

曾经沧海难为水

客

过是外星人家中的卫生死角

计算机学校排名



云服务器吧



访问：11100次

积分：422

等级：BLOG > 2

排名：千里之外

原创：32篇 转载：1篇

译文：0篇 评论：10条

博客专栏



浅读
java.util.Map及其实现类
文章：5篇
阅读：693

文章分类

总结 (32)

Java (21)

流媒体 (4)

代码方案 (5)

前端技术 (5)

Linux (2)

文章存档

2017年08月 (2)

2017年07月 (1)

2017年06月 (7)

2017年05月 (2)

2017年04月 (2)

展开

阅读排行

java实现svn，svnkit框架 (1969)

分享一下最近对直播的研 (1066)

nw.js 0.19打包exe (951)

自己做个java sqlite连接 (933)

Red5+SpringMVC整合(F (710)

WebStorm下载及破解 (617)

centos下安装openmeetir (513)

目录视图

摘要视图

RSS 订阅

异步赠书：Kotlin领书10本好书

免费直播：AI时代，机器学习如何入门？

程序员8月书讯

项目管理+

作，开发更流畅

java实现svn，svnkit框架的简单应用

标签：java svn svnkit

2016-11-17 14:16 1976人阅读 评论(4)

分类：

Java (20) 总结 (31) 代码方案 (4)

版权声明：原创文章，禁止转载，违者必究

SvnKit

地址：<https://svnkit.com/download.php>

项目需要做了个简单的demo，可以进行基础操作。demo可以运行，但是更多高级更多实现需要自己在扩展。因为这个项目已经进入Thread.sleep :(，开展新项目了 :(

功能

1.实现了几个基础操作

2.提供了日志操作

项目结构



```
src
├── com
│   ├── svn
│   │   ├── conf
│   │   │   ├── ErrorVal.java
│   │   │   └── SvnConfig.java
│   │   ├── core
│   │   ├── factory
│   │   │   └── DemoSvn.java
│   │   ├── impl
│   │   │   ├── service
│   │   │   │   ├── SvnCommonImpl.java
│   │   │   │   ├── SvnServiceImpl.java
│   │   │   │   └── SvnServicePoolImpl.java
│   │   │   └── SvnBaseImpl.java
│   │   ├── inf
│   │   │   ├── service
│   │   │   │   ├── ISvnCommon.java
│   │   │   │   ├── ISvnDbLog.java
│   │   │   │   └── ISvnService.java
│   │   │   └── ISvn.java
│   │   ├── model
│   │   │   ├── SvnAccountPojo.java
│   │   │   ├── SvnLinkPojo.java
│   │   │   └── SvnRepoPojo.java
│   │   ├── tools
│   │   │   ├── ConsoleLog.java
│   │   │   └── StringOutputStream.java
│   │   └── example.java
│   └── example
```

首先我们先创建3个对象来为后面服务

```
1 package com.svn.model;
2 /**
3  * Svn账号信息
4  *
```

游戏培训学校





```

5  * @author Allen
6  * @date 2016年8月8日
7  */
8  public class SvnAccountPojo implements java.io.Serializable {
9      /**
10       *
11       */
12     private static final long serialVersionUID = 1L;
13     private String svnAccount;// svn账号
14     private String svnPassword;// svn密码
15
16     protected SvnAccountPojo() {
17         // TODO Auto-generated constructor stub
18     }
19
20     public SvnAccountPojo(String svnAccount, String svnPassword) {
21         super();
22         this.svnAccount = svnAccount;
23         this.svnPassword = svnPassword;
24     }
25
26     public String getSvnAccount() {
27         return svnAccount;
28     }
29
30     public void setSvnAccount(String svnAccount) {
31         this.svnAccount = svnAccount;
32     }
33
34     public String getSvnPassword() {
35         return svnPassword;
36     }
37
38     public void setSvnPassword(String svnPassword) {
39         this.svnPassword = svnPassword;
40     }
41
42 }

```

```

1  package com.svn.model;
2
3  /**
4   * Svn链接状态信息
5   *
6   * @author Allen
7   * @date 2016年8月8日
8   */
9  public class SvnLinkPojo extends SvnAccountPojo {
10
11     /**
12      *
13      */
14     private static final long serialVersionUID = 1L;
15     private String repoPath;// 库链接路径
16
17     public SvnLinkPojo(String repoPath, String svnAccount, String svnPassword) {
18         super(svnAccount, svnPassword);
19         this.repoPath = repoPath;
20     }
21
22     public SvnLinkPojo(String svnAccount, String svnPassword) {
23         super(svnAccount, svnPassword);
24     }
25
26     public String getRepoPath() {
27         return repoPath;
28     }
29
30     public void setRepoPath(String repoPath) {
31         this.repoPath = repoPath;
32     }

```

关闭





```

33
34 }

1  package com.svn.model;
2
3  import java.util.Date;
4
5  /**
6   * SVN资源库对象
7   *
8   * @author Allen
9   * @date 2016年8月8日
10  */
11 public class SvnRepoPojo implements java.io.Serializable {
12     /**
13      *
14      */
15     private static final long serialVersionUID = 1L;
16     private String commitMessage; // 提交信息
17     private Date date; // 提交日期
18     private String kind; // 提交方式 dir目录 file文件 none空 unknown 未知
19     private String name; // 目录名
20     private String repositoryRoot; // 资源库路径
21     private long revision; // 提交的svn版本号
22     private long size; // 提交的文件数
23     private String url; // 变更的目录地址
24     private String author; // 作者
25     private String state; // 状态
26
27     public String getCommitMessage() {
28         return commitMessage;
29     }
30
31     public void setCommitMessage(String commitMessage) {
32         this.commitMessage = commitMessage;
33     }
34
35     public Date getDate() {
36         return date;
37     }
38
39     public void setDate(Date date) {
40         this.date = date;
41     }
42
43     public String getKind() {
44         return kind;
45     }
46
47     public void setKind(String kind) {
48         this.kind = kind;
49     }
50
51     public String getName() {
52         return name;
53     }
54
55     public void setName(String name) {
56         this.name = name;
57     }
58
59     public String getRepositoryRoot() {
60         return repositoryRoot;
61     }
62
63     public void setRepositoryRoot(String repositoryRoot) {
64         this.repositoryRoot = repositoryRoot;
65     }
66
67     public long getRevision() {
68         return revision;

