

公告

昵称：鱼我所欲也
园龄：7个月
粉丝：7
关注：1
+加关注

搜索

找找看

谷歌搜索

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

我的标签

- java(13)
- svn(9)
- 多线程(8)
- 缓存(6)
- 版本管理(6)
- 版本控制(6)
- nosql(6)
- linux(6)
- java web(5)
- javaweb(5)
- 更多

随笔分类

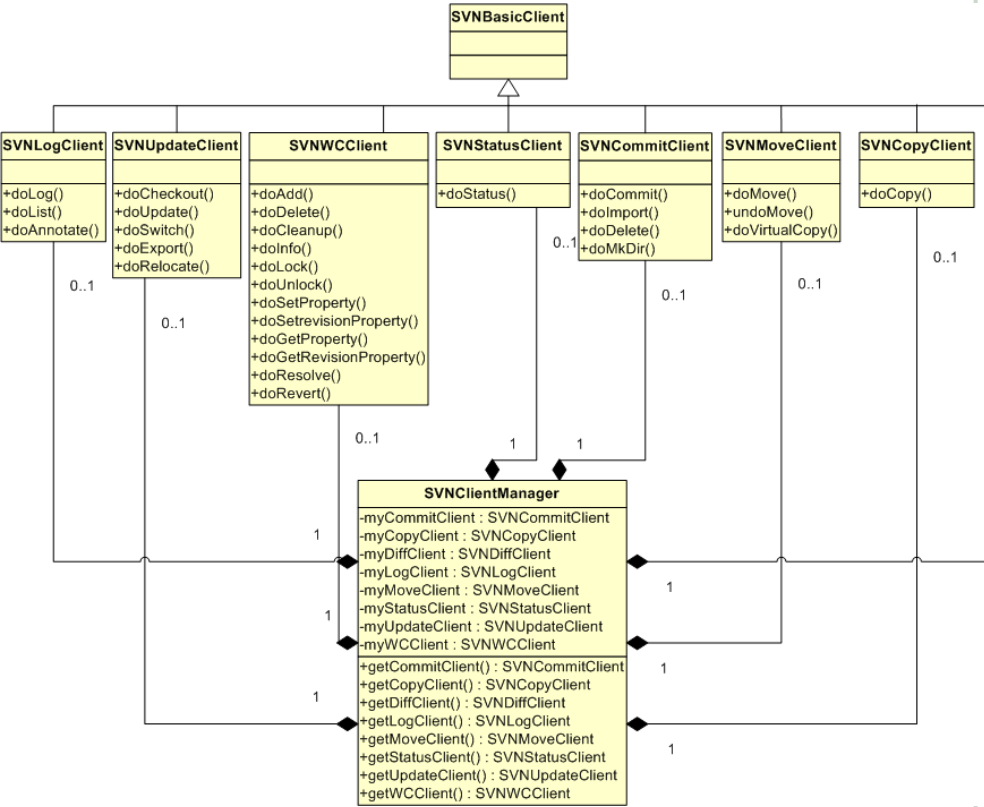
- ant(1)
- cache(4)
- dwr(1)
- git(1)
- haproxy(2)
- java(20)
- javascript(1)
- jboss(1)
- jdbc
- jndi
- linux(6)
- maven(7)
- memcached(4)
- mysql(3)
- redis(2)
- solr(1)
- springMVC(6)
- Swing(1)
- tomcat(1)
- 版本控制、管理工具(11)
- 计算机基础(2)
- 硬件、装机(1)

随笔-72 文章-0 评论-2

SVNKit学习——使用High-Level API管理Working Copy示例(六)

本篇内容是基于SVNKit High-Level API实现的针对Working copy的操作，操作内容与SVN图形化界面、命令行类似。

High-Level API类图：



核心思想：

所有操作由各种SVN*Client来管理，而SVNClientManager类封装了所有的SVN*Client，所以我们通常有两种做法（更倾向于后一种）：

- 1. 可以分别实例化各种SVN*Client类。
- 2. 实例化一个SVNClientManager类，实际上在首次请求SVNClient类的时候，它就已经实例化了每个SVN*Client类了。

