

初识Spring Boot, 开发社区首页

牛客Java高级工程师 第一章

1. 课程介绍

课程目标

- 学会主流的web开发技术和框架
- 积累一个真实的web项目的开发经验
- 掌握热点面试题的答题策略

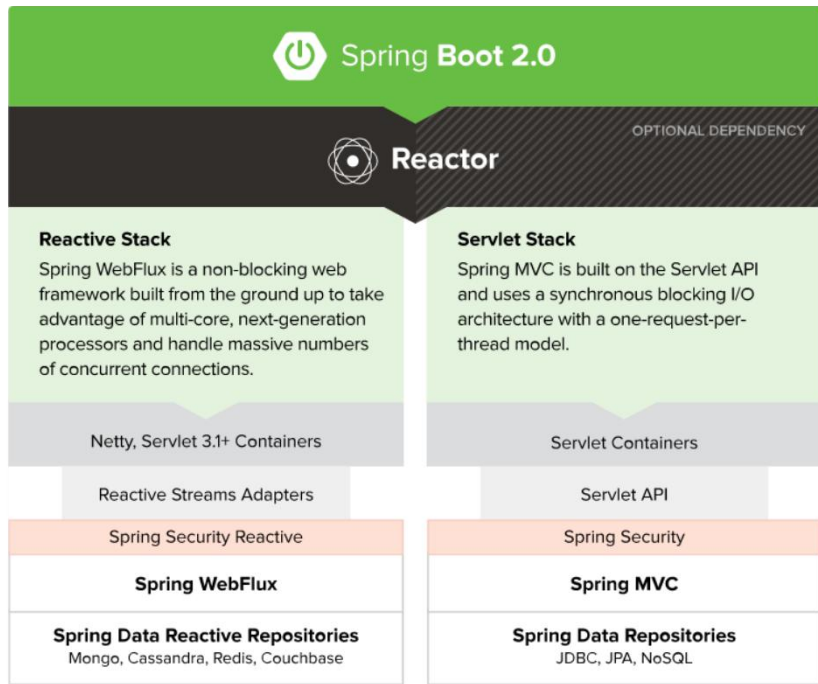
[/nowcoder.com/community/index.html](http://nowcoder.com/community/index.html)



► 1. 课程介绍

技术架构

- Spring Boot
- Spring、Spring MVC、MyBatis
- Redis、Kafka、Elasticsearch
- Spring Security、Spring Actuator



► 1. 课程介绍

开发环境

- 构建工具：Apache Maven
- 集成开发工具：IntelliJ IDEA
- 数据库：MySQL、Redis
- 应用服务器：Apache Tomcat
- 版本控制工具：Git



► 2. 搭建开发环境

Apache Maven

- 可以帮助我们构建项目、管理项目中的jar包
- Maven仓库：存放构件的位置
 - 本地仓库：默认是 `~/.m2/repository`
 - 远程仓库：中央仓库、镜像仓库、私服仓库
- 示例：安装、配置、常用命令

