

Eclipse+Github+"hello world" (简单清楚)

Eclipse+Github+"hello world" (简单清楚)

下载链接

安装

Github仓库

创建一个GitHub仓库

GitHub客户端

Eclipse

安装EGit插件

创建hello world代码

创建本地仓库

push到GitHub的仓库

由GitHub上传

下载链接

- Eclipse下载及安装: [Eclipse Downloads | The Eclipse Foundation](#)
- Github账号注册: [GitHub](#)
- Github客户端下载: [GitHub Desktop | Simple collaboration from your desktop](#)
- Java下载jdk: [Java Archive Downloads - Java SE 8 \(oracle.com\)](#)

安装

1. 将[jdk](#)、[Eclipse](#)、[Github客户端](#)下载下来
2. 全默认无脑下一步安装完成

Github仓库

创建一个GitHub仓库

1. 点击 `Create a new repository`

取一个**项目名**, **点击** `Create repository`, 仓库建立完毕

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *



Star-tears ▾

Repository name *

project_name

Great repository names are short, lowercase, and unique. **project_name is available.** Need inspiration? How about **musical-guacamole?**

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

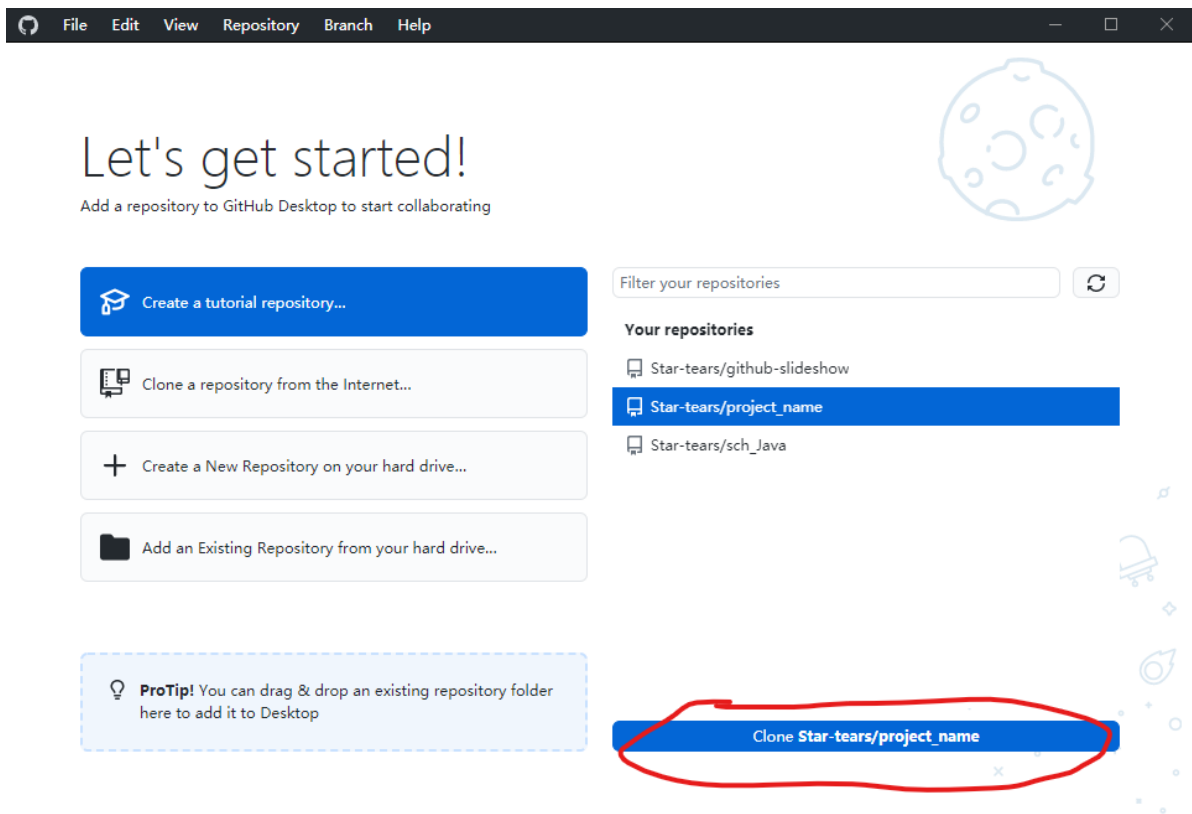
☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

