

# 1.IntelliJ-IDEA预装的版本控制介绍

我们来看IntelliJ-IDEA的版本控制设置区域

打开File>Settings>Version Control



#### 可以看到里面有git插件

这让很多人认为 IntelliJ IDEA 自带了 SVN 或是 Git 等版本控制工具,认为只要安装了 IntelliJ IDEA 就可以完全使用版本控制应有的 功能。这完全是一种错误的解读, IntelliJ IDEA 是自带对这些版本 控制工具的支持插件,但是该装什么版本控制客户端还是要照样装 的.

上面可以看到里面还自带了github插件,这是因为目前太多人使用

Github 进行协同或是项目版本管理

# 2.git.exe和Github的配置 1.配置git.exe

#### 先确保电脑上已经安装了Git和TortoiseGit

接着再配置git客户端



## 2.配置Github

下面是选择账号和密码登录

U Settings	-		×
٩		Version Control > GitHub @ For current project R	eset
Version Control Background		Host: github.com 用户名 Auth Type: Password	<b>•</b>
Ignored Files		Login: milsevol	
Issue Navigation		Password: ······ — 客码	
Changelist Conflicts		Ter	
Shelf		Do not have an account at github.com? Sign up	
GitHub	G		
CVS	ē	Connection successful	
ClearCase		Connection t	
Git		ОК	
Mercurial			
Perforce			
Subversion			

### 3.clone项目从Github上

接着可以将GitHub上的项目checkout下来



如果GitHub上有多个项目的话,那么可以选择其中的一个



点击clone后, IntelliJ-IDEA就会开始clone项目

## 4.将本地项目上传到Github上

#### 选择Import into Version Control

VC <u>S</u> <u>W</u> indow <u>H</u> elp	
Local <u>H</u> istory	
Enable Version Control Integration	
VCS Operations Popup Alt+后引号	
Apply Patch	
Apply Patch from Clipboard	
Checkout from Version Control	
Import into Version Control	Import into C <u>V</u> S
Browse VCS Repository	Create Git Repository
	Import into Subversion
Search Everywhere Double Shift	Share Project (Subversion)
	Create Mercurial Repository
Go to File Ctrl+Shift+Nhttp:/	🖸 Sharê Project oh/GitiHubevol

#### 然后填写相关的信息



这时候会提示你选择上传的文件



#### 仓库创建成功后,会出现如下的提示

13:50 Successfully shared project on GitHub: Hello-world sevol

### 再看GitHub下,多了一个项目仓库

 Popular repositories
 Customize your pinned repositories

 Hello-word
 IM-project.github.io

 Github新手引导测试
 仿微信的即时通讯项目

 Java
 Java

 mavn\_springmvc\_mybatis\_test
 ¥1

 mavn\_springmvc\_mybatis@htt
 Hello-world

 Java
 Java

#### 点击进去,项目的相关结构如下

📮 milsevol / Hello-world			Unwatch - 1 Star 0 §Fork 0
<>Code (!) Issues 0 (?) Pull re	equests 0 III Projects 0	🗉 Wiki 🔅 Settings Insigh	nts 🕶
No description, website, or topics prov Add topics	vided.		Edit
🕝 1 commit	ဖို <b>၊</b> branch	🟷 <b>0</b> releases	🚨 0 contributors
Branch: master   New pull request		Create new file	Upload files Find file Clone or download -
💽 user3 Initial commit			Latest commit 76d5fdf 7 minutes ago
🖿 .idea		Initial commit	7 minutes ago
🖿 src/com/cui/demo		Initial commit	7 minutes ago
Hello-world.iml		Initial commit	7 minutes ago
Help people interested in this repositor	v understand vour project by addi	ng a README.	Add a README