<http://maven.apache.org>

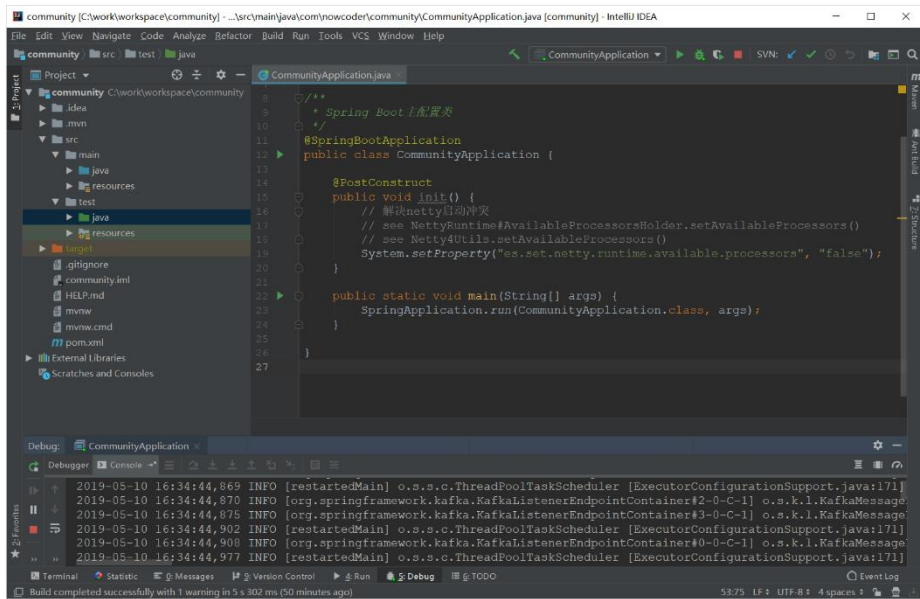
```
c:\work\workspace>mvn archetype:generate -DgroupId=com.nowcoder -DartifactId=mavende
[INFO] Scanning for projects...
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO] >>> maven-archetype-plugin:3.0.1:generate (default-cli) > generate-sources @
[INFO] <<< maven-archetype-plugin:3.0.1:generate (default-cli) < generate-sources @
[INFO] --- maven-archetype-plugin:3.0.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[WARNING] No archetype found in remote catalog. Defaulting to internal catalog
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.arch
[INFO] Using following parameters for creating project from Old (1.x) Archetype: mav
[INFO] Parameter: basedir, Value: c:\work\workspace
[INFO] Parameter: package, Value: com.nowcoder
[INFO] Parameter: groupId, Value: com.nowcoder
[INFO] Parameter: artifactId, Value: mavendemo3
[INFO] Parameter: packageName, Value: com.nowcoder
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: c:\work\workspace\mavendemo3
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.522 s
[INFO] Finished at: 2019-05-10T17:45:55+08:00
[INFO] -----
```

