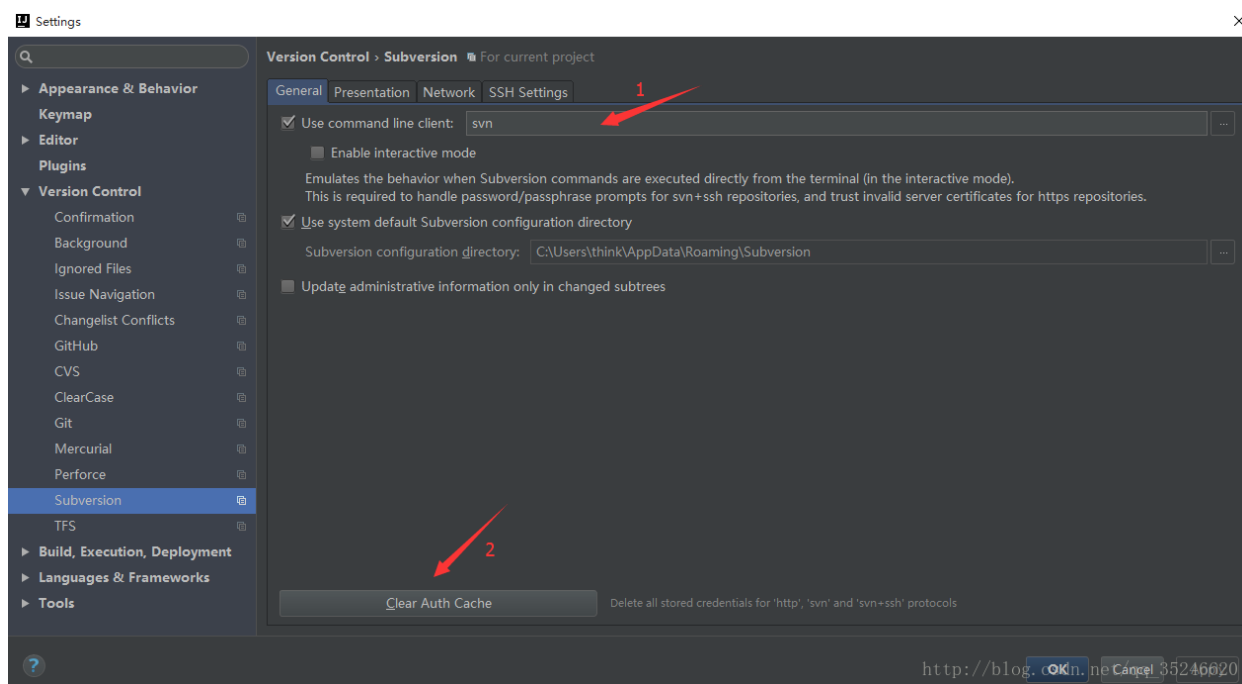


由于 [IntelliJ IDEA](#) 支持的版本控制工具非常的多，但我们真正能够用到的也就两三个而已，因此在本文中，咱们主要介绍 SVN、Git 和 GitHub 的配置方法。

SVN

如果想要在 IntelliJ IDEA 中使用 SVN，则需要事先安装 SVN 客户端或是 TortoiseSVN 这类图形化工具。对于 Windows 系统，推荐大家安装 TortoiseSVN；对于 Mac 系统，则推荐大家安装 CornerStone。