# 二、版本控制的主要操作按钮

#### git的操作的入口如下:

rien ► ■ .i ► ■ c ▼ ■ s	C <u>o</u> py Path Copy as Plain Text Cop <u>y</u> Relative Path <u>P</u> aste	Ctrl+Shift+C Ctrl+Alt+Shift+C Ctrl+V	ui.demo;
V E	Find <u>U</u> sages Find in <u>P</u> ath Replace in Path	Alt+F7 Ctrl+Shift+F Ctrl+Shift+R	<pre>xueyong on 2017/5/27. HelloWorld {</pre>
<b>é -</b> F	Analy <u>z</u> e	•	+ Add Ctrl+Alt+A
► II <b>II</b> I Exte	<u>R</u> efactor Add to F <u>a</u> vorites Show Image Thumbnails	► Ctrl+Shift+T	A <u>n</u> notate Show Current Revision Gampare with the Same Repository Version
	<u>R</u> eformat Code Optimi <u>z</u> e Imports Remove Module	Ctrl+Alt+L Ctrl+Alt+O Delete	Compare with Latest Repository <u>V</u> ersion <u>C</u> ompare with Compare with Branch
/ersion Co	Build <u>M</u> odule 'Hello-world' R <u>e</u> build Module 'Hello-world'	Ctrl+Shift+F9	Show <u>History</u> Show History <u>for</u> Selection
user3	Local <u>H</u> istory <u>G</u> it	۲ ۲	ש Revert Ctrl+Alt+Z Repositiony blog. csdn. net/milsevol

#### 也可以在工具栏中进行操作

注意这五个按钮



五个按钮的功能如下:

第一个按钮: Update Project 更新项目。 第二个按钮: Commit changes 提交项目上所有变化文件。 点击这个按钮不会立马提交所有文件,而是先弹出一个被修改 文件的一个汇总框,具体操作下面会有图片进行专门介绍。 第三个按钮: Compare with the Same Repository Version 当前文件与服务器上该文件通版本的内容进行比较。如果当前 编辑的文件没有修改,则是灰色不可点击。 第四个按钮: Show history 显示当前文件的历史记录。

第五个按钮:Revert 还原当前被修改的文件到未被修改的版

#### 本状态下。如果当前编辑的文件没有修改,则是灰色不可点

击。

具体作用如下,如下面添加了一行,如果不满意,可以还

原



#### 但是如果想提交的话,可以选择这个





#### 填写相关的信息后,可以直接选择commit和push



点击push



#### 提交完毕后,我们可以查看相关的记录



### 可以看到如下的记录



#### 这时候GitHub上项目就会出现相关的信息了

第一次 🎖 master	r修改 「	(提交			Browse files
🔘 user	r3 ? c	ommitted 5 r	ninutes ago	1 parent 76d5fdf	commit 171e3f497bc6a8581139f155477f05f456ceb8df
D Showir	ng <b>1 cl</b>	nanged file v	with <b>1 addition</b> and <b>0 deletions</b> .		Unified Split
1	src	/com/cui/dem	no/HelloWorld.java		View 🗸
Σţ		@@ -5,6 +5	,7 @@		
5	5	*/			
6	6	public cl	ass HelloWorld {		
7	7	p	u <u>blic static void main(String[] agrs){</u>		
	8	+	System.out.println("版本控制相关");		
8	9		System.out.println("Hello word");		
9	10	)			
10	11	3			http://blog.ogdp.not/milcovol

# 三、版本控制的常用设置

## 1.Version Control面板

U Settings		
٩		Version Control 🖷 For current project
► Appearance & Behavior		Directory 1 VCS +
Keymap		C:\Users\xueyong\IdeaProjects\Hello-world Git
▶ Editor		
Plugins		
<ul> <li>Version Control</li> </ul>	G	
Confirmation		
Background		<project> - Content roots of all modules, all immediate descendants of project base directory, and .idea directory</project>
Ignored Files		contents
Issue Navigation		✓ Limit history to: 1,000 ¢ rows
Changelist Conflicts		Show directories with changed descendants
Shelf		Show changed in last 31 🜲 days
GitHub		Filter Update Project information by scope Manage Scopes
CVS		
ClearCase		Commit message right margin (columns): 2 🗘 🖾 Wrap when typing reaches right margin
Git		Show unversioned files in Commit dialog
Mercurial		Check commit message spelling
?		http://b <b>9%g.</b> csd <b>hanfe</b> t/mi <b>1</b> 98V