► 2. 搭建开发环境

IntelliJ IDEA

- 目前最流行的Java集成开发工具
- 示例：安装、配置、创建项目

<http://www.jetbrains.com/idea>

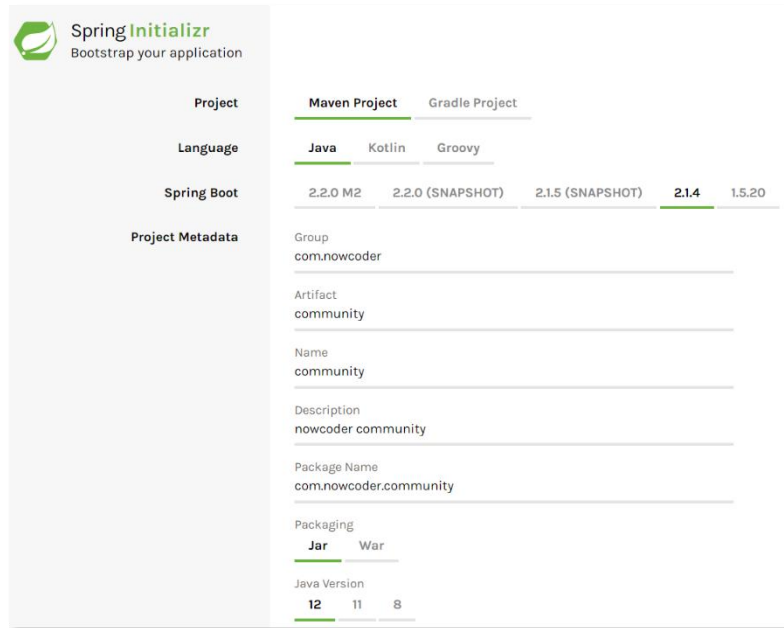


► 2. 搭建开发环境

Spring Initializr

- 创建 Spring Boot 项目的引导工具
- 示例：创建“牛客社区”项目

<https://start.spring.io>



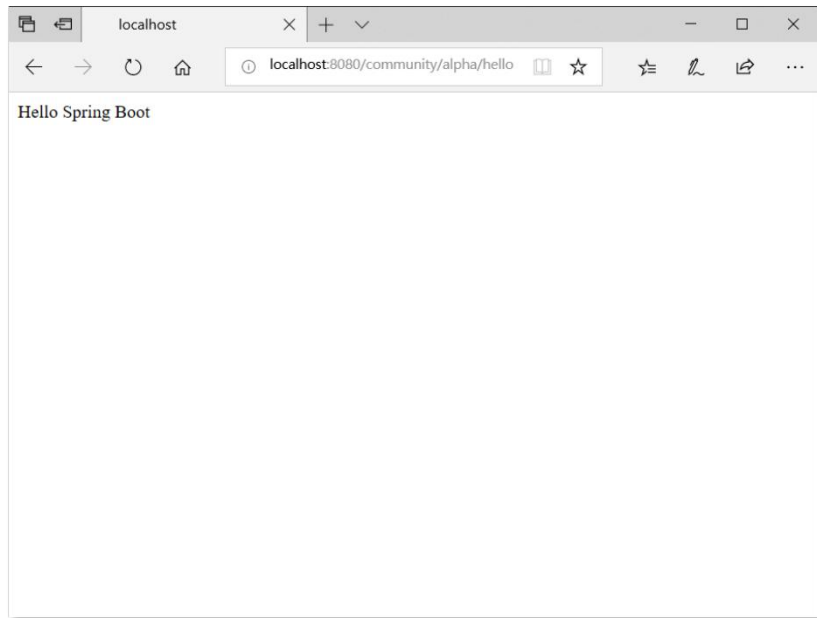
The image shows the Spring Initializr web interface. It has a sidebar on the left with navigation links: Project, Language, Spring Boot, and Project Metadata. The main area is titled 'Bootstrap your application' and contains several tabs and input fields. The 'Project' tab is active, showing 'Maven Project' and 'Gradle Project' options. The 'Language' tab is active, showing 'Java', 'Kotlin', and 'Groovy' options. The 'Spring Boot' tab is active, showing version options: '2.2.0 M2', '2.2.0 (SNAPSHOT)', '2.1.5 (SNAPSHOT)', '2.1.4' (selected), and '1.5.20'. The 'Project Metadata' section contains input fields for 'Group' (com.nowcoder), 'Artifact' (community), 'Name' (community), 'Description' (nowcoder community), 'Package Name' (com.nowcoder.community), 'Packaging' (Jar selected, War also available), and 'Java Version' (12 selected, 11 and 8 also available).

Project	Language	Spring Boot	Project Metadata
Maven Project	Java	2.2.0 M2, 2.2.0 (SNAPSHOT), 2.1.5 (SNAPSHOT), 2.1.4, 1.5.20	Group: com.nowcoder, Artifact: community, Name: community, Description: nowcoder community, Package Name: com.nowcoder.community, Packaging: Jar, War, Java Version: 12, 11, 8