```

关闭





```
69     }
70
71     public void setRevision(long revision) {
72         this.revision = revision;
73     }
74
75     public long getSize() {
76         return size;
77     }
78
79     public void setSize(long size) {
80         this.size = size;
81     }
82
83     public String getUrl() {
84         return url;
85     }
86
87     public void setUrl(String url) {
88         this.url = url;
89     }
90
91     public String getAuthor() {
92         return author;
93     }
94
95     public void setAuthor(String author) {
96         this.author = author;
97     }
98
99     public String getState() {
100         return state;
101     }
102
103     public void setState(String state) {
104         this.state = state;
105     }
106
107 }
```

创建svn基础服务接口

```
1  package com.svn.inf.service;
2
3  /**
4   * svn主服务创建
5   *
6   * @author Allen
7   * @date 2016年8月8日
8   */
9  public interface ISvnService {
10
11      /**
12       * 创建SNV版本库服务
13       *
14       * @author Allen
15       * @date 2016年8月11日
16       */
17      public void createSVNRepository();
18
19      /**
20       * 关闭版本库容器, 便于刷新重连等
21       *
22       * @author Allen
23       * @date 2016年8月11日
24       */
25      public void closeRepo();
26
27      /**
```

关闭





```
28      * 创建svn客户操作服务
29      *
30      * @author Allen
31      * @date 2016年8月11日
32      */
33      public void createSVNClientManager();
34  }
```

建立主功能接口

```
1  package com.svn.inf;
2
3  import java.io.File;
4  import java.util.List;
5
6  import com.svn.inf.service.ISvnDbLog;
7  import com.svn.inf.service.ISvnService;
8  import com.svn.model.SvnRepoPojo;
9
10 /*
11  * svn操作大全
12  *
13  * @author Allen
14  * @date 2016年8月8日
15  */
16 public interface ISvn extends ISvnService {
17
18     /**
19      * 获取目标路径下版本库数据信息
20      *
21      * @param openPath
22      *          需要查看的版本库路径
23      * @return 版本库列表 {@link SvnRepoPojo}
24      * @author Allen
25      * @date 2016年8月11日
26      */
27     public List<SvnRepoPojo> getRepoCatalog(String openPath);
28
29     /**
30      * 检出到目录
31      *
32      * @param checkUrl
33      *          检出目标URL
34      * @param savePath
35      *          检出到本地路径
36      * @return true|false
37      * @author Allen
38      * @date 2016年8月11日
39      */
40     public boolean checkOut(String checkUrl, String savePath);
41
42     /**
43      * 添加到版本库
44      *
45      * @see 先添加文件夹再添加文件
46      * @param paths
47      *          提交文件路径
48      * @param message
49      *          提交信息
50      * @param uLocks
51      *          是否解锁
52      * @param isvnLog
53      *          数据持久化接口 {@link ISvnDbLog}
54      * @return true|false
55      * @author Allen
56      * @date 2016年8月11日
57      */
58     public <T> boolean add(String[] paths, String mes
59 }
```



关闭





```

60  /**
61   * 提交到版本库(所有写操作已在内部调用过COMMIT, 自行调用则需要手动同步到DbLog)
62   *
63   * @param files
64   *         提交的文件路径
65   * @param message
66   *         提交信息
67   * @param uLocks
68   *         是否解锁
69   * @return 返回提交后版本号-1为提交失败
70   * @author Allen
71   * @date 2016年8月11日
72   */
73  public Long commit(File[] files, String message, boolean uLocks);
74
75  /**
76   * 删除到版本库
77   *
78   * @see 先删除文件再删除文件夹
79   * @param paths
80   *         提交文件路径
81   * @param localDelete
82   *         <ul>
83   *         <li>如果是true则在本地也删除此文件, false则只删除版本库中的此文件</li>
84   *         <li>删除实体文件时要注意</li>
85   *         <li>删除文件夹时其目录下所有内容都要提交到<b>参数paths</b>中, 否则无法删除
86   *         </ul>
87   * @param message
88   *         提交内容解释
89   * @param uLock
90   *         是否解锁
91   * @param isvnLog
92   *         数据持久化接口 {@link ISvnDbLog}
93   * @return true|false
94   * @author Allen
95   * @date 2016年8月11日
96   */
97  public <T> boolean delete(String[] paths, boolean localDelete, String message, boolean
98
99  /**
100   * 更新到版本库
101   *
102   * @param path
103   *         要更新的文件目录
104   * @param message
105   *         提交内容解释
106   * @param uLock
107   *         是否解锁
108   * @param isvnLog
109   *         数据持久化接口 {@link ISvnDbLog}
110   * @return true|false
111   * @author Allen
112   * @date 2016年8月11日
113   */
114  public <T> boolean update(String path, String message, boolean uLock, ISvnDbLog<? supe
115
116  /**
117   * 比对目录下内容信息
118   *
119   * @see 返回delete,update文件列表
120   * @param file
121   *         待比对的目标文件路径
122   * @return 返回有差异的文件路径否则为null
123   * @author Allen
124   * @date 2016年8月11日
125   */
126  public List<String> diffPath(File file);
127
128  /**
129   * 清理目录
130   */

```

关闭





```
131      * @param file
132      *          待清理目录
133      * @return true|false
134      * @author Allen
135      * @date 2016年8月17日
136      */
137      public boolean cleanUp(File file);
138
139      public boolean doLock();
140
141      public boolean unLock();
142  }
```

当然我们还有通用的工具接口

```
1  package com.svn.inf.service;
2
3  import java.io.File;
4  import java.util.List;
5
6  /**
7   * Svn功能组件
8   *
9   * @author Allen
10   * @date 2016年8月12日
11   *
12   */
13  public interface ISvnCommon {
14      /**
15       * 组装file[]分离文件与文件夹
16       *
17       * @param files
18       *          待重组的文件列表
19       * @return 文件容器组</br>index:0 文件夹</br> index:1 文件
20       * @author Allen
21       * @date 2016年8月11日
22       */
23      public List<List<File>> bindFile(File[] files);
24
25      /**
26       * 排序</br> 父级 ->子级
27       *
28       * @param files
29       *          待排序文件数组
30       * @return 排序后文件数组
31       * @author Allen
32       * @date 2016年8月11日
33       */
34      public File[] sortF_S(File[] files);
35
36      /**
37       * 排序</br> 子级 ->父级
38       *
39       * @param files
40       *          待排序文件数组
41       * @return 排序后文件数组
42       * @author Allen
43       * @date 2016年8月11日
44       */
45      public File[] sortS_F(File[] files);
46
47      /**
48       * 检查文件路径信息并组装到文件容器
49       *
50       * @param paths
51       *          待组装文件路径
52       * @return 文件数组
53       * @throws Exception
54       *          文件路径中有寻找不到的地址
```