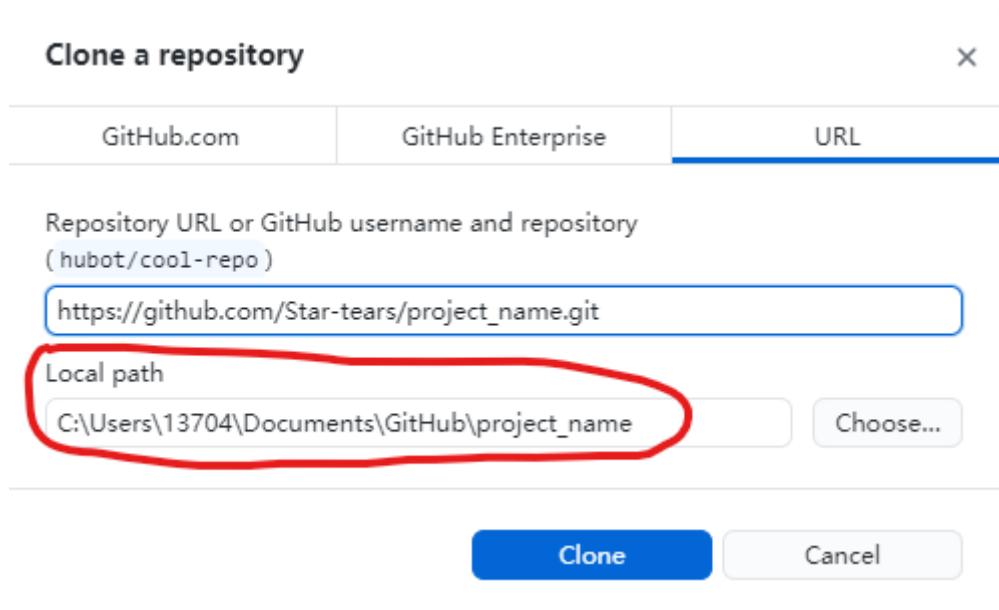
Create repository

GitHub客户端

登录, 选择你的仓库并克隆仓库到本地



记一下仓库的 local path，后面要用到



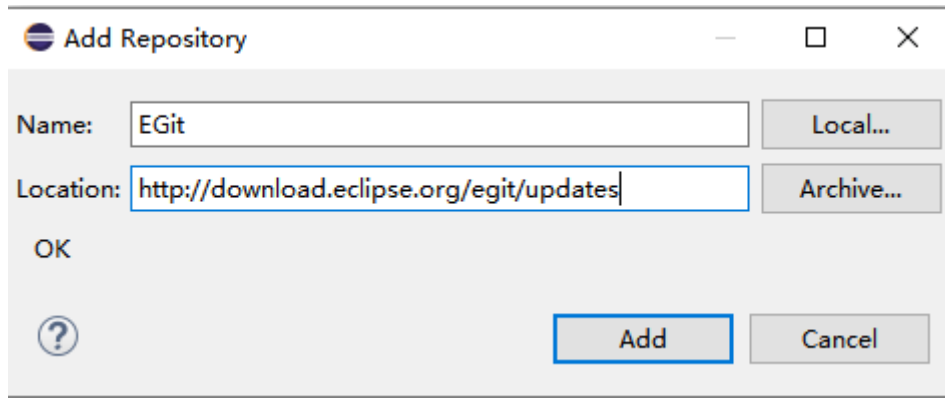
Eclipse

安装EGit插件

插件地址: <http://download.eclipse.org/egit/updates>

1. Help -> Install New Software... -> Add

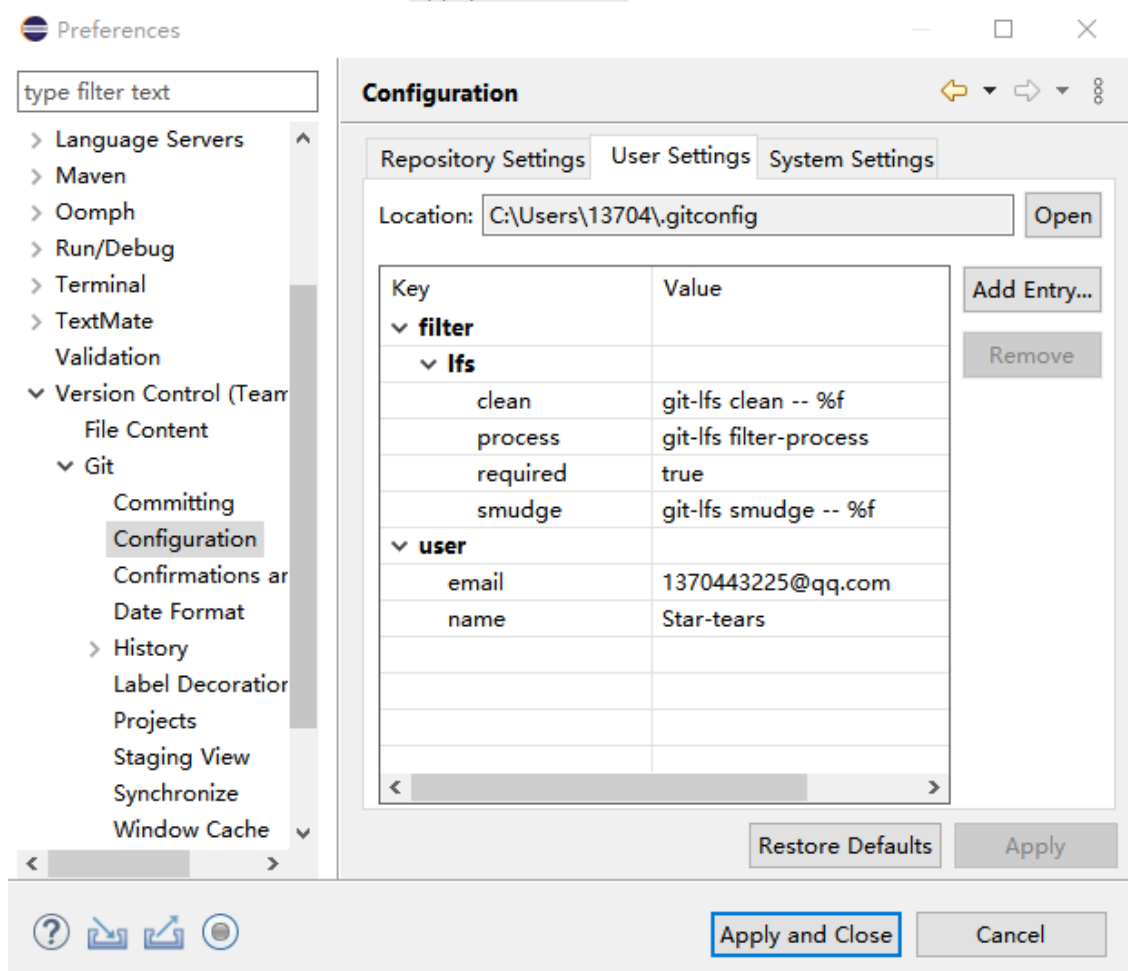
2. 输入地址到Location



3. 全选安装, 重启Eclipse

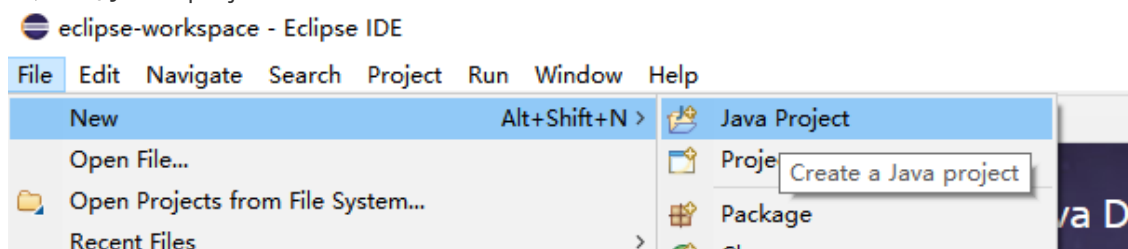
4. windows -> Preferences -> Version Control(Team) -> Git -> Configuration -> Add Entry

5. 添加自己的GitHub邮箱、用户名, Apply and Close



创建hello world代码

1. 创建一个Java的project



2. 取一个名字, finish

The screenshot shows the 'New Java Project' dialog box in Eclipse. The title bar says 'New Java Project'. Below the title bar, it says 'Create a Java Project' and 'Create a Java project in the workspace or in an external location.' There is a folder icon in the top right corner.