► 2. 搭建开发环境

Spring Boot 入门示例

- Spring Boot 核心作用
 - 起步依赖、自动配置、端点监控
- 示例
 - 一个简单的处理客户端请求案例



► 3. Spring入门

Spring全家桶

- Spring Framework
- Spring Boot
- Spring Cloud
- Spring Cloud Data Flow

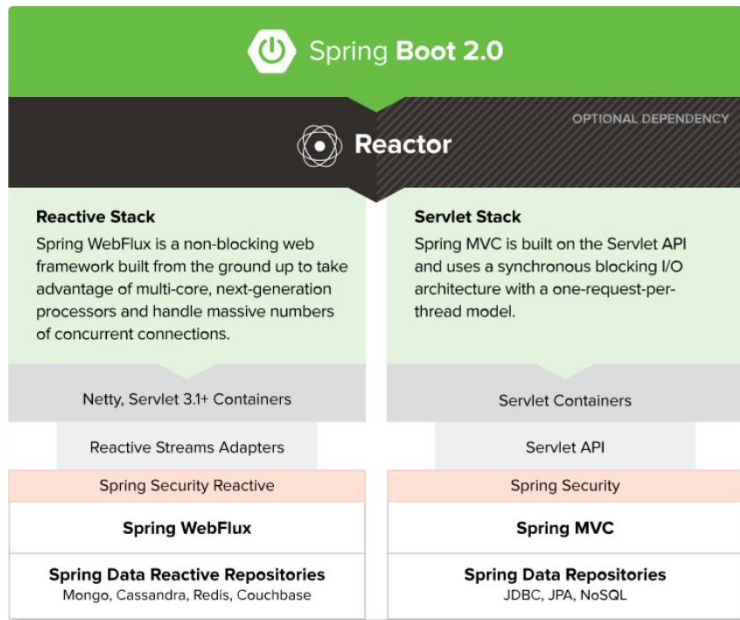


<https://spring.io>

► 3. Spring入门

Spring Framework

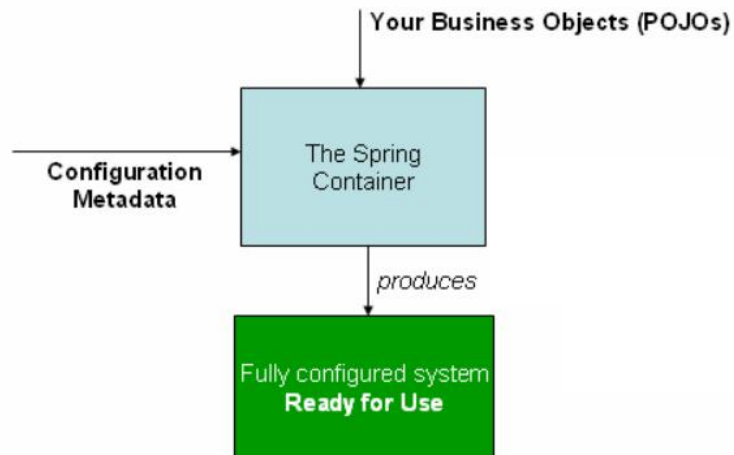
- Spring Core
 - IoC、AOP
- Spring Data Access
 - Transactions、Spring MyBatis
- Web Servlet
 - Spring MVC
- Integration
 - Email、Scheduling、AMQP、Security



► 3. Spring入门

Spring IoC

- Inversion of Control
 - **控制反转**，是一种面向对象编程的设计思想。
- Dependency Injection
 - **依赖注入**，是IoC思想的实现方式。
- IoC Container
 - **IoC容器**，是实现依赖注入的关键，本质上是一个工厂。