关闭





```
55      * @author Allen
56      * @date 2016年8月11日
57      */
58      public File[] checkFilePaths(String[] paths) throws Exception;
59  }
```

为了满足上级要求我们提供了db接口

```
1  package com.svn.inf.service;
2
3  import java.io.File;
4  import java.util.Date;
5  import java.util.List;
6
7  import com.svn.conf.SvnConfig;
8
9  /**
10   * 记录svn操作人
11   *
12   * @author Allen
13   * @date 2016年8月8日
14   */
15  public interface ISvnDbLog<T> {
16      /**
17       * 添加日志
18       *
19       * @param name
20       *      操作人账号
21       * @param dbType
22       *      数据类型{@link SvnConfig}
23       * @param versionId
24       *      版本号
25       * @param files
26       *      操作的文件组
27       * @return true|false
28       * @author Allen
29       * @date 2016年8月11日
30       */
31      public boolean addLog(String name, SvnConfig dbType, long versionId, File[] files);
32
33      /**
34       * 获取日志
35       *
36       * @param name
37       *      操作人账号
38       * @param startTime
39       *      日志开始时间
40       * @param endTime
41       *      日志结束时间
42       * @return T 类型列表
43       * @author Allen
44       * @date 2016年8月11日
45       */
46      public List<? super T> getLog(String name, Date startTime, Date endTime);
47  }
```

到上面为止我们的接口全部做好。这是一个Java项目是希望做了一套基础的实现，按照这个思路根据不同需求可以慢慢拆

工具类

其实这里自己写了个简单的错误匹配，主要是针对svnkit的er等开源日志。不过毕竟是demo么：)

```
1  package com.svn.tools;
2
```





```

3  import java.io.File;
4  import java.text.SimpleDateFormat;
5  import java.util.Arrays;
6  import java.util.Date;
7
8  /**
9   * 控制台输出log信息
10  *
11  * @author Allen
12  * @date 2016年8月8日
13  */
14  public class ConsoleLog {
15      protected boolean log = false;
16
17      /**
18       * 错误编码
19       */
20      private String[] errorCode = { "E155010", "E200005", "E204899", "E200009" };
21
22      /**
23       * 编码解释
24       */
25      private String[] errorInfo = { "找不到文件或文件夹, 其所在目录未被纳入版本控制的", "文
26
27      /**
28       * 输出log到console
29       *
30       * @param msg
31       *         输出信息
32       */
33      protected void log(String msg, String... content) {
34          if (log) {
35              StackTraceElement stack[] = (new Throwable()).getStackTrace();
36              String time = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss").format(new Date());
37              System.out.println(new StringBuffer("[").append(time).append("]").append(stack
38                  .append(stack[1].getLineNumber()));
39              System.out.println(new StringBuffer("[").append(time).append("]").append(msg))
40          }
41      }
42
43      protected void log(Exception e) {
44          StringBuffer sbf = new StringBuffer("【SVN-ERROR】");
45          for (int i = 0; i < errorCode.length; i++) {
46              if (e.getMessage().indexOf(errorCode[i]) != -1)
47                  this.log(sbf.append(errorInfo[i]).toString());
48          }
49      }
50  }

```

这里主要是输出文本比定时差异信息

说实话版本冲突这里我svn是可以对列表进行更新或写操作的时候, 批量返回错误信息, 而svn提供的cmd入口及svnkit对其cmd命令的实现, 都是提供的单条问题抛出, 譬如我update一个文件目录, 在第3, 第5个文件会出现冲突, 则接口在第三个文件直接就exception了, 无法往下进行了。当然你可以递归从第四条再更新执行, 最后把结果合并显示:

```

1  package com.svn.tools;
2
3  import java.io.IOException;
4  import java.io.OutputStream;
5  import java.util.List;
6
7  /**
8   * Output输出到字符串
9   *
10  * @author Allen
11  * @date 2016年8月11日
12  *
13  */
14  public final class StringOutputSteam extends OutputSt
15

```

关闭





```

16     public List<String> s;
17
18     public StringOutputStream(List<String> s) {
19         this.s = s;
20     }
21
22     @Override
23     public void write(int b) throws IOException {
24     }
25
26     @Override
27     public void write(byte b[], int off, int len) throws IOException {
28         String temp = new String(b);
29         String[] temps = temp.split("\\r\\n");
30         //System.out.println(temp);
31         boolean flag = false;
32         for (int i = 0; i < temps.length; i++) {
33             // 根据svn命令返回Index: xxxx(path)获取所有包含delete update的path并打印
34             if (temps[i].split(" ")[0].equalsIgnoreCase("Index:")) {
35                 String p = temps[i].substring(7, temps[i].length());
36                 if (s.indexOf(p) == -1) {
37                     s.add(p);
38                     flag = true;
39                     continue;
40                 }
41             }
42             // 以Index: 为开始点来判定svn diff返回的Log信息
43             // LogFormat:
44             // Index: path
45             // =====
46             // xxxxxxx
47             // Index: path
48             // =====
49             // xxxxxxx
50             if (flag) {
51                 String state = "";
52                 for (int j = i; j < temps.length; j++) {
53                     if (temps[j].split(" ")[0].equalsIgnoreCase("Index:"))
54                         break;
55                     if (temps[j].split(" ")[0].equalsIgnoreCase("----"))
56                         state = "冲突";
57                     else if (state.equals(""))
58                         state = "删除";
59                 }
60                 s.add(state);
61                 flag = false;
62             }
63         }
64     }
65 }

```

这里是自定义的错误信息

```

1 package com.svn.conf;
2
3 /**
4  * 错误信息库
5  *
6  * @author Allen
7  * @date 2016年8月8日
8  *
9  */
10 public final class ErrorVal {
11
12     public final static String SVNRepository_is_null
13     public final static String SVNClientManager = "";
14     public final static String SVNRepository_is_having
15     public final static String ISVNAuthenticationManager
16     public final static String Path_no_having = "目标

