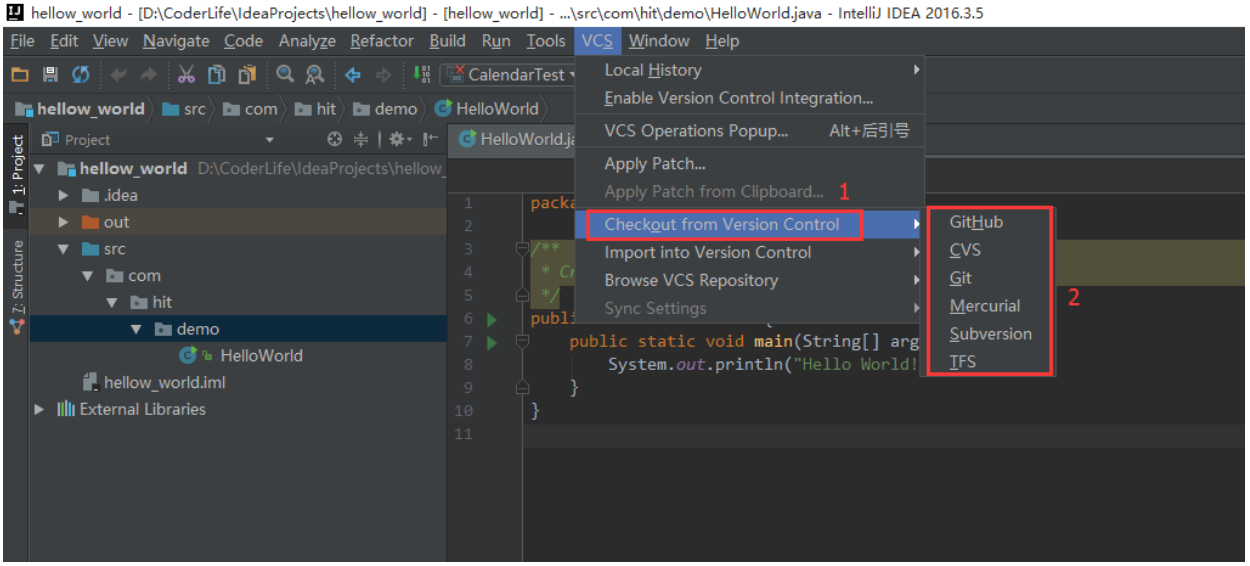
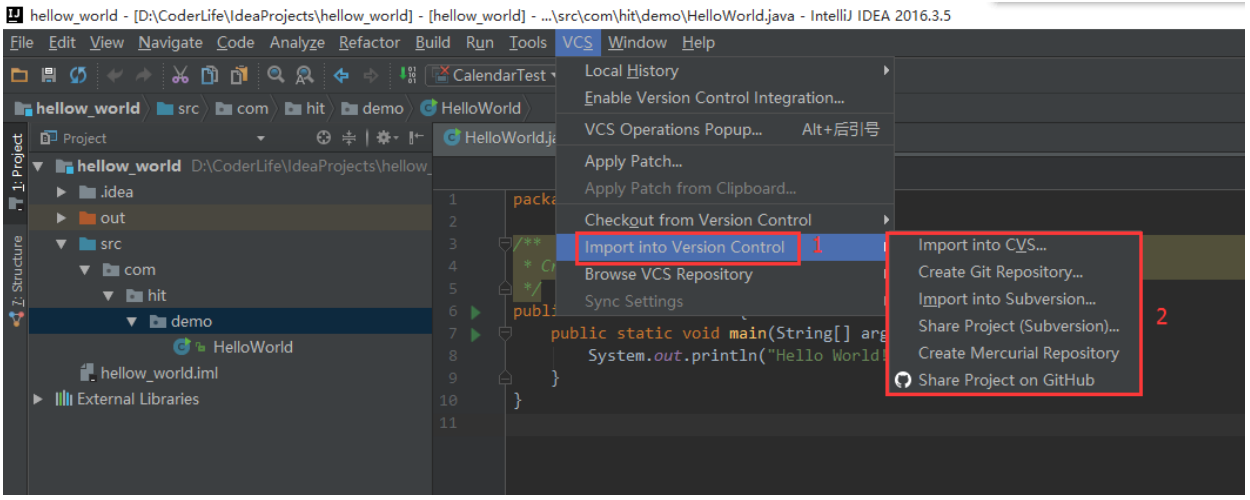
前两次我们已经简单了解了 IntelliJ IDEA 的版本控制机制，那么接下来，就让我们一起看看在 IntelliJ IDEA 中进行具体的版本控制操作。