- 1. 代表这个项目的版本控制是GIT
- 2. Show directories with changed descendants 表示子目

#### 录有文件被修改了,则该文件的所有上层目录都显示版本

控制被修改的颜色(建议勾选)

## 2.Confimation确认信息面板

添加新文件和删除文件的提示

٩			Version Control > Confirmation n For cur			
►	Appearance & Behavior		When files are created	When files are deleted		
Keymap ▶ Editor Plugins			<ul> <li>Show options before adding to version control</li> </ul>	Show options before removing from version control		
			• <u>A</u> dd silently	Remove silently		
▼	▼ Version Control @		Do not add	Do not remove		
	Confirmation 🛛 🖻					
	Background		When empty changelist becomes inactive	<ul> <li>Display options dialog when these commands are invoked -</li> </ul>		
	Ignored Files		Show options before removing	🗹 Checkout		
	Issue Navigation 🖷		Bemove silently			
	Changelist Conflicts		Kentove silentiy	🔤 Openetp://blog.csdn.net/milsevo		

# 3.Ignored Files面板





## 4.commit文件的面板介绍

commit文件的时候会出现如下的面板



### **1.show Diff**

show Diff可以比较本地跟服务器文件的不同



### 2. Move to Another ChangeList

这个选项可以将改动的文件分门别类的放置在一个文件夹 中,等修改完毕,可以一起提交(一般用在分模块开发 上,也就是开发完一个模块时,可以将这个模块上改动的 代码设置到一个文件夹中,然后再一起祈祷)



### 3.查看提交信息



# 四、项目分支设置的入口

如果使用Git版本控制的话,可以看到右下角上有相关的控

### 制入口



# 五、Gitlab的使用

Gitlab跟Github类似,都是代码托管的网站,最大的不同是Gitlab创建的项目可以免费私有的,不必像Github那样收费,而且Gitlab还可以搭建自己的私服。所以开源项目一般都是放置在Github,个人私有项目可以放置在公网的Gitlab上,而公司私有的项目可以放置在自己搭建的Gitlab上。 1.创建项目

## 2.对项目进行操作

项目创建完后,可以点击左上角进行查看

= 🦊 Projects	
Projects	
Activity	
Groups	
Milestones	
Issues	0
Merge Requests	0
Snippets	
Help	

#### 点击Project,可以看到项目的列表





#### 码的方式clone下来



If you already have files you can push them using command line instructions below.

Otherwise you can start with adding a <u>README</u>, a <u>LICENSE</u>, or a <u>.gitignore</u> to this project.

You will need to be owner or have the master permission level for the initial push, as the master branch is automatically protected.