```

关闭





计算机学校排名



云服务器吧



```

17     public final static String Url_no_having = "目标库路径 不存在";
18     public final static String File_not_exist_repo = "文件所在目录没有被svn版本控制";
19     public final static String SvnConfig_is_null = "错误的配置";
20     public final static String Commit_error = "提交错误[==请确认文件是否已有其他版本操作,i
21     public final static String AddDbLog_error = "添加到数据日志错误";
22     public final static String File_Repo_no_having = "文件不存在版本库中";
23     public final static String Repo_Status_error = "版本状态错误";
24     public final static String Update_no_change = "没有变化的更新";
25     public final static String File_is_not_directory = "File不是一个目录";
26 }

```

这里不仅为小工厂提供了选项，还提供了对抽象db日志的数据设置

```

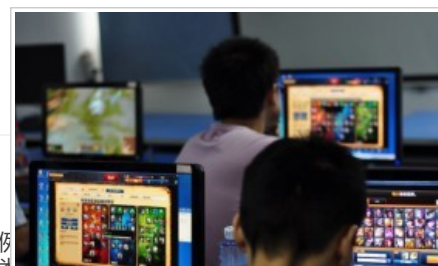
1  package com.svn.conf;
2
3  /**
4   * 系统参数设置
5   *
6   * @see 外部通过<b>SvnConfig.xx</b>来一个<b>SvnConfig</b>类型传入不同的配置
7   * @see 外部通过<b>SvnConfig.xx.get</b>来以<b>String</b>类型得到不同的配置
8   * @author Allen
9   * @date 2016年8月8日
10  *
11  */
12  public class SvnConfig {
13      /**
14       * 选择非日志模式
15       */
16      public static final SvnConfig normal = new SvnConfig("normal");
17      /**
18       * 选择console日志模式
19       */
20      public static final SvnConfig log = new SvnConfig("log");
21      /**
22       * 数据存储类型Add
23       */
24      public static final SvnConfig add = new SvnConfig("add");
25      /**
26       * 数据存储类型Update
27       */
28      public static final SvnConfig update = new SvnConfig("update");
29      /**
30       * 数据存储类型Delete
31       */
32      public static final SvnConfig delete = new SvnConfig("delete");
33      private String val;
34
35      public String getVal() {
36          return val;
37      }
38
39      private SvnConfig() {
40      }
41
42      private SvnConfig(String val) {
43          this.val = val;
44      }
45  }
46 }

```

关闭

实现类来了。。。拦不住的implements

提供了对通用工具接口的实现，为什么要那么复杂呢。因为你的文件夹目录从深层提交到高层，然后再提交其文件，以此类法建立文件的。



游戏培训学校





```

1 package com.svn.impl.service;
2
3 import java.io.File;
4 import java.util.ArrayList;
5 import java.util.Arrays;
6 import java.util.List;
7
8 import com.svn.conf.ErrorVal;
9 import com.svn.inf.service.ISvnCommon;
10 import com.svn.tools.ConsoleLog;
11
12 /**
13  * {@link ISvnCommon}
14  *
15  * @author Allen
16  * @date 2016年8月12日
17  *
18  */
19 public class SvnCommonImpl extends ConsoleLog implements ISvnCommon {
20
21     @Override
22     public List<List<File>> bindFile(File[] files) {
23         List<List<File>> fileList = new ArrayList<List<File>>();
24         fileList.add(new ArrayList<File>());
25         fileList.add(new ArrayList<File>());
26         for (File f : files) {
27             if (f.isDirectory())
28                 // 0是文件目录, 先提交文件目录让再提交文件
29                 fileList.get(0).add(f);
30             else
31                 fileList.get(1).add(f);
32         }
33         for (int i = 0; i < fileList.size(); i++) {
34             if (fileList.get(i).size() == 0)
35                 fileList.remove(i);
36         }
37         return fileList;
38     }
39
40     @Override
41     public File[] sortF_S(File[] files) {
42         Arrays.sort(files);
43         return files;
44     }
45
46     @Override
47     public File[] sortS_F(File[] files) {
48         files = sortF_S(files);
49         File[] f = new File[files.length];
50         for (int i = 0; i < files.length; i++) {
51             f[i] = files[files.length - (i+1)];
52         }
53         return f;
54     }
55
56     @Override
57     public File[] checkFilePaths(String[] paths) throws Exception {
58         if (paths == null || paths.length == 0)
59             throw new Exception(ErrorVal.Path_no_havi
60         File[] files = new File[paths.length];
61         for (int i = 0; i < paths.length; i++) {
62             if (!files[i] = new File(paths[i])).exists
63                 throw new Exception(ErrorVal.Path_no
64         }
65         return files;
66     }
67 }
68 }

```

关闭





服务接口实现类，快乐的建立库容器

因为整个项目保持开闭原则，让你用的你随便用，不让你用的你就老实点:)
一次创建整个实例都能在开闭原则下快乐的使用各种创建好的实例/各种服务类

```

1  package com.svn.impl.service;
2
3  import org.tmatesoft.svn.core.SVNException;
4  import org.tmatesoft.svn.core.SVNURL;
5  import org.tmatesoft.svn.core.auth.ISVNAuthenticationManager;
6  import org.tmatesoft.svn.core.io.SVNRepository;
7  import org.tmatesoft.svn.core.io.SVNRepositoryFactory;
8  import org.tmatesoft.svn.core.wc.SVNClientManager;
9  import org.tmatesoft.svn.core.wc.SVNWCUtil;
10
11 import com.svn.conf.ErrorVal;
12 import com.svn.inf.service.ISvnService;
13
14 /**
15  * {@link ISvnService}
16  *
17  * @author Allen
18  * @date 2016年8月8日
19  */
20 public class SvnServiceImpl extends SvnCommonImpl implements ISvnService {
21
22     protected String svnAccount; // svn账号
23     protected String svnPassword; // svn密码
24     protected String svnRepoPath; // svn版本库根目录
25     protected boolean logStatus = false; // 日志状态
26     protected SVNRepository repository = null; // 版本库服务
27     protected ISVNAuthenticationManager authManager; // 身份验证器
28     protected SVNClientManager clientManager; // svn客户操作服务
29
30     /**
31      *
32      * @param account
33      *      账号
34      * @param password
35      *      密码
36      * @param logStatus
37      *      是否开启日志状态(默认false)
38      * @param repoPath
39      *      svn库根目录
40      *
41      */
42     public SvnServiceImpl(String account, String password, boolean logStatus, String repoPath) {
43         this.svnAccount = account;
44         this.svnPassword = password;
45         this.svnRepoPath = repoPath;
46         super.log = logStatus;
47     }
48
49     @Override
50     public void createSVNRepository() {
51         try {
52             if (repository != null)
53                 throw new Exception(ErrorVal.SVNRepository_is_having);
54             // 创建库连接
55             SVNRepository repository = SVNRepositoryFactory.create(this.svnRepoPath);
56             super.log("创建版本库连接");
57             // 身份验证
58             this.authManager = SVNWCUtil.createDefaultAuthenticationManager(this.svnAccount, this.svnPassword);
59             super.log("创建身份验证");
60             // 创建身份验证管理器
61             repository.setAuthenticationManager(authManager);
62             this.repository = repository;
63             super.log("设置版本库身份验证");
64         } catch (SVNException e) {
65             super.log(e);
66             e.printStackTrace();
67         }
68     }
69 }