2

- 标注1：Checkout from Version Control，从版本控制系统中检出项目；
- 标注2：IntelliJ IDEA 支持的版本控制系统，包括GitHub、CVS和Git等。

如上图所示，我们可以通过Checkout from Version Control，从版本控制系统，如GitHub、CVS和Git等中检查项目。相对的，既然我们可以从版本控制系统中检出项目，那么自然也可以将项目上传到版本控制系统之中。

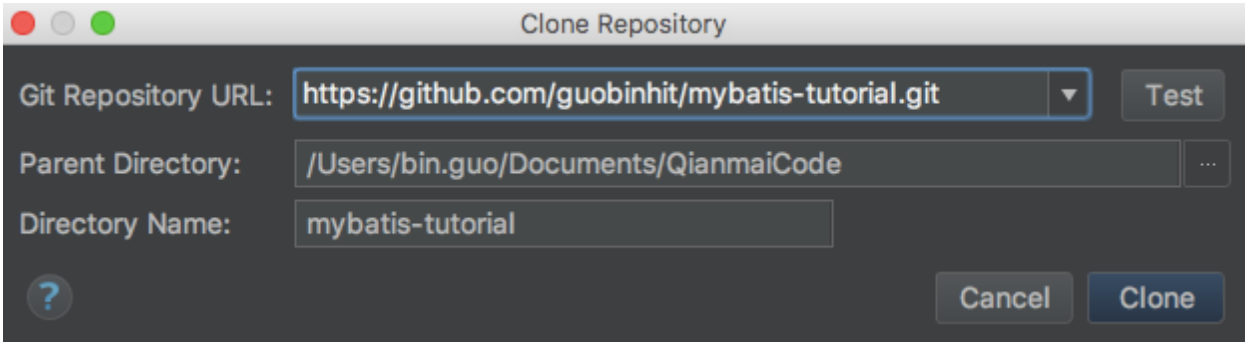


3

- 标注1：Import into Version Control，将项目上传到版本控制系统；
- 标注2：IntelliJ IDEA 支持的版本控制系统，包括GitHub、CVS和Git等。