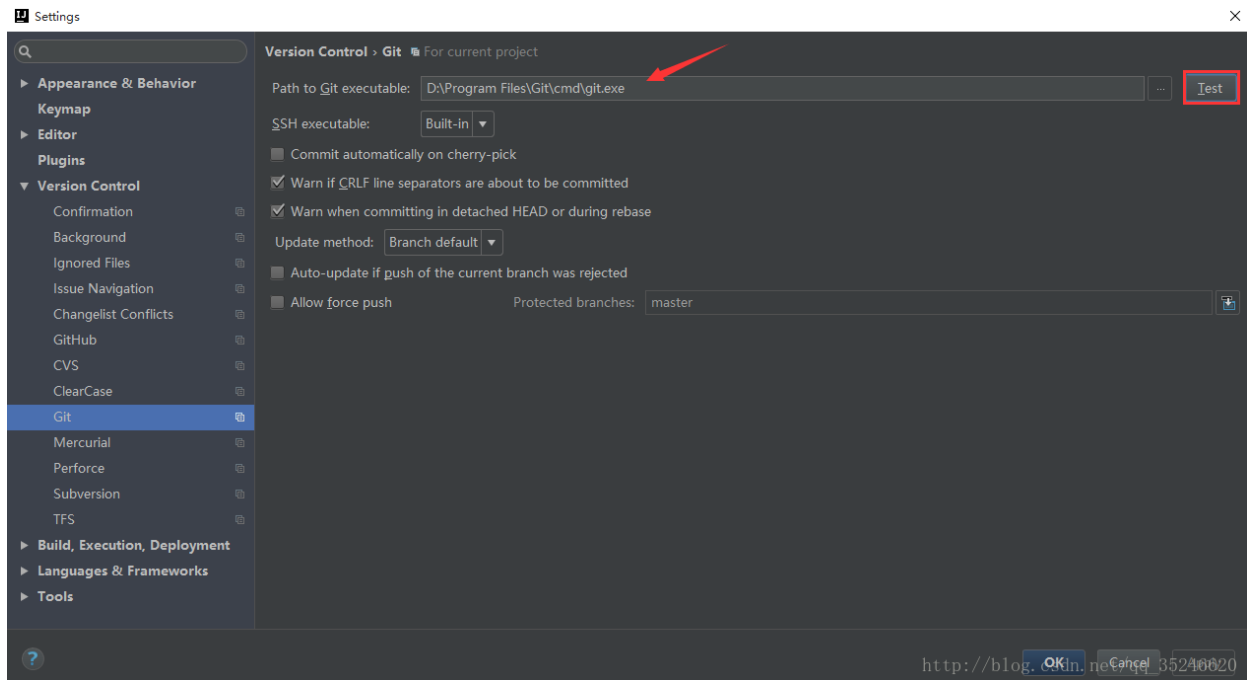
1

- 标注1：Use command line client，使用命令行客户端；
- 标注2：Clear Auth Cache，清除缓存。

如上图所示，勾选 标注1 所示的选项后，表示使用 SVN 命令行客户端，在这里，建议 SVN 的路径根据咱们安装后的路径进行选择，否则 IntelliJ IDEA 可能无法识别到 SVN，以至于报出：Cannot run program "svn"这类错误；在咱们使用 SVN 一段时间之后，如果发现 SVN 有些问题无法解决的话，可以考虑点击标注2 所示的清除缓存按钮。

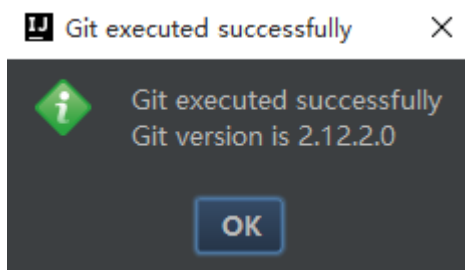
Git

如果想要在 IntelliJ IDEA 中使用 Git，同样需要事先安装 Git 客户端，不过在安装 Git 客户端的过程中，咱们可以自由选择是否同时使用 Windows 命令行工具。



2

如上图所示，如果咱们事先安装了 Git 客户端的话，则会在 Path to Git executable 中自动定位到 Git 的可执行文件，然后点击 Test：

