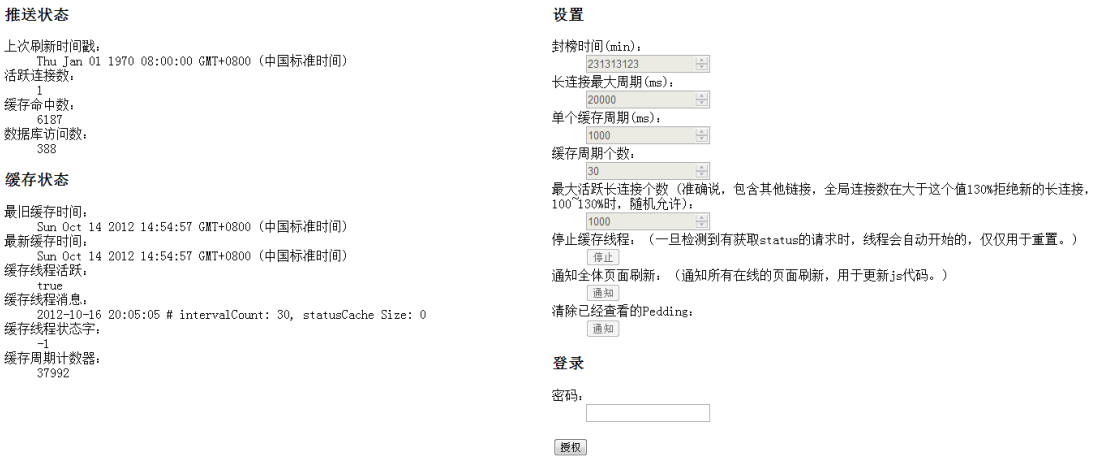
3. 出现如下画面：



依次说明如下：
分榜动画，队伍在榜单位置发生变化，会飘动。
自动滚动，当空闲的时候，自动上下滚动榜，让所有队伍都有机会显示。（FireFox 有 bug，还没修复，你自己来吧）
滚动条跟随，当队伍飘动时，滚动页面跟随队伍的飘动。
自动切换显示名称，自动在队伍名和学校名直接切换。
立刻切换显示名称。
使用长连接，长连接会跟服务器保持一个 20 秒（默认）的连接，能够几乎实时的获得所有情况的变化（理论上最多 3 秒），不选上默认 60s 一次更新。

系统参数

打开 system_admin.jsp，可以看到这个：



详细就不用说了，都已经写到页面里面了。页面 5 秒钟已更新，如果已经登录，也是 5 秒钟一个同步设置的参数。

长连接周期 20 秒，不建议再短，也不建议长，否则服务器容易过载。

单个缓存周期不要短于 500ms，每个周期都会查询数据库的，但是太长就没有实时效果了。

缓存周期个数乘以单个缓存周期不要超过 1 分钟，否则有可能造成网络流量消耗过快，因为榜刚刚运行的时候，缓存的内容在这个时间内是很多的，11、12 号现场测试的时候，180 台机器 1900 个 runs 的情况下，开始几分钟就一下子跑完全部带宽（90k*180，基本就能跑满 100Mbps 的网络了）——当然是个极端情况，综合分析，仅仅是开场的时候，缓存堆积得太厉害了，实际上长连接是为了节约带宽设置的。

最大活跃连接数不要超过 1000，具体上限没估测过，现场 180 个队伍全部打开长连接都没什么压力，不过 Tomcat 有个并发线程数限制的，注意去调整一下参数。一般来说，除了大屏幕并没有其他地方要求那么同步的，我默认就没打开，采取 60 秒短连接刷新。长春站现场赛外网实时访问量不超过 400 同时在线，但是也有个别同学打开了长连接，二三十吧，无压力。

通知全体页面刷新，如果是长连接的话，全部页面会几乎同时刷新的，180 个队伍，也能瞬间跑慢 100Mbps 带宽的，可以用来测试压力。

登录的密码在 Login.java 里面改。

外网同步

建议使用 SSH 隧道实现，将远程 MySQL 端口映射到本机端口。

我当时用的指令是：

ssh XXX@acm.nenu.edu.cn -L 3307:localhost:3306

导出两份 API jar，注意，数据库端口变成 3307 了，另外呢，PC^2 不允许一个账户同时登录的，所以要改成另外一个 scoreboard 账户。

内外网接触的服务器，记得防火墙全开。虽然我们现场赛外网服务器 DDOS 挂掉了，但是内网一点事都没有。

缺陷说明及后面维护建议

IE9 似乎会假死。IE8 是最低兼容的浏览器。

榜单有颁奖功能，但是模式不是特别好。我们最终用用的时候，仅仅是干脆从数据库里面标记一下（如果你有印象，你会看到当天我运行了一个 SQL 语句，然后立刻切换回榜的画面的），让榜在最后 34 名里面播放了一次动画——其实真心说，是那天限于时间。实际上，我们原来想法是，内外网同步的，可惜 acm.nenu.edu.cn 那个时候被拒绝服务了，而场内颁奖又出了点问题，没空弄。

管理员登录后，如果存在 pedding 的队伍，点击该队伍，就会清楚第一个能够让这个队伍升上去的提交，而且还是同步的。清楚的粒度可以是一个题目，详细见代码。由于最开始设计并无太多考虑颁奖，这个功能有点太耗时间，所以也是没使用的原因。

建议开发相关统计功能，例如说，显示整场比赛的 status，提交次数和时段，队伍排名变化等。——其实我是计划中的，不过只有半个月，真心无能力写那么多。

建议仔细**回归测试**。

鸣谢

感谢周治国老师信任，将这个任务交给我完成，并提供各种修改意见。感谢杨贵福老师、李辉老师信任我在各种系统运行问题上提供方便，以及技术上的建议。感谢赛场内外各位老师协调，包括供电、网络、设备等方便的支持。谢谢各位 NENU ACM 队员、赛场志愿者帮忙完成各项测试工作。没有上述各位的帮助，这项工作无法在如此短期内完成。

我这一辈子估计就参与这样一次比赛的组织，希望大家喜欢我的工作。感谢 ACM/ICPC 给我这样一个舞台。

开发前查阅了复旦大学黄磊学长写的 API，虽然我开始也已经写了一份，并且大家获取方式都很相似，但是我还是领教和学习了很多。同时感谢复旦大学，也是我高中校友朱健维（纳米）同学。

感谢舒啸（兔子），在大一时让我从颓废中走出来，走到了现在。