```

关闭





```

67         } catch (Exception e) {
68             e.printStackTrace();
69         }
70     }
71
72     @Override
73     public void closeRepo() {
74         if (repository == null)
75             try {
76                 throw new NullPointerException(ErrorVal.SVNRepository_is_nul
77             } catch (Exception e) {
78                 e.printStackTrace();
79             }
80         else {
81             repository.closeSession();
82             repository = null;
83             super.log("关闭版本库");
84         }
85     }
86
87     @Override
88     public void createSVNClientManager() {
89         if (authManager == null)
90             try {
91                 throw new NullPointerException(ErrorVal.ISVNAuthenticationManager_is_null)
92             } catch (Exception e) {
93                 e.printStackTrace();
94             }
95         clientManager = SVNClientManager.newInstance(SVNWUtil.createDefaultOptions(true),
96             super.log("创建svn客户操作服务");
97     }
98
99 }

```

其实这里本来还想写个池服务。。不过时间关系就没写。毕竟demo没有后续跟进计划了。以后有机会再补

```

1  package com.svn.impl.service;
2
3  import com.svn.impl.service.SvnCommonImpl;
4  import com.svn.inf.service.ISvnService;
5
6  /**
7   * {@link ISvnService} 池的实现
8   *
9   * @author Allen
10  * @date 2016年8月8日
11  */
12  public class SvnServicePoolImpl extends SvnCommonImpl implements ISvnService {
13
14      @Override
15      public void createSVNRepository() {
16          // TODO Auto-generated method stub
17      }
18
19
20      @Override
21      public void closeRepo() {
22          // TODO Auto-generated method stub
23      }
24
25
26      @Override
27      public void createSVNClientManager() {
28          // TODO Auto-generated method stub
29      }
30
31  }

```

关闭



这里..这里就是业务实现了，其实就是对svnkit进行了一个封装，让他更符合项目要求



```

1  package com.svn.impl;
2
3  import java.io.File;
4  import java.util.ArrayList;
5  import java.util.Collection;
6  import java.util.Iterator;
7  import java.util.List;
8
9  import org.tmatesoft.svn.core.SVNDepth;
10 import org.tmatesoft.svn.core.SVNDirEntry;
11 import org.tmatesoft.svn.core.SVNException;
12 import org.tmatesoft.svn.core.SVNURL;
13 import org.tmatesoft.svn.core.wc.SVNDiffClient;
14 import org.tmatesoft.svn.core.wc.SVNRevision;
15 import org.tmatesoft.svn.core.wc.SVNStatus;
16 import org.tmatesoft.svn.core.wc.SVNStatusType;
17 import org.tmatesoft.svn.core.wc.SVNUpdateClient;
18
19 import com.svn.conf.ErrorVal;
20 import com.svn.conf.SvnConfig;
21 import com.svn.impl.service.SvnServiceImpl;
22 import com.svn.inf.ISvn;
23 import com.svn.inf.service.ISvnDbLog;
24 import com.svn.inf.service.ISvnService;
25 import com.svn.model.SvnRepoPojo;
26 import com.svn.tools.StringOutputStream;
27
28 /**
29  *
30  * {@link ISvnService}
31  *
32  * @author Allen
33  * @date 2016年8月8日
34  *
35  */
36 public class SvnBaseImpl extends SvnServiceImpl implements ISvn {
37
38     public SvnBaseImpl(String account, String password, boolean logStatus, String repoPath) {
39         super(account, password, logStatus, repoPath);
40     }
41
42     @SuppressWarnings("unchecked")
43     @Override
44     public List<SvnRepoPojo> getRepoCatalog(String openPath) {
45         try {
46             if (repository == null)
47                 throw new Exception(ErrorVal.SVNRepository_is_null);
48             Collection<SVNDirEntry> entries = repository.getDir(openPath, -1, null, (Collection<SvnRepoPojo> svns = new ArrayList<SvnRepoPojo>());
49             Iterator<SVNDirEntry> it = entries.iterator();
50             while (it.hasNext()) {
51                 SVNDirEntry entry = it.next();
52                 SvnRepoPojo svn = new SvnRepoPojo();
53                 svn.setCommitMessage(entry.getCommitMessage());
54                 svn.setDate(entry.getDate());
55                 svn.setKind(entry.getKind().toString());
56                 svn.setName(entry.getName());
57                 svn.setRepositoryRoot(entry.getRepositoryRoot());
58                 svn.setRevision(entry.getRevision());
59                 svn.setSize(entry.getSize() / 1024);
60                 svn.setUrl(openPath.equals("") ? new SvnRepoPojo().getUrl() : openPath + "/" + entry.getName());
61                 svn.setAuthor(entry.getAuthor());
62                 svn.setState(svn.getKind() == "file" ? "file" : "dir");
63                 svns.add(svn);
64             }
65             super.log("获得版本库文件信息");
66             return svns;
67         } catch (SVNException e) {
68