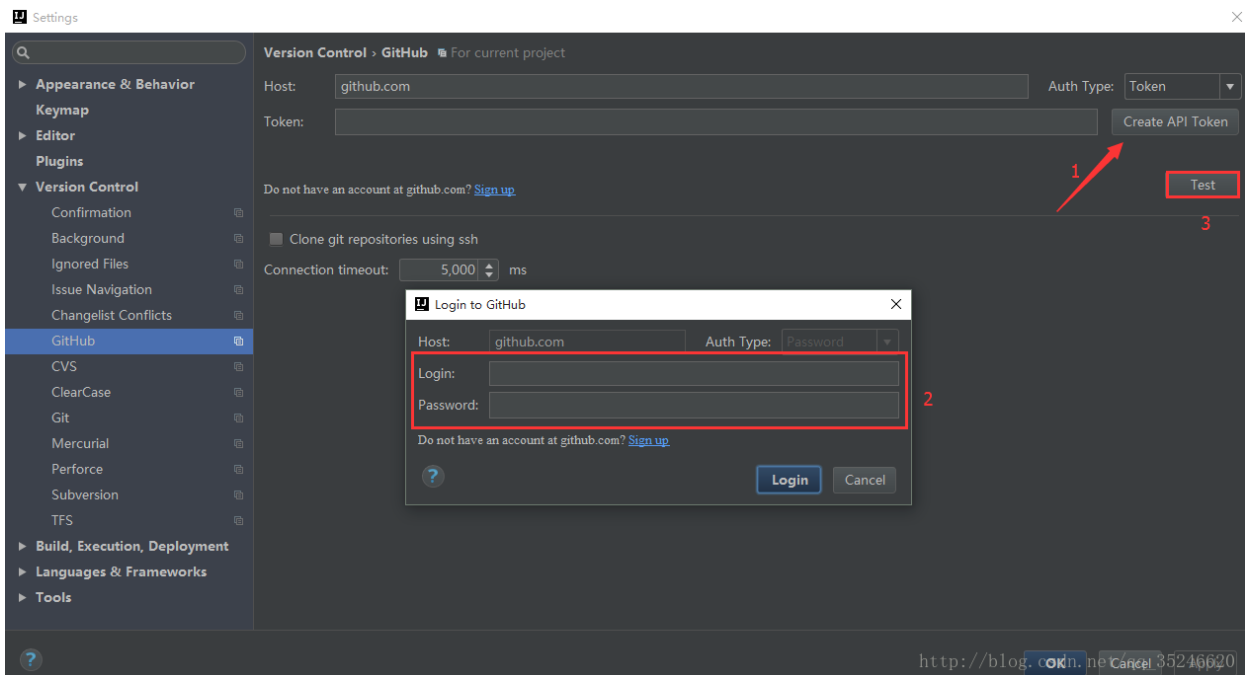


3

如上图所示，显示 Git executed successfully，则表示分布式版本控制系统 Git 可用。

GitHub

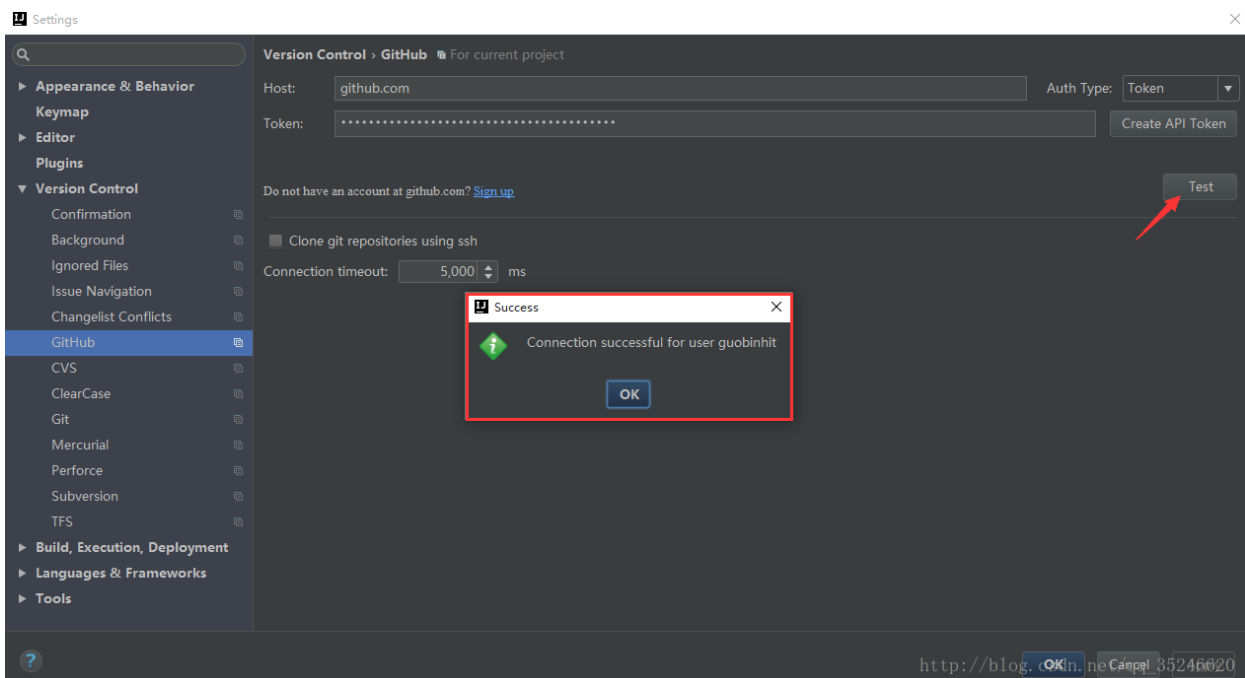
对于一个励志于在互联网浪潮中闯出一番天地的高逼格程序猿来说，如果不知道 GitHub 的话，貌似有些太 low 啦！



5

- 标注1 : Create API Token , 创建 API Token ;
- 标注2 : Login to GitHub , 登录 GitHub 账号 ;
- 标注3 : Test , 测试连接是否成功。

如上图所示，在 IntelliJ IDEA 中，提供了对 GitHub 的支持功能。当咱们登录 GitHub 账号之后，点击Test进行测试：



6

如上图所示，显示Connection successfully for user guobinhit，表示咱们已经将此 IntelliJ IDEA 连接到 GitHub 账号为guobinhit的账号之中啦！