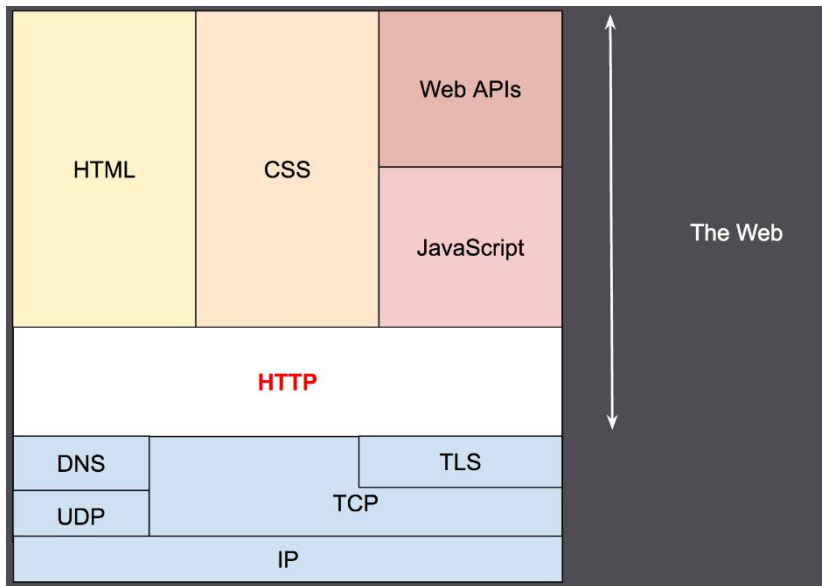
► 4. Spring MVC入门

HTTP

- HyperText Transfer Protocol
- 用于传输HTML等内容的应用层协议
- 规定了浏览器和服务端之间如何通信，以及通信时的数据格式。

<https://www.ietf.org>

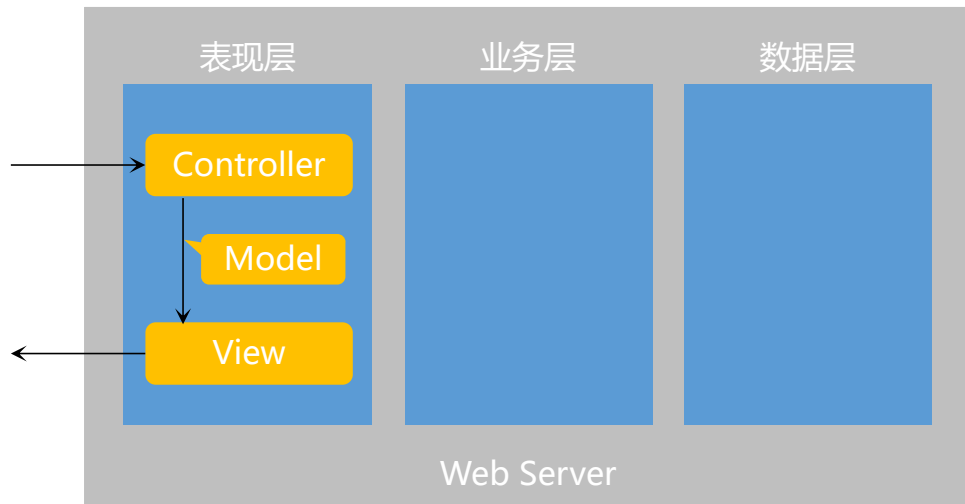
<https://developer.mozilla.org/zh-CN>



► 4. Spring MVC入门

Spring MVC

- 三层架构
 - 表现层、业务层、数据访问层
- MVC
 - Model: 模型层
 - View: 视图层
 - Controller: 控制层
- 核心组件
 - 前端控制器: DispatcherServlet



► 4. Spring MVC入门

Thymeleaf

- 模板引擎
 - 生成动态的HTML。
- Thymeleaf
 - 倡导自然模板，即以HTML文件为模板。
- 常用语法
 - 标准表达式、判断与循环、模板的布局。

<https://www.thymeleaf.org>

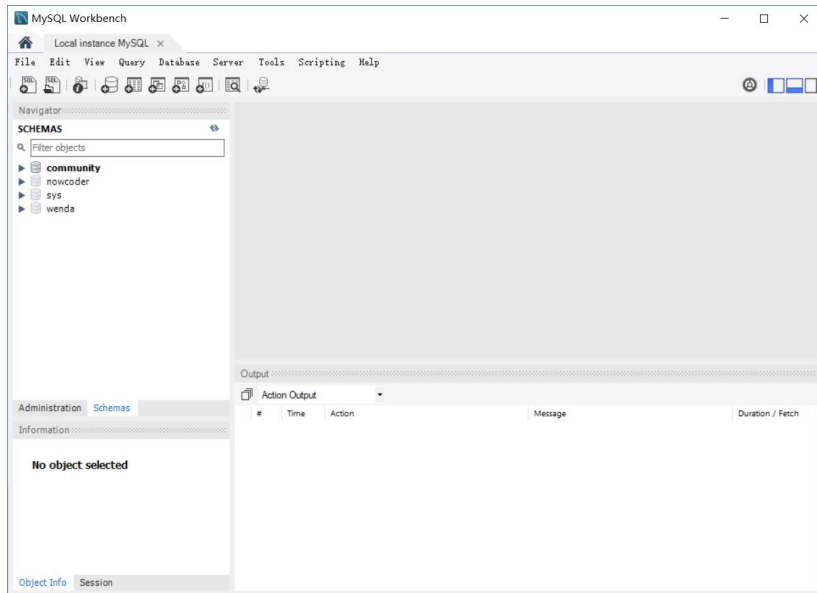


► 5. MyBatis入门

安装数据库

- 安装MySQL Server
- 安装MySQL Workbench

<https://dev.mysql.com/downloads/mysql>
<https://dev.mysql.com/downloads/workbench>