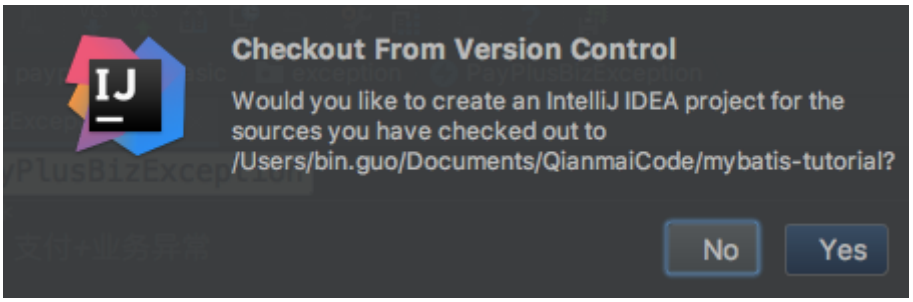
如上图所示，通过以上操作，就可以将代码上传到版本控制系统之中。

现在，以博主的 GitHub 上面的项目mybatis-tutorial为例，检出项目：



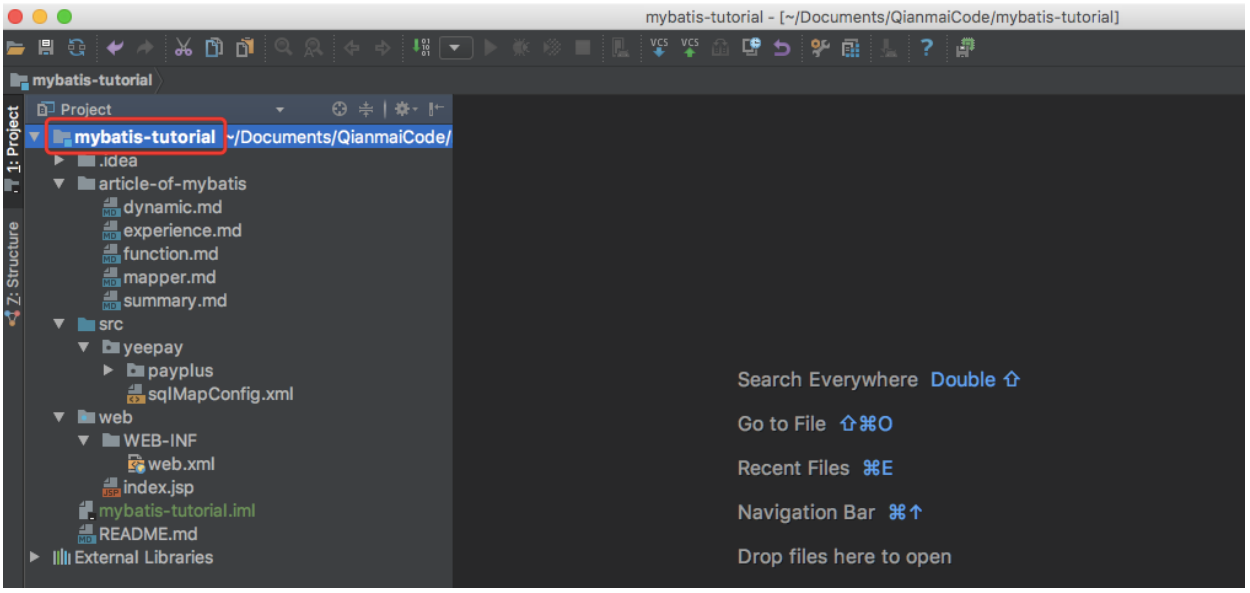
GitHub

如上图所示，首先选择Checkout from Version Control -> GitHub，登录账号，然后选择我们想要检出的项目，点击Clone，此“克隆”的概念来自于 Git，表示把远程仓库的项目检出到本地：