```

关闭





计算机学校排名



云服务器吧



游戏培训学校

```

69         super.log(e);
70         e.printStackTrace();
71         return null;
72     } catch (Exception e) {
73         super.log(e);
74         e.printStackTrace();
75         return null;
76     }
77 }
78
79 @Override
80 public boolean checkOut(String checkUrl, String savePath) {
81     SVNUpdateClient updateClient = clientManager.getUpdateClient();
82     updateClient.setIgnoreExternals(false);
83     try {
84         if (savePath == null || savePath.trim().equals(""))
85             throw new Exception(ErrorVal.Path_no_having);
86         else if (checkUrl == null || checkUrl.trim().equals(""))
87             throw new Exception(ErrorVal.Url_no_having);
88         File save = new File(savePath);
89         if (!save.isDirectory())
90             save.mkdir();
91         updateClient.doCheckout(SVNURL.parseURIEncoded(checkUrl), save, SVNRevision.HI
92         super.log("检出版本库信息");
93         return true;
94     } catch (SVNException e) {
95         super.log(e);
96         e.printStackTrace();
97     } catch (Exception e) {
98         e.printStackTrace();
99     }
100     return false;
101 }
102
103 /**
104  * 此实现自动对Add后的数据进行提交
105  */
106 @Override
107 public <T> boolean add(String[] paths, String message, boolean uLocks, ISvnDbLog<? ext
108     try {
109         File[] files = checkFilePaths(paths);
110         files = sortF_S(files);
111         SVNStatus status;
112         List<File> targetList = new ArrayList<File>();
113         List<List<File>> fileList = bindFile(files);
114         for (int i = 0; i < fileList.size(); i++) {
115             for (File f : fileList.get(i)) {
116                 if ((status = clientManager.getStatusClient().doStatus(f, true, true))
117                     && status.getContentsStatus() != (SVNStatusType.STATUS_NONE))
118                     continue;
119                 else if (f.isFile()) {
120                     clientManager.getWCCClient().doAdd(f, true, false, true, SVNDepth.f
121                     targetList.add(f);
122                     super.log("添加文件到提交队列");
123                 } else if (f.isDirectory()) {
124                     // SVNDepth.empty 保证不递归文件夹下文件
125                     clientManager.getWCCClient().doAdd(f, false, false, false, SVNDepth
126                     targetList.add(f);
127                     super.log("添加文件夹到提交队列");
128                 }
129             }
130         }
131         long versionId = commit(targetList.toArray(), message, uLocks);
132         if (versionId == -1)
133             throw new Exception(ErrorVal.Commit_error);
134         if (!isvnLog.addLog(this.svnAccount, SvnLogType.ADD, versionId, message))
135             throw new Exception(ErrorVal.AddDbLog_error);
136         return true;
137     } catch (SVNException e) {
138         super.log(e);
139         e.printStackTrace();

```

关闭



```

140     } catch (Exception e) {
141         e.printStackTrace();
142     }
143     return false;
144 }
145
146 @Override
147 public Long commit(File[] files, String message, boolean uLocks) {
148     try {
149         if (files.length == 0) {
150             super.log("无效的提交信息");
151             return -1;
152         }
153         long versionId = clientManager.getCommitClient().doCommit(files, uLocks, message);
154         super.log("提交队列中预处理的提交操作 => 版本号: " + versionId);
155         return versionId;
156     } catch (SVNException e) {
157         super.log(e);
158         e.printStackTrace();
159     } catch (Exception e) {
160         e.printStackTrace();
161     }
162     return -1;
163 }
164
165 @Override
166 public <T> boolean delete(String[] paths, boolean localDelete, String message, boolean
167     try {
168         File[] files = checkFilePaths(paths);
169         files = sortS_F(files);
170         SVNStatus status = null;
171         {
172             List<File> targetList = new ArrayList<File>();
173             List<List<File>> fileList = bindFile(files);
174             for (int i = fileList.size() - 1; i >= 0; i--) {
175                 for (File f : fileList.get(i)) {
176                     if ((status = clientManager.getStatusClient().doStatus(f, true, true)
177                         throw new Exception(ErrorVal.File_Repo_no_having);
178                     else if (status.getContentsStatus() != SVNStatusType.STATUS_NORMAL
179                         throw new Exception(ErrorVal.Repo_Status_error + status.getCor
180                     else {
181                         clientManager.getWCCClient().doDelete(f, false, localDelete, false
182                         if (f.isFile())
183                             super.log("添加文件到删除队列");
184                         else
185                             super.log("添加文件夹到删除队列");
186                         targetList.add(f);
187                     }
188                 }
189             }
190             long versionId = commit(targetList.toArray(new File[targetList.size()]), message,
191             if (versionId == -1)
192                 throw new Exception(ErrorVal.Commit_error);
193             if (!svnLog.addLog(this.svnAccount, SvnConfig.delete, versionId, files))
194                 throw new Exception(ErrorVal.AddDbLog_error);
195         }
196         return true;
197     } catch (SVNException e) {
198         super.log(e);
199         e.printStackTrace();
200     } catch (Exception e) {
201         e.printStackTrace();
202     }
203     return false;
204 }
205
206 @Override
207 public <T> boolean update(String path, String message, boolean localUpdate, boolean
208     try {
209         File[] files = checkFilePaths(new String[] { path });
210         // diffPath(files);

```

关闭





```

211         long[] l = clientManager.getUpdateClient().doUpdate(files, SVNRevision.HEAD, 1);
212         super.log("更新文件到操作队列");
213         long versionId = l[0]; // commit(files, message, uLocks);
214         if (versionId == -1)
215             throw new Exception(ErrorVal.Update_no_change);
216         if (!svnLog.addLog(this.svnAccount, SvnConfig.update, versionId, files))
217             throw new Exception(ErrorVal.AddDbLog_error);
218     } catch (SVNException e) {
219         super.log(e);
220         e.printStackTrace();
221     } catch (Exception e) {
222         e.printStackTrace();
223     }
224     return false;
225 }
226
227 @Override
228 public List<String> diffPath(File file) {
229     try {
230         if (file == null || !file.exists())
231             throw new Exception(ErrorVal.Path_no_having);
232         // 获取SVNDiffClient
233         SVNDiffClient diffClient = clientManager.getDiffClient();
234         diffClient.setIgnoreExternals(false);
235         StringOutputStream os = new StringOutputStream(new ArrayList<String>());
236         diffClient.doDiff(new File[] { file }, SVNRevision.HEAD, SVNRevision.BASE, null);
237         super.log("比对库路径");
238         return os.s;
239     } catch (SVNException e) {
240         super.log(e);
241         e.printStackTrace();
242     } catch (Exception e) {
243         e.printStackTrace();
244     }
245     return null;
246 }
247
248
249 @Override
250 public boolean cleanUp(File file) {
251     try {
252         if (!file.exists())
253             throw new Exception(ErrorVal.Path_no_having);
254         else if (!file.isDirectory())
255             throw new Exception(ErrorVal.File_is_not_directory);
256         clientManager.getWCClient().doCleanup(file, false, false, false, true, true, 1);
257         super.log("清理完成");
258     } catch (SVNException e) {
259         super.log(e);
260         e.printStackTrace();
261     } catch (Exception e) {
262         e.printStackTrace();
263     }
264
265     return false;
266 }
267
268 @Override
269 public boolean doLock() {
270     // TODO Auto-generated method stub
271     return false;
272 }
273
274 @Override
275 public boolean unlock() {
276     // TODO Auto-generated method stub
277     return false;
278 }
279
280 }