The 'Project name:' field contains the text 'hello'.

The 'Use default location' checkbox is checked. The 'Location:' field shows 'F:\eclipse-workspace\hello' and a 'Browse...' button is next to it.

The 'JRE' section has three radio buttons: 'Use an execution environment JRE:' (selected), 'Use a project specific JRE:', and 'Use default JRE 'jdk-11.0.11' and workspace compiler preferences'. The first option has a dropdown menu showing 'JavaSE-11'. The second option has a dropdown menu showing 'jdk-11.0.11'. There is a 'Configure JREs...' link next to the third option.

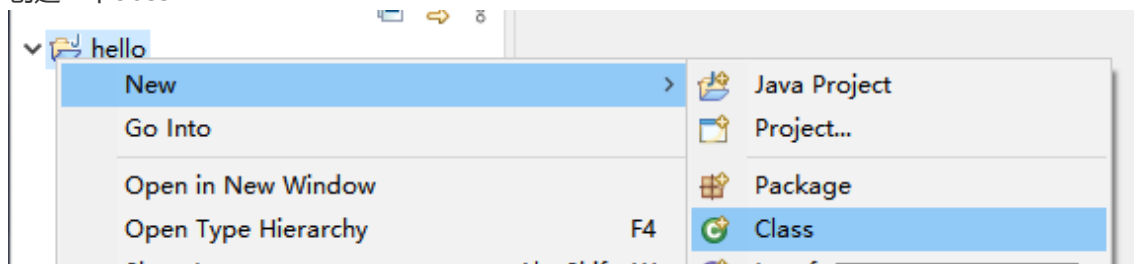
The 'Project layout' section has two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected). There is a 'Configure default...' link next to the second option.