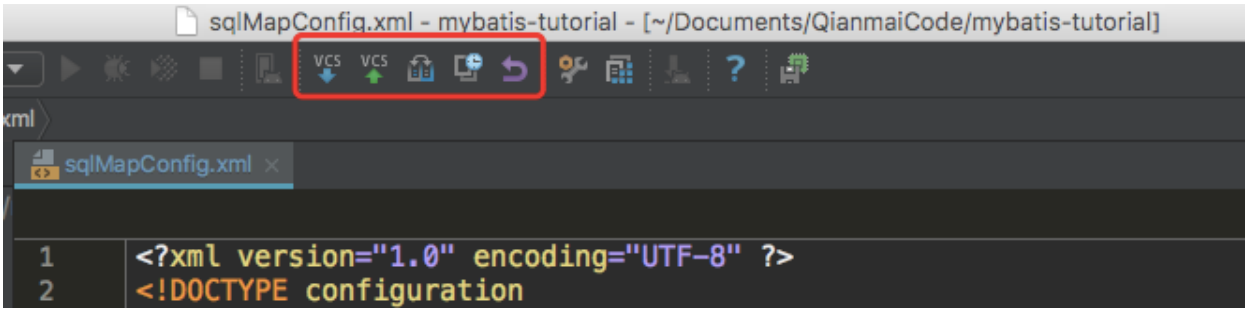
check

如上图所示，点击Clone之后，提示我们对将要检出的项目进行确认，点击Yes，然后一路Next，最后点击Finish：



mybatis

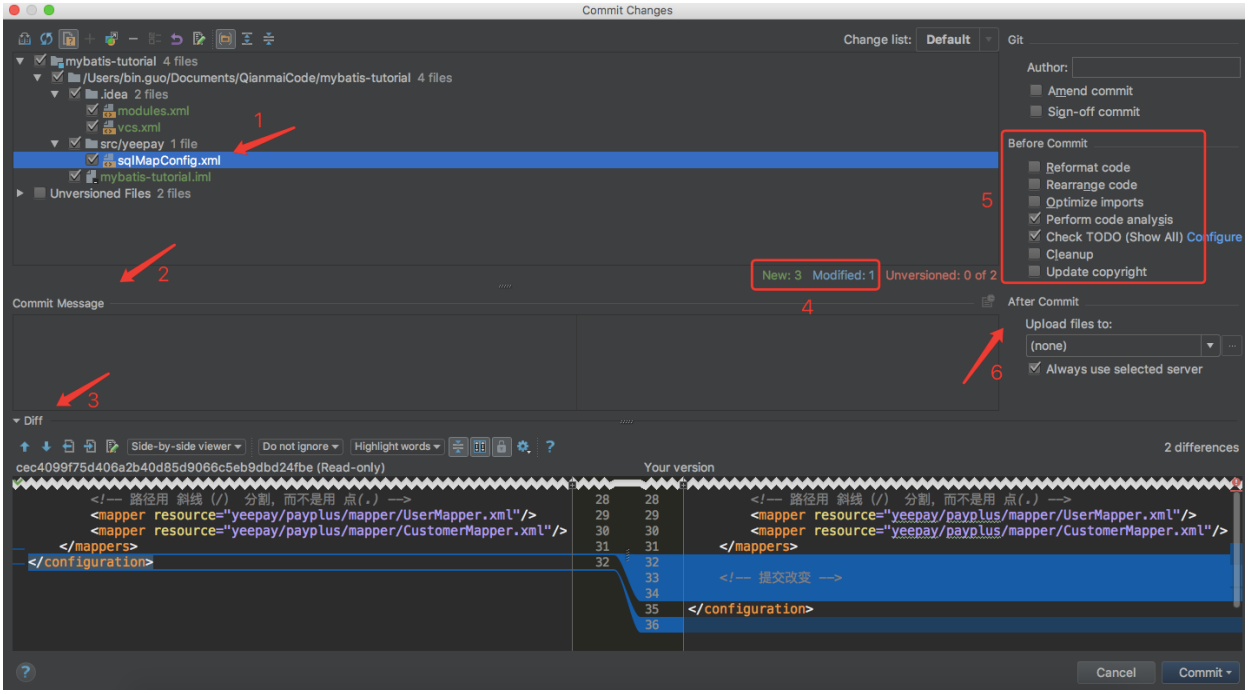
如上图所示，至此，项目`mybatis-tutorial`已经成功从 GitHub 检出到本地啦！



button

如上图红色标记所示，皆为进行版本控制的按钮，从左至右分别为：

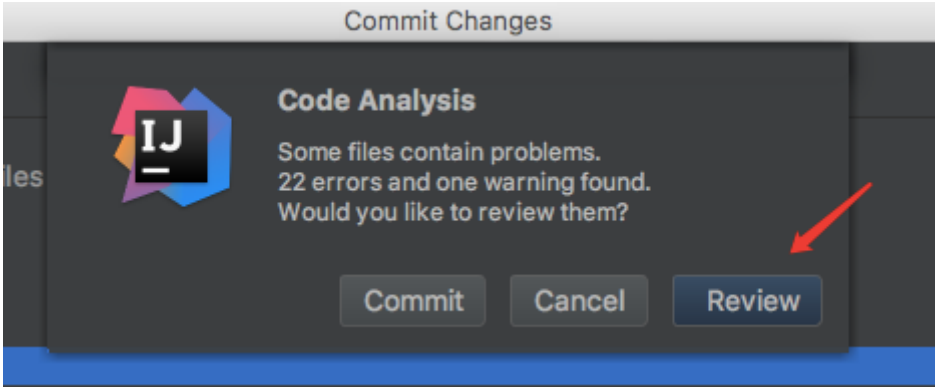
- `Update Project`，更新项目，即从检出仓库下载最新版本的代码；
- `Commit changes`，提交此检出版本项目上所有变化的文件；
- `Compare with the Same Repository Version`，比较当前文件与远程仓库版本文件之间的差异；
- `Show history`，显示当前文件的历史记录；
- `Revert`，还原当前被修改的文件到未被修改的版本状态。