```

关闭





计算机学校排名



云服务器吧



实现代码比较多。不要慌

仔细看看加断点看看
加锁解锁还有一些接口的高级应用并没有深入去写，因为毕竟是demo。上级觉得技术上是
可以搞定这个功能的，虽然因为其他原因项目阉割了 :(所以也没有后续进行深层开发
<https://svnkit.com/documentation.html> 这个是svnkit 文档，不过文档比svnkit.jar版本要老：
((

关于参数中

```
1 | ISvnDbLog<? super T> isvnLog
```

可以让使用者提供一个ISvnDbLog的实现，来进行日志保存，无论是db还是file
最后我们提供一个简单纺织小工厂

```
1 | package com.svn.factory;
2 |
3 | import com.svn.conf.ErrorVal;
4 | import com.svn.conf.SvnConfig;
5 | import com.svn.impl.SvnBaseImpl;
6 | import com.svn.inf.ISvn;
7 | import com.svn.model.SvnLinkPojo;
8 |
9 | /**
10 |  * DemoSvn主体
11 |  *
12 |  * @author Allen
13 |  * @date 2016年8月8日
14 |  */
15 | public final class DemoSvn {
16 |
17 |     SvnLinkPojo svnLink;
18 |
19 |     public SvnLinkPojo getSvnLink() {
20 |         return svnLink;
21 |     }
22 |
23 |     /**
24 |      * 私有构造
25 |      */
26 |     public DemoSvn() {
27 |     }
28 |
29 |     public DemoSvn(String svnAccount, String svnPassword, String repoPath) {
30 |         this.svnLink = new SvnLinkPojo(repoPath, svnAccount, svnPassword);
31 |     }
32 |
33 |     /**
34 |      * 获取SVN操作
35 |      *
36 |      * @param val
37 |      *      default 不设置日志状态 log 开启console日志状态
38 |      * @throws 没有操作匹配
39 |      * @return {@link ISvn}
40 |      */
41 |     public ISvn execute(SvnConfig val) throws Exception {
42 |         ISvn is = null;
43 |         if (val == null)
44 |             throw new Exception(ErrorVal.SvnConfig_is
45 |                 switch (val.getVal()) {
46 |                     case "normal":
47 |                         is = new SvnBaseImpl(svnLink.getSvnAccount,
48 |                             svnLink.getSvnPassword);
49 |                         break;
50 |                     case "log":
51 |                         is = new SvnBaseImpl(svnLink.getSvnAccount,
```

关闭

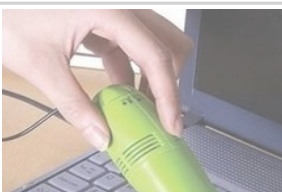


游戏培训学校





计算机学校排名



云服务器吧



```

51         break;
52     default:
53         throw new Exception(ErrorVal.SvnConfig_is_null);
54     }
55     return is;
56 }
57 }

```

来一个测试函数吧

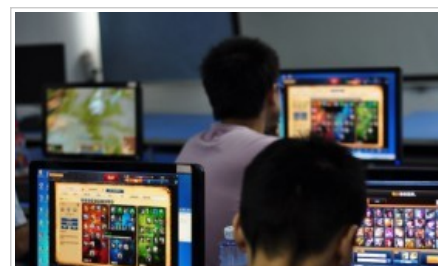
其实创建版本库，操作对象，初始化实例，close都可以做的更简洁，然后做到事务中理.. :)

```

1  package com.svn;
2
3  import java.io.File;
4  import java.util.Collections;
5  import java.util.Comparator;
6  import java.util.List;
7  import java.util.Date;
8
9  import com.svn.conf.SvnConfig;
10 import com.svn.factory.DemoSvn;
11 import com.svn.inf.ISvn;
12 import com.svn.inf.service.ISvnDbLog;
13 import com.svn.model.SvnRepoPojo;
14
15 /**
16  * 测试类
17  *
18  * @author Allen
19  * @date 2016年8月8日
20  */
21 public class example {
22     String account = "allen";
23     String password = "123456";
24     String path = "http://192.168.1.xx/svn/abc";
25     String targetHead = "e:/测试";
26     ISvn svn;
27
28     /**
29      * 样例
30      *
31      * @throws Exception
32      * @author Allen
33      * @date 2016年8月12日
34      */
35
36     private void testCore() throws Exception {
37         // 初始化实例
38         DemoSvn ts = new DemoSvn(account, password, path);
39         // 获得操作对象
40         this.svn = ts.execute(SvnConfig.log);
41         // 得到版本库信息
42         svn.createSVNRepository();
43         // 得到基础操作对象
44         svn.createSVNClientManager();
45         /** 测试--Start-- */
46         testGetRepo();
47         testCheckOut();
48         testAdd();
49         testDel();
50         testCleanUp();
51         testUpdate();
52         testDiff();
53         /** 测试 --End-- */
54         // 关闭库容器
55         svn.closeRepo();
56
57     }