The 'Working sets' section has a checkbox 'Add project to working sets' which is unchecked. There is a 'New...' button next to it. Below the checkbox is a 'Working sets:' dropdown menu and a 'Select...' button.

The 'Module' section has a checkbox 'Create module-info.java file' which is unchecked.

At the bottom, there is a '?' icon, a '< Back' button, a 'Next >' button, a 'Finish' button (highlighted with a blue border), and a 'Cancel' button.

3. 创建一个class



4. 取一个名字, finish

New Java Class

Java Class

⚠ Type name is discouraged. By convention, Java type names usually start with an uppercase letter

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☒ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

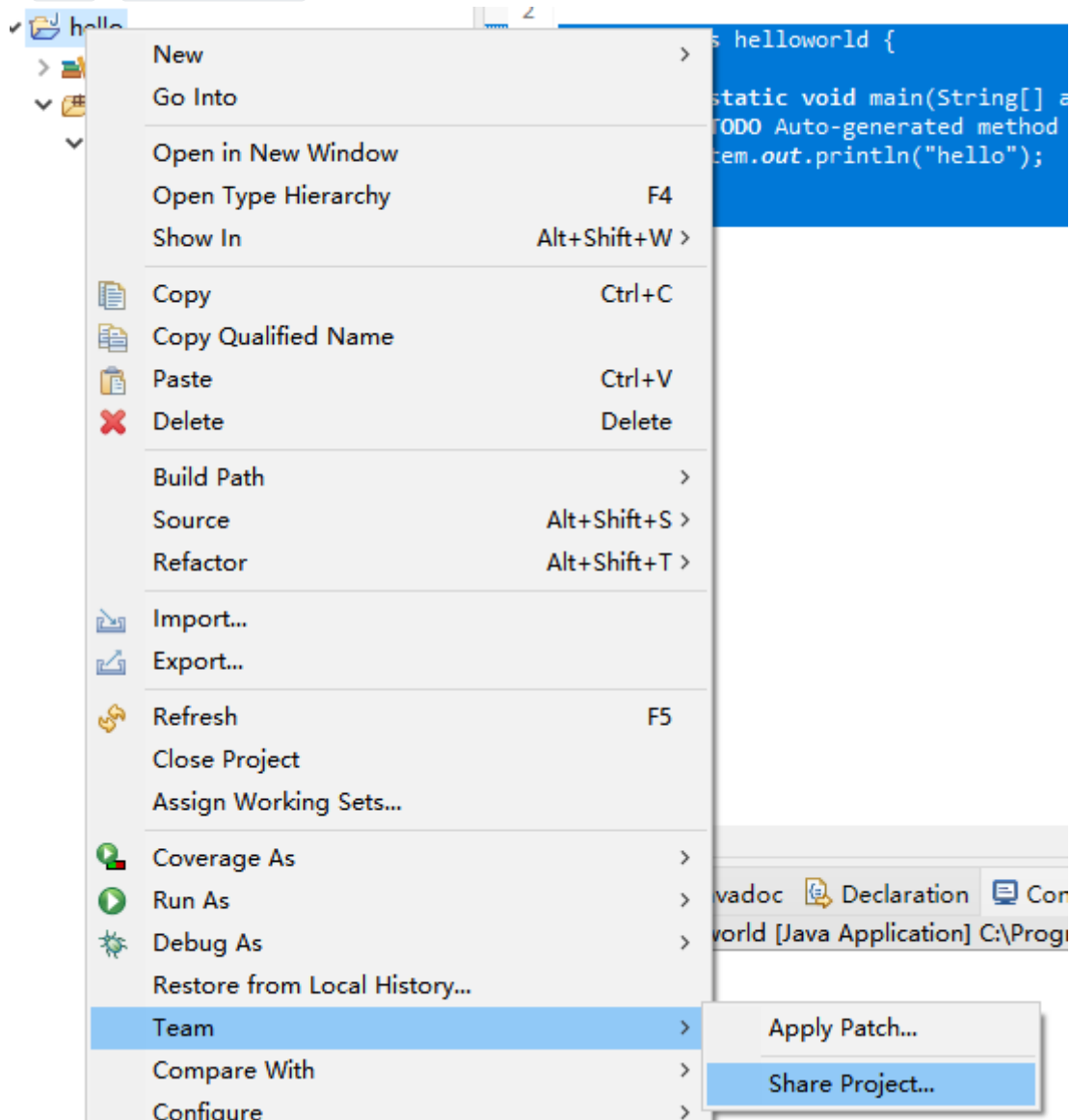
☐ Generate comments

5. hello world代码

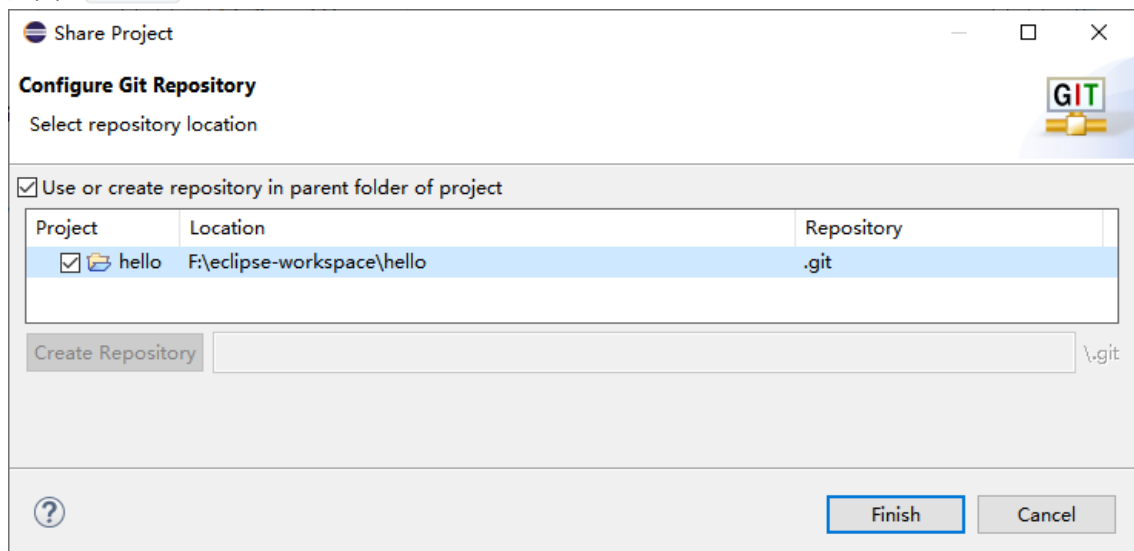
```
public class helloworld {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println("hello");  
    }  
}
```