commit

- 标注 1：在检出项目中有过修改的文件；
- 标注 2：`Comiit Messsage`提交信息，需要我们自己填写；
- 标注 3：`Diff`，展示文件修改前后对比；
- 标注 4：展示修改了几个文件，新建了几个文件；
- 标注 5：`Before Commit`，在提交项目前，进行一些前置操作；
- 标注 6：`After Commit`，在提交项目后，进行一些后置操作。

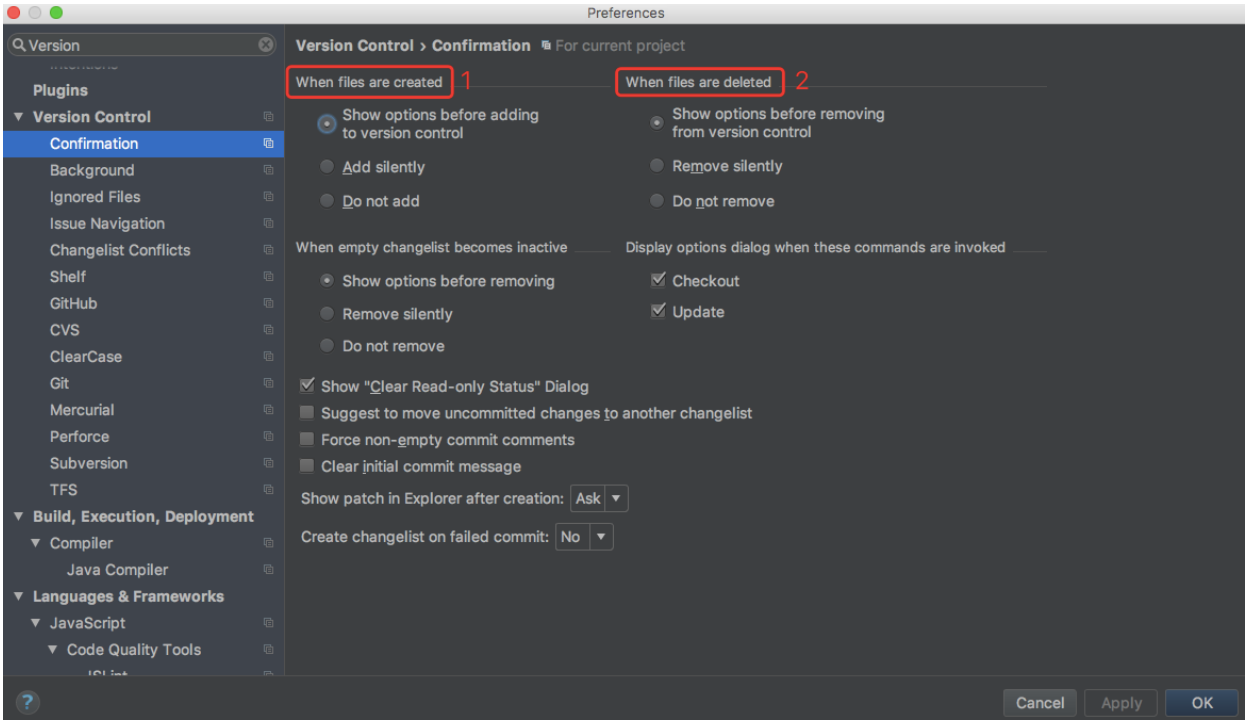
其中，`Diff`展示了文件修改前后详细的对比，我们需要好好利用；`Before Commit`，默认进行提交前的代码分析，可以检查出一些错误与警告。此外，我们也可以通过双击 **标注1** 所示的文件，放大文件修改前后的差异对比。接下来，点击`Comiit`进行验证：