示例编码：

```
package com.demo;

import com.google.gson.Gson;
import org.tmatesoft.svn.core.*;
import org.tmatesoft.svn.core.auth.ISVNAuthenticationManager;
import org.tmatesoft.svn.core.internal.io.dav.DAVRepositoryFactory;
import org.tmatesoft.svn.core.wc.*;
import java.io.File;

/**
 * 使用High-Level API来操作Working Copy，由于所有的操作比较贴近SVN命令行、客户端的操作，所以理解起来相对容易一些。
 * 对于Working Copy的不同操作被分类封装到不同的SVN*Client中，SVNClientManager类包含这些SVN*Client，所以通常使用SVNClientManager类可以很方便的管理这些SVN*Client
 * 所以我们在操作Working Copy时有两种选择：1.可以分别实例化各种SVN*Client类。2.实例化一个SVNClientManager类，实际上在首次请求SVNClient类的时候，它就已经实例化了每个SVN*Client类了。
 * 具体结构详见类图
 */
public class ManageWorkingCopy {
```

随笔档案

- 2017年4月 (6)
- 2017年3月 (7)
- 2017年2月 (47)
- 2017年1月 (4)
- 2016年12月 (7)
- 2016年11月 (1)

最新评论

- 1. Re:SVNKit学习——svn二次开发背景和闲谈 (一)
使用svnkit如何解决merge或者commit时候的冲突问题,我是直接合并服务器端的trunk和branch
--rking_java
- 2. Re:SVNKit学习——使用低级别的API (ISVNEditor接口) 直接操作Repository的目录和文件 (五)
受益匪浅。最近我们也正在用svnkit做二次开发，刚做了版本变更的通知功能。现在正在找API，能查出目录的用户列表，但一直没找到，请问楼主有啥建议不？
在windows服务端VisualSvnServ.....
--tinyhead

阅读排行榜

- 1. SVN学习——简单入门之创建仓库、导入、检出 (一) (2506)
- 2. 关于程序中使用servlet-api.jar和jsp-api.jar与服务器libjar包冲突的问题(1617)
- 3. mysql的sql_mode介绍和修改 (1190)
- 4. SVNKit学习——svn二次开发背景和闲谈 (一) (791)
- 5. Swing入门学习(705)

评论排行榜

- 1. SVNKit学习——svn二次开发背景和闲谈 (一) (1)
- 2. SVNKit学习——使用低级别的API (ISVNEditor接口) 直接操作Repository的目录和文件 (五) (1)