#### 接着在IntellIj-IDEA上使用Git将其Clone下来



如上图, check out项目的时候因为不是Github, 所以不再

选择Github,而是选择普通的Git来将项目clone下来



### 最后输入密码,再将其clone下来



#### clone下来后,项目结构如下:



这时候就可以进行正常的add、commit、push操作了

## 3.将项目发布到gitlab上

先在本地创建一个项目

先将项目添加到版本控制中



#### 添加到版本控制后,我们可以看到文件都变成绿色了



接着往本地提交项目

Add to Favorites       >         Show Image Thumbnails       Ctrl+Shift+T         Reformat Code       Ctrl+Alt+L         Optimize Imports       Ctrl+Alt+O         Remove Module       Delete         Build Module 'demo'       Show Current Revision         Rebuild Module 'demo'       Compare with the Same Repository Version         Rebuild Module 'demo'       Ctrl+Shift+F9         Run 'All Tests'       Ctrl+Shift+F10         © Debug 'All Tests'       Ctrl+Shift+F10         % Run 'All Tests' with Coyerage       Show History         Im Create 'All Tests'       Show History for Selection         Local History       Show History	<u>R</u> efactor		•		<pre>PDOCTYPE HIML PU html&gt;</pre>	BFIC\\M3C
Reformat Code       Ctrl+Alt+L         Optimize Imports       Ctrl+Alt+O         Remove Module       Delete         Build Module 'demo'       Show Current Revision         Rebuild Module 'demo'       Ctrl+Shift+F9         Run 'All Tests'       Ctrl+Shift+F10         © Debug 'All Tests'       Ctrl+Shift+F10         % Run 'All Tests' with Coverage       Show History         Im Create 'All Tests'       Show History for Selection         Local History       Show History	Add to F <u>a</u> v Show Imag	orites e Thumbnails	► Ctrl+Shift+T	Þ	head> Commit Directory	
Build Module 'demo'       Image: Compare with the Same Repository Version         Rgbuild Module 'demo'       Ctrl+Shift+F9         Rgbuild Module 'demo'       Ctrl+Shift+F9         Rgbuild Module 'demo'       Ctrl+Shift+F1         Compare with Latest Repository Version         Gompare with Latest Repository Version         Compare with Latest Repository Version         Show History         Compare with Branch         Show History for Selection         Local History       Shevert	<u>R</u> eformat C Optimi <u>z</u> e Ir Remove M	òde nports odule	Ctrl+Alt+L Ctrl+Alt+O Delete	+	Add Annotate Show Current Revision	Ctrl+Alt+A
Greate 'All Tests'     Show History for Selection     Local History     ★ Revert     Ctrl+Alt+Z	Build <u>M</u> odu R <u>e</u> build Ma ▶ Run 'All Tes <u>₩ D</u> ebug 'All ₩ Run 'All Tes	ule 'demo' odule 'demo' tts' Tests' sts' with Co <u>v</u> eraç	Ctrl+Shift+F9 Ctrl+Shift+F10 ge	Ē	Compare with the Same Re Compare with Latest Repos <u>C</u> ompare with Compare with Branch Show <u>H</u> istory	
	Create 'All Local <u>H</u> isto	Tests' ry	•	5	Show History <u>f</u> or Selection <u>Revert</u>	Ctrl+Alt+Z

### 填写提交的相关信息



注意perform code analysis 就不要勾选了,这个选项会

自动检查代码, 会很慢

接着将项目push到服务器中

下面点击push

	Add to ravonites	· · · ·	P	⊲ <neau></neau>				
í	Show Image Thumbnails	Ctrl+Shift+T		Commit Directory		::	Rranches	
r.	<u>R</u> eformat Code	Ctrl+Alt+L		Add	Ctrl+Alt+A	٢	Tan	
2	Optimi <u>z</u> e Imports	Ctrl+Alt+O				-6	Merge Changes	
	Remove Module	Delete					Stash Changes	
	Build <u>M</u> odule 'demo'						UnStash Changes	
	R <u>e</u> build Module 'demo'	Ctrl+Shift+F9				B	Reset HEAD	
	R <u>u</u> n 'All Tests' (	Ctrl+Shift+F10					Remotes	
1	Debug 'All Tests'						Clone	
	Run 'All Tests' with Co <u>v</u> erage		C.	Show <u>H</u> istory			Fetch	
	Create 'All Tests'					vçs	Pull	
	Local <u>H</u> istory	Þ				VCS		
	<u>G</u> it				Þ		Rebase	

### 接下来需要定义远程的服务

1	Push Cor	nmits	-			×
) <	master –	<u>Define remote</u>		Î	D E	) >>
% < < <						
<						
< T <	Push ]	[ags				
	?				Ca	

此时需要再弹出框中设置url,我们先在gitlab新建一个

project,这样的目的是为了获得相关的url

填写url后,再点击push



此时还需要填写gitlab上的密码



push成功后,会出现这样的提示,这是成功的提示:



再查看gitlab上的项目信息,可以看到提交的代码,具体如

下图

≡	🦊 milsevol / demo 🗸	
		Project <b>Repository</b> Registry Issues 0 Merge Requests
		Files Commits Branches Tags Contributors Gra
	master 🗸 den	o / 🛨
	Name	Last commit > 6ae3606b 🜓 49 minutes ago - 提交到本地 🔰 History
	🖿 .idea	提交到本地
	src	提交到本地
	🖹 demo.iml	提交到本地
	pom.xml	提交到本地