创建本地仓库

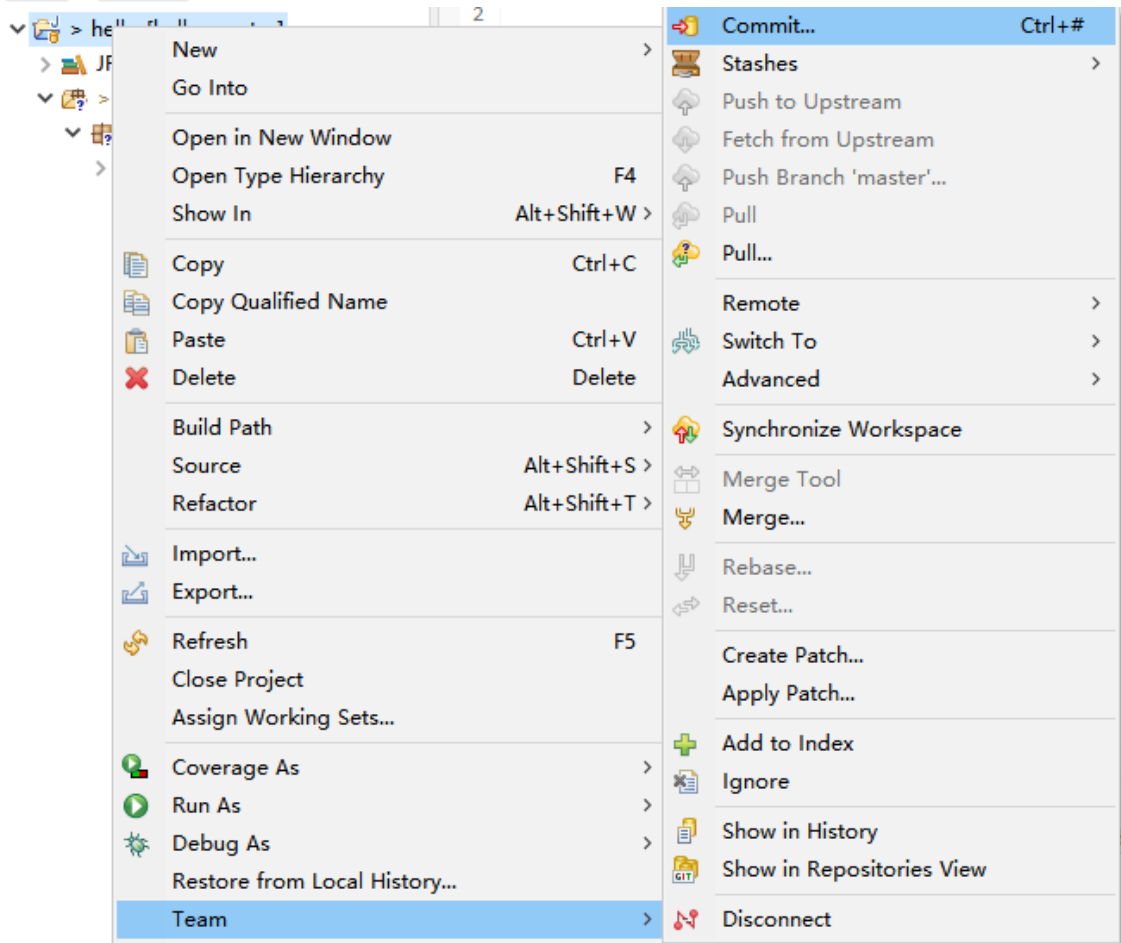
1. Team->Share project



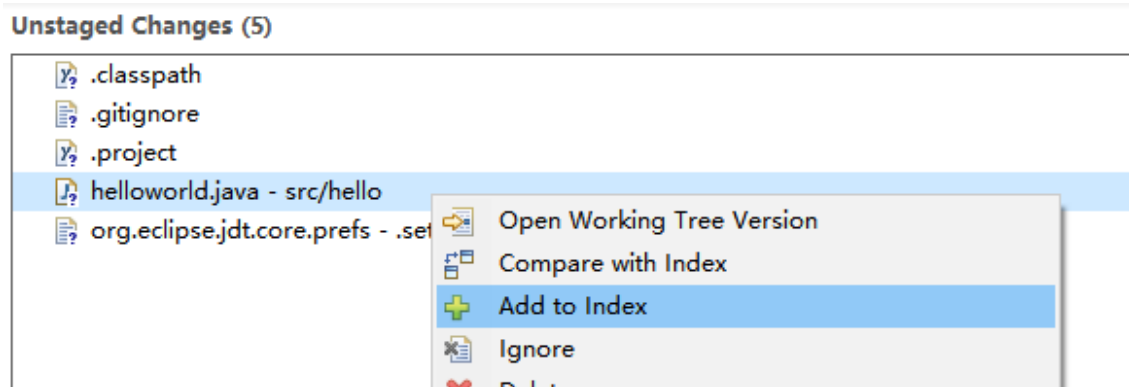
2. 如图-> finish



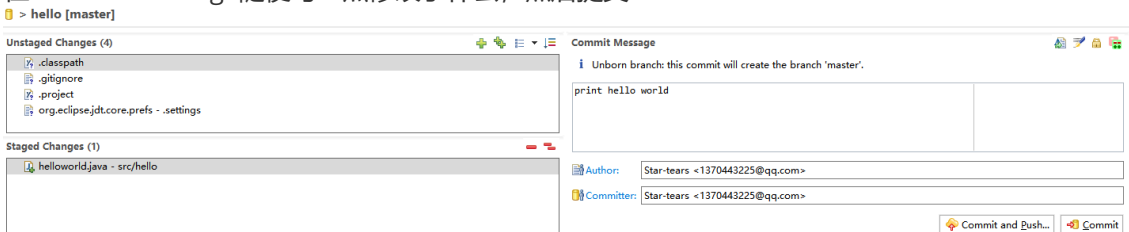
3. Team -> Commit



4. 将需要提交文件加入Index

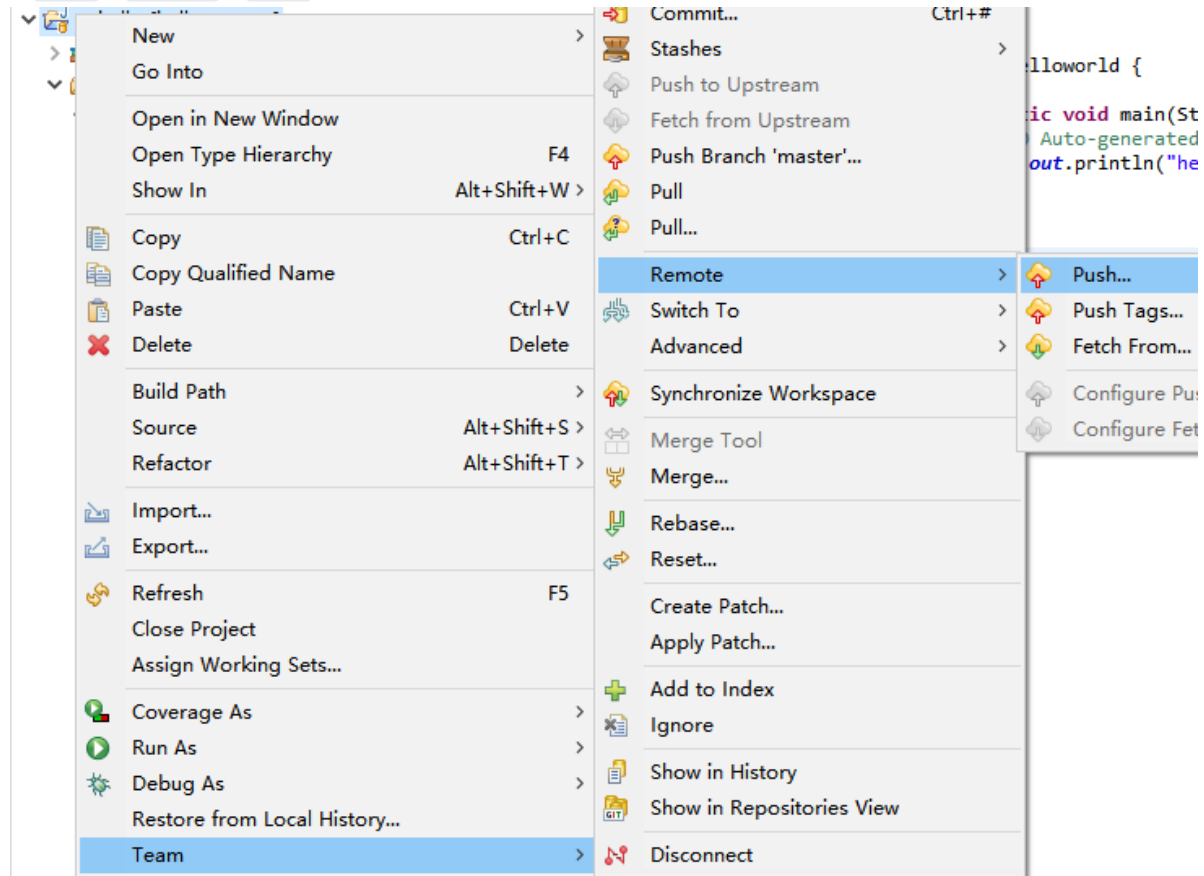


5. 在Commit Message随便写一点修改了什么，然后提交



push到GitHub的仓库

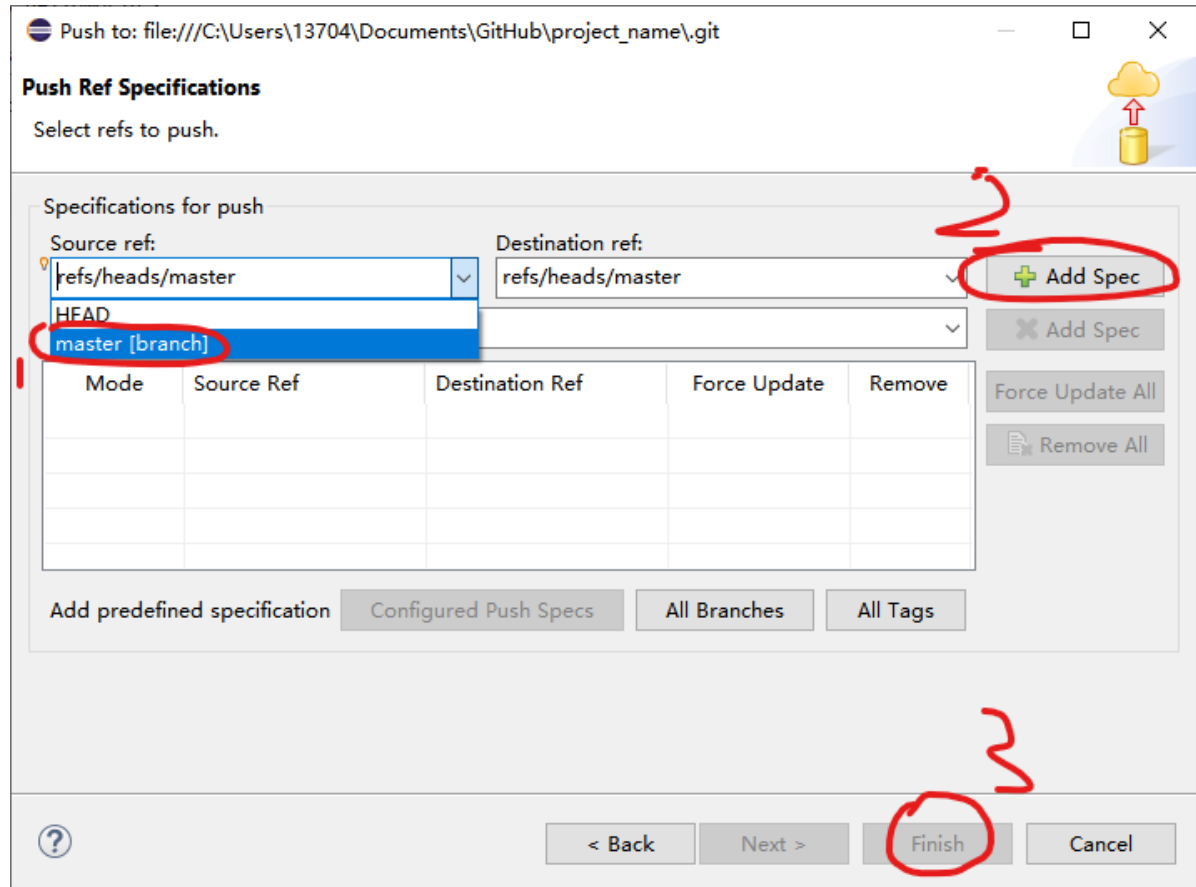
1. Team -> Remote -> Push