Code

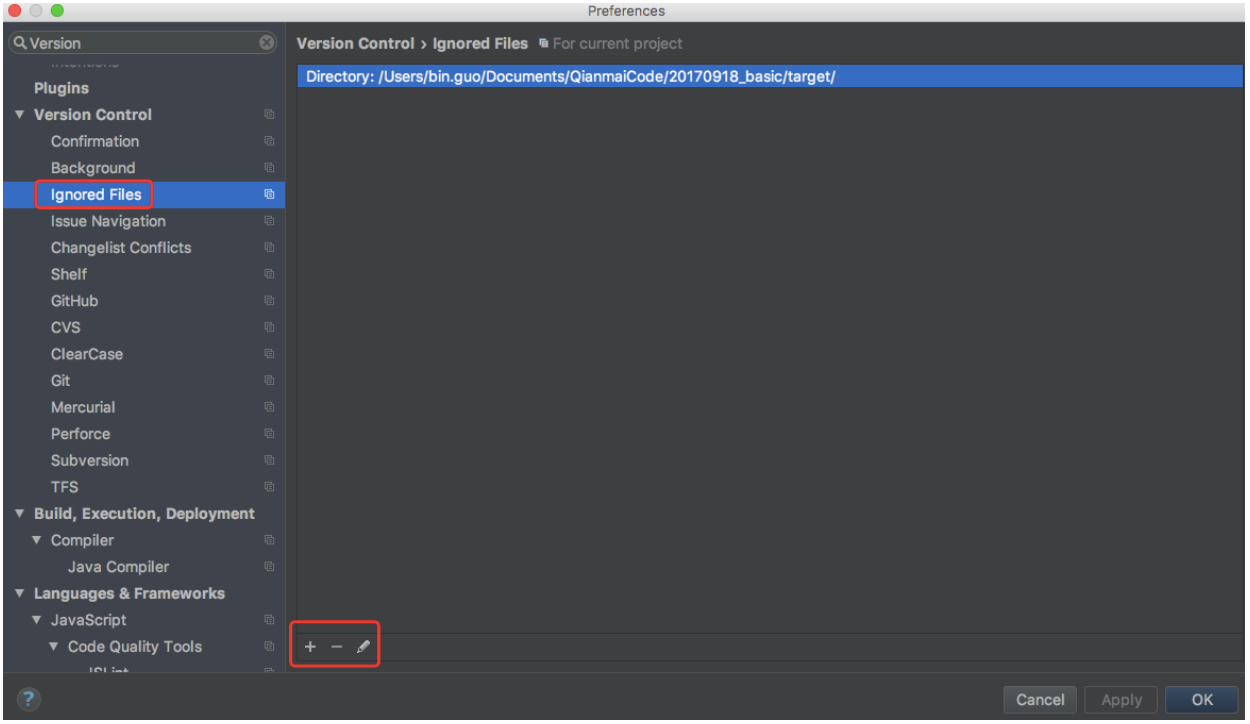
如上图所示，显示了代码分析的结果，具体可以参考「详述 IntelliJ IDEA 提交代码前的 Code Analysis 机制」.

最后，我们再回到`Version Control`，了解一些常用的操作：



config

- 标注1 ：`When files are created`，表示当有新文件放进项目中的时候 IntelliJ IDEA 做如何处理，默认是 Show options before adding to version control，表示弹出提示选项，让我们自己决定是否将这些新文件加入到版本控制。如果不想弹出提示，则选择下面两个选项进行默认操作。
- 标注 2：`When files are deleted`，表示当有新文件在项目中被删除的时候 IntelliJ IDEA 做如何处理，默认是 Show options before removing from version control，表示弹出提示选项，让我们自己决定是否将这些被删除的文件从版本控制中删除。如果不想弹出提示，则选择下面两个选项进行默认操作。



ignore

如上图所示，我们可以通过红色标记圈出的+，把不想加入版本控制的文件或目录添加到忽略列表中；反之，我们也可以通过红色标记圈出的-，把想加入版本控制的文件或目录从忽略列表中移除。在这里，我们需要注意：**当某文件或目录被添加到此“忽略列表”的之后，则该文件或目录不能进行版本控制的相关操作，例如提交。**