推荐排行榜

- 1. SVNKit学习——svn二次开发背景和闲谈 (一) (1)
- 2. SVNKit学习——使用低级别的API (ISVNEditor接口) 直接操作Repository的目录和文件 (五) (1)
- 3. git学习——简介、使用 (一) (1)
- 4. memcached学习——常用命令+基于java客户端的3种简单实现 (二) (1)
- 5. Swing入门学习(1)

```
private static SVNClientManager svnClientManager;
private static final Gson gson = new Gson();

public static void main(String[] args) throws Exception{
    //1.根据访问协议初始化工厂
    DAVRepositoryFactory.setup();
    //2.使用默认选项
    ISVNOptions isvnOptions = SVNWCUtil.createDefaultOptions(true);
    //3.初始化权限
    String username = "wly";
    String password = "wly";
    char[] pwd = password.toCharArray();
    ISVNAuthenticationManager isvnAuthenticationManager =
SVNWCUtil.createDefaultAuthenticationManager(username,pwd);
    //4.创建SVNClientManager的实例
    svnClientManager =
SVNClientManager.newInstance(isvnOptions,isvnAuthenticationManager);
    //=====demo
start=====
    //1.导入-import
    // doImport();
    //2.检出-checkout
    // doCheckout();
    //3.创建目录-mkdir
    // doMkDir();
    //4.提交-commit
    // doCommit();
    //5.更新-update
    // doUpdate();
    //6.切换分支-switch
    //7.添加-add, 在commit中有体现, 提交前, 没有加入版本控制的文件需要加入版本控制
    //8.锁定-lock
    // doLock();
    //9.删除-delete
    // doDelete();
    //10.拷贝-copy
    // doCopy();
    //11.状态-status
    // doStatus();
    //12.信息-info
    // doInfo();
}

/*
 * 导入-import
 * @throws SVNException
 */
private static void doImport() throws SVNException{
    SVNProperties svnProperties = new SVNProperties();
    boolean useGlobalIgnores = true;
    boolean ignoreUnknownNodeTypes = true;
    SVNCommitInfo svnCommitInfo = svnClientManager.getCommitClient().doImport(new
File("E:\\svnWorkspace\\commonProject"),SVNURL.parseURIEncoded("https://wlyfree-
PC:8443/svn/testRepository/trunk"),"初始化导
入",svnProperties,useGlobalIgnores,ignoreUnknownNodeTypes,SVNDepth.INFINITY);
    System.out.println("执行import操作成功, 导入结果:" + gson.toJson(svnCommitInfo));
}

/**
 * 检出-checkout
 * 参数:
 * 仓库地址
 * 本地Working Copy地址
 * Peg Revision
 * Revision
 * 检出深度, 一般递归检出
 * 是否允许没有版本的障碍物, true的话允许, false不允许, false在checkout的时候如果有障碍物就会停止检
出, 所以一般是true
 * 返回值:long 当前版本号
 * @throws SVNException
 */
private static void doCheckout() throws SVNException{
    long nowRevision =
svnClientManager.getUpdateClient().doCheckout(SVNURL.parseURIEncoded("https://wlyfree-
```

```

PC:8443/svn/testRepository/trunk"), new File("E:\\svnWorkspace\\projectWorkingCopy"),
SVNRevision.HEAD, SVNRevision.HEAD, SVNDepth.INFINITY, true);

    System.out.println("执行checkout操作成功, 当前检出的版本号是: " + nowRevision);
}

/**
 * 创建目录-mkdir
 * @throws SVNException
 */
private static void doMkDir() throws SVNException{
    String commitMessage = "创建一个目录";
    SVNCommitInfo svnCommitInfo = svnClientManager.getCommitClient().doMkDir(new SVNURL[]
(SVNURL.parseURIEncoded("https://wlyfree-
PC:8443/svn/testRepository/trunk/aaa"), SVNURL.parseURIEncoded("https://wlyfree-
PC:8443/svn/testRepository/trunk/bbb")), commitMessage);
    System.out.println("执行mkdir操作成功, 操作结果: " + gson.toJson(svnCommitInfo));
}

/**
 * 提交-commit
 * commit更改一个文件时, 如果文件本身存在, 则需要提交其父目录
 * @throws SVNException
 */
private static void doCommit() throws SVNException{
    File[] files = new File[]{new File("E:\\svnWorkspace\\projectWorkingCopy"), new
File("E:\\svnWorkspace\\projectWorkingCopy\\ccc")};
    //没有版本号的先执行add操作
    for(File tempFile : files){
        SVNStatus status = svnClientManager.getStatusClient().doStatus(tempFile, true);
        System.err.println(status);
        if(status == null || status.getContentsStatus() ==
SVNStatusType.STATUS_UNVERSIONED) {
            System.out.println("文件" + tempFile.getName() + "无版本号");
            svnClientManager.getWCClient().doAdd(tempFile, false, false, false,
SVNDepth.INFINITY, false, false);
        }
    }
    //执行commit操作
    svnClientManager.getCommitClient().setIgnoreExternals(false);
    SVNProperties svnProperties = new SVNProperties();
    String[] changeLists = new String[]{};
    SVNCommitInfo svnCommitInfo =
svnClientManager.getCommitClient().doCommit(files, false, "提交操
作", svnProperties, changeLists, false, false, SVNDepth.fromRecurse(true));
    System.out.println("执行commit操作成功, 操作结果: " + gson.toJson(svnCommitInfo));
}

/**
 * 更新-update
 * @throws SVNException
 */
private static void doUpdate() throws SVNException{
    long nowRevision = svnClientManager.getUpdateClient().doUpdate(new
File("E:\\svnWorkspace\\projectWorkingCopy"), SVNRevision.HEAD, SVNDepth.INFINITY, true, false);
    System.out.println("执行update操作成功, 当前版本号: " + nowRevision);
}

/**
 * 锁定-lock
 * @throws SVNException
 */
private static void doLock() throws SVNException{
    //    svnClientManager.getWCClient().doLock(new SVNURL[]
{SVNURL.parseURIEncoded("https://wlyfree-
PC:8443/svn/testRepository/trunk/bbb/aa.txt")}, true, "给文件加锁");
    svnClientManager.getWCClient().doLock(new File[]{new
File("E:\\svnWorkspace\\projectWorkingCopy\\bbb\\aa.txt")}, true, "给文件加锁");
    System.out.println("给文件加锁成功");
}

/**
 * 删除-delete
 * @throws SVNException
 */