► 5. MyBatis入门

MyBatis



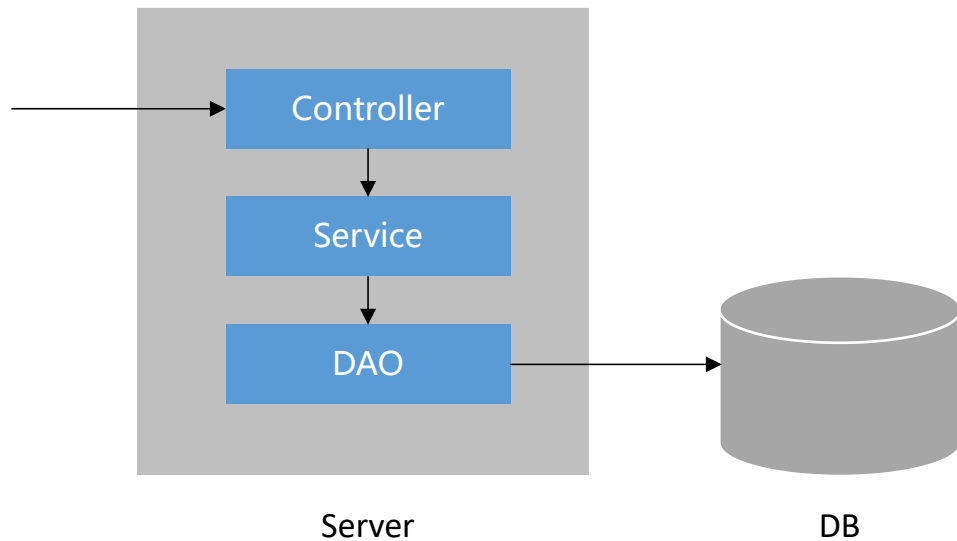
- 核心组件
 - SqlSessionFactory: 用于创建SqlSession的工厂类。
 - SqlSession: MyBatis的核心组件, 用于向数据库执行SQL。
 - 主配置文件: XML配置文件, 可以对MyBatis的底层行为做出详细的配置。
 - Mapper接口: 就是DAO接口, 在MyBatis中习惯性的称之为Mapper。
 - Mapper映射器: 用于编写SQL, 并将SQL和实体类映射的组件, 采用XML、注解均可实现。
- 示例
 - 使用MyBatis对用户表进行CRUD操作。

<http://www.mybatis.org/mybatis-3>

<http://www.mybatis.org/spring>

► 6. 开发社区首页

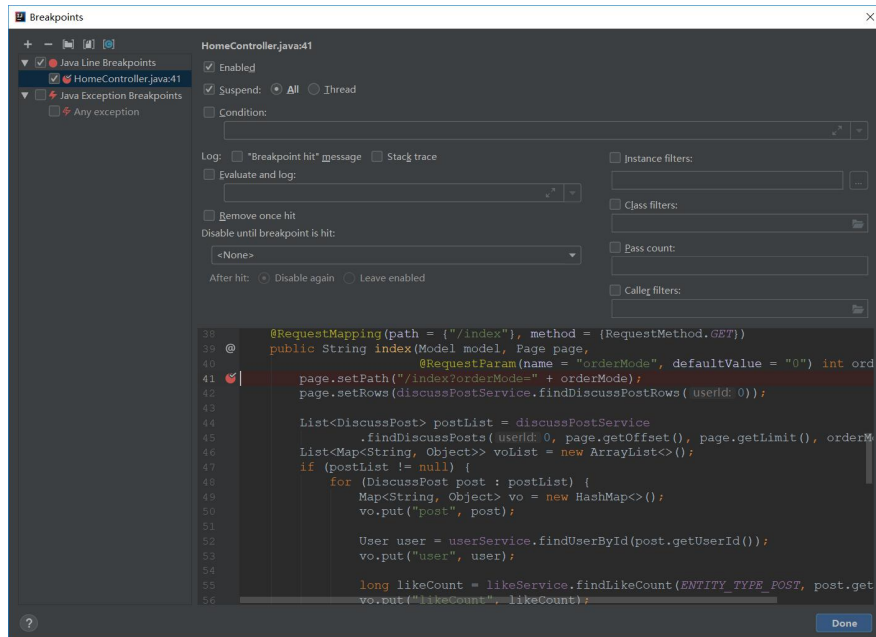
- 开发流程
 - 1次请求的执行过程
- 分步实现
 - 开发社区首页，显示前10个帖子
 - 开发分页组件，分页显示所有的帖子



► 7. 项目调试技巧

- 响应状态码的含义
- 服务端断点调试技巧
- 客户端断点调试技巧
- 设置日志级别，并将日志输出到不同的终端

<https://logback.qos.ch>



► 8. 版本控制

- 认识Git
 - Git简介
 - Git的安装与配置
- Git常用命令
 - 将代码提交至本地仓库
 - 将代码上传至远程仓库
- IDEA集成Git
 - 在IDEA中配置并使用Git

<https://git-scm.com>

```
# 账号配置
git config --list
git config --global user.name "lihonghe"
git config --global user.email "lihonghe@nowcoder.com"

# 本地仓库
git init
git status -s
git add *
git commit -m '...'

# 生成秘钥
ssh-keygen -t rsa -C "lihonghe@nowcoder.com"

# 推送已有项目
git remote add origin
https://git.nowcoder.com/334190970/Test.git
git push -u origin master

# 克隆已有仓库
git clone https://git.nowcoder.com/334190970/Test.git
```

Thanks