2. 用之前的Local Path（就是克隆到本地的GitHub仓库）填入

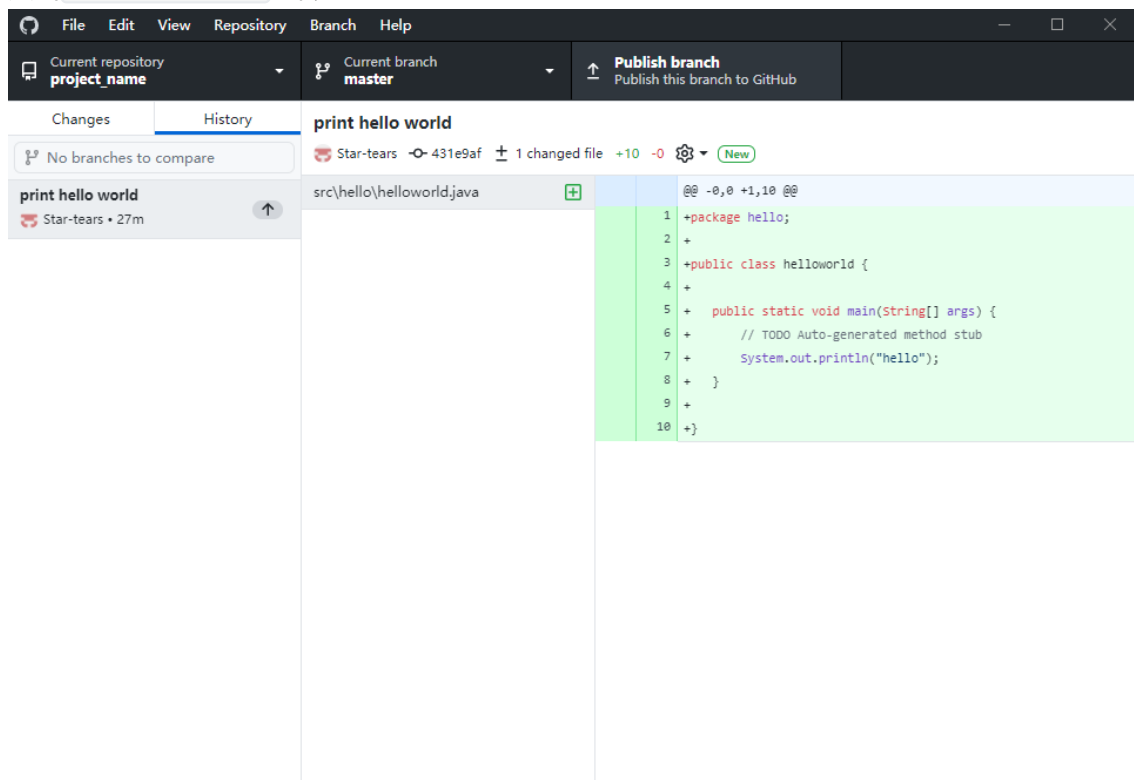
A screenshot of the 'Push to Another Repository' dialog box in IntelliJ IDEA. The dialog has a title bar with a question mark icon and standard window controls. Below the title, it says 'Destination Git Repository' and 'Enter the location of the destination repository.' There is a cloud icon with an upward arrow. The 'Location' section contains three text fields: 'URI' with the value 'file:///C:/Users/13704/Documents/GitHub/project_name/.git', 'Host' (empty), and 'Repository path' with the value 'C:/Users/13704/Documents/GitHub/project_name/.git'. There is a 'Local Folder...' button next to the URI field. The 'Connection' section has a 'Protocol' dropdown set to 'file' and an empty 'Port' field. The 'Authentication' section has 'User' and 'Password' text fields, and a checkbox labeled 'Store in Secure Store' which is unchecked. At the bottom, there are buttons for '?', '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted.

3.



由GitHub上传

1. 修改情况已经能从GitHub客户端查看
2. 点击 Publish branch 上传



3. GitHub上就有上传的代码啦