```

关闭



游戏培训学校





```

58
59  /**
60   * 获得版本路径文件信息
61   *
62   * @author Allen
63   * @date 2016年8月12日
64   */
65  private void testGetRepo() {
66      print(svn.getRepoCatalog(""));
67  }
68
69  /**
70   * 检出到本地路径
71   *
72   * @author Allen
73   * @date 2016年8月12日
74   */
75
76  private void testCheckOut() {
77      svn.checkOut("xxx", targetHead);
78  }
79
80  /**
81   * 添加文件到svn
82   *
83   * @author Allen
84   * @date 2016年8月12日
85   */
86
87  private void testAdd() {
88      String[] str = new String[] { targetHead + "/4/a/", targetHead + "/4/", targetHead + "/4/" };
89      svn.add(str, "haha", false, new ISvnDbLog<String>() {
90          @Override
91          public boolean addLog(String name, SvnConfig dbType, long versionId, File[] files) {
92              System.out.println("Add 到 DB 了");
93              return true;
94          }
95      });
96
97      @Override
98      public List<String> getLog(String name, Date startTime, Date endTime) {
99          System.out.println("get 到 log 了");
100         return null;
101     }
102 }
103
104 /**
105  * 删除文件到svn
106  *
107  * @author Allen
108  * @date 2016年8月12日
109  */
110
111 private void testDel() {
112     String[] str = new String[] { targetHead + "/4/a/", targetHead + "/4/", targetHead + "/4/" };
113     svn.delete(str, true, "haha", false, new ISvnDbLog<String>() {
114         @Override
115         public boolean addLog(String name, SvnConfig dbType, long versionId, File[] files) {
116             System.out.println("del 到 DB 了");
117             return true;
118         }
119     });
120
121     @Override
122     public List<String> getLog(String name, Date startTime, Date endTime) {
123         System.out.println("get 到 log 了");
124         return null;
125     }
126 }
127
128 /**

```

关闭





```

129      * 更新文件到svn
130      *
131      * @author Allen
132      * @date 2016年8月12日
133      */
134
135      private void testUpdate() {
136          String str = targetHead + "/4/a/b/";
137          svn.update(strs, "哈哈", false, new ISvnDbLog<String>() {
138              @Override
139              public boolean addLog(String name, SvnConfig dbType, long versic
140                  System.out.println("update 到 DB 了");
141                  return true;
142              }
143
144              @Override
145              public List<String> getLog(String name, Date startTime, Date endT
146                  System.out.println("get 到 log 了");
147                  return null;
148              }
149          });
150      }
151
152      /**
153       * 测试库比对
154       *
155       * @author Allen
156       * @date 2016年8月12日
157       */
158
159      private void testDiff() {
160          String[] strs = new String[] { targetHead + "/4/a/b/" };
161          List<String> s = svn.diffPath(new File(strs[0]));
162          for (String t : s)
163              System.out.println(t);
164      }
165
166      private void testCleanup() {
167          String[] strs = new String[] { targetHead + "/4/a/b/" };
168          svn.cleanup(new File(strs[0]));
169      }
170
171      /**
172       * 打印当前版本库路径目录
173       */
174      private void print(List<SvnRepoPojo> paramList) {
175          System.out.print("commitMessage ");
176          System.out.print("\t\t date \t ");
177          System.out.print("\t kind \t ");
178          System.out.print("\t name \t ");
179          System.out.print("\t repositoryRoot \t ");
180          System.out.print("\t revision \t ");
181          System.out.print("\t size \t ");
182          System.out.print("\t url \t ");
183          System.out.print("\t author \t ");
184          System.out.println("\t state \t ");
185          Collections.sort(paramList, new Comparator<SvnRepoPojo>() {
186              @Override
187              public int compare(SvnRepoPojo o1, SvnRepoPojo o2) {
188                  return o1.getName().compareTo(o2.getName());
189              }
190          });
191          for (SvnRepoPojo pojo : paramList) {
192              System.out.print("\t" + pojo.getCommitMessage() + "\n");
193              System.out.print("\t" + pojo.getDate().toString() + "\n");
194              System.out.print("\t" + pojo.getKind() + "\n");
195              System.out.print("\t" + pojo.getName() + "\n");
196              System.out.print("\t" + pojo.getRepositoryRoot() + "\n");
197              System.out.print("\t" + pojo.getRevision() + "\n");
198              System.out.print("\t" + pojo.getSize() + "\n");
199              System.out.print("\t" + pojo.getUrl() + "\n");

```

关闭





计算机学校排名





云服务器吧



```
200         System.out.print("\t" + pojo.getAuthor() + "\t");
201         System.out.print("\t" + pojo.getState() + "\t");
202         System.out.print("\r\n");
203     }
204 }
205
206 public static void main(String[] args) throws Exception {
207     new exmple().testCore();
208 }
209 }
```

就这样吧。demo而已看个用法和思路。具体优化及细节也没有时间去做了，毕竟是d.....，

下载地址
<http://download.csdn.net/detail/crazyxljing0621/9755958>

顶

0

踩

0

上一篇 ant发布war项目,支持多服务器选择

下一篇 自己做个java sqlite连接池

相关文章推荐

- 通过SVNKit 用Java操作svn
- 【直播】打通Linux脉络 进程、线程和调度--宋宝华
- java实现svn，svnkit框架的简单应用
- 【直播】系统集成工程师必过冲刺--任钰
- java调用svnkit工具类上传本地文件到svn服务器
- 【直播】机器学习30天系统掌握--唐宇迪
- JAVA调用svnkit实现checkout
- 【课程】Oracle从入门到精通--文心

- 使用svnkit api，纯java操作svn，实现svn提交，...
- 【套餐】Android入门实战教程--巫文杰
- SVNKit 示例 简介
- 【课程】 C++语言基础 --贺利坚
- SVN版本控制服务搭建、配置和使用详细步骤！...
- 版本控制：SVN和GIT的一些使用感受
- svn代码版本管理总结
- SVN的java版WEB管理工具-svnadmin-3.0.5

查看评论

- 3楼 MDyuezhang 2017-08-10 11:29发表



可以发一下源码吗，谢谢。1321345131@qq.com
- 2楼 尘封的软件 2017-02-14 18:02发表



求整个project的demo发我一份，谢谢！ duyishuaixue@163.com
- 1楼 尘封的软件 2017-02-14 18:01发表



求源代码，谢谢。
- Re: is Allen 2017-02-16 16:02发表



回复duyishuaixue：已经更新到csdn download

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

关闭



游戏培训学校



2017/9/4

java实现svn，svnkit框架的简单应用 - 曾经沧海难为水的博客 - CSDN博客

公

江

京



计算机学校排名





云服务器吧



服

webmaster@csdn.net

400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

1999-2017, CSDN.NET, All Rights Reserved



关闭



游戏培训学校