```

```
private static void doDelete() throws SVNException{
    SVNCommitInfo svnCommitInfo = svnClientManager.getCommitClient().doDelete(new SVNURL[]
{SVNURL.parseURIEncoded("https://wlyfree-
PC:8443/svn/testRepository/trunk/bbb"),SVNURL.parseURIEncoded("https://wlyfree-
PC:8443/svn/testRepository/trunk/b.txt")}),"执行删除操作, 删除一个目录bbb一个文件b.txt");
    System.out.println("执行delete操作成功, 操作结果: " + gson.toJson(svnCommitInfo));
}

/**
 * 复制-copy
 * @throws SVNException
 */
private static void doCopy() throws SVNException{
    SVNCopySource svnCopySource1 = new
SVNCopySource(SVNRevision.HEAD,SVNRevision.HEAD,SVNURL.parseURIEncoded("https://wlyfree-
PC:8443/svn/testRepository/trunk/aaa/aa.txt"));
    SVNCopySource svnCopySource2 = new
SVNCopySource(SVNRevision.HEAD,SVNRevision.HEAD,SVNURL.parseURIEncoded("https://wlyfree-
PC:8443/svn/testRepository/trunk/aaa/bb.txt"));
    svnClientManager.getCopyClient().doCopy(new SVNCopySource[]
{svnCopySource1,svnCopySource2},new
File("E:\\svnWorkspace\\projectWorkingCopy\\bbb"),false,false,true);
    System.out.println("执行copy操作成功");
}

/**
 * 状态-status
 * @throws SVNException
 */
private static void doStatus() throws SVNException{
    SVNStatus svnStatus = svnClientManager.getStatusClient().doStatus(new
File("E:\\svnWorkspace\\projectWorkingCopy\\a.txt"),true);
    System.out.println("执行status操作成功, 操作结果: " + gson.toJson(svnStatus));
}

/**
 * 信息-info
 * @throws SVNException
 */
private static void doInfo() throws SVNException{
    SVNInfo svnInfo = svnClientManager.getWCClient().doInfo(new
File("E:\\svnWorkspace\\projectWorkingCopy\\a.txt"),SVNRevision.HEAD);
    System.out.println("执行info操作成功, 操作结果: " + gson.toJson(svnInfo));
}
}
```

分类: [版本控制、管理工具](#)

标签: [svnkit](#), [java](#), [svn](#), [svn二次开发](#), [版本控制](#), [版本管理](#)

好文要顶

关注我

收藏该文

鱼我所欲也

关注 - 1

粉丝 - 7

+加关注

1

0

« 上一篇: [SVNKit学习——使用低级别的API \(ISVNEditor接口\) 直接操作Repository的目录和文件 \(五\)](#)
» 下一篇: [Swing入门学习](#)

posted @ 2016-12-07 18:39 [鱼我所欲也](#) 阅读(572) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
【免费】从零开始学编程, 开发者专属实验平台免费实践!

**最新IT新闻:**

- 日媒：东芝白电借助美的重返中国市场
 - 太空探索的钱没白花 GPS一年能带来560亿美元回报
 - 硅谷热钱涌动 Jawbone恰恰是因为钱太多才走上绝路
 - 苹果9亿美元在丹麦德国边境建数据中心 2019年启用
 - 法拉第美国建厂计划搁浅 贾跃亭造车前景堪忧
- » 更多新闻...

**最新知识库文章:**

- 小printf的故事：什么是真正的程序员？
 - 程序员的工作、学习与绩效
 - 软件开发为什么很难
 - 唱吧DevOps的落地，微服务CI/CD的范本技术解读
 - 程序员，如何从平庸走向理想？
- » 更多知识库文章...