

Week 1

环境搭建、IDEA基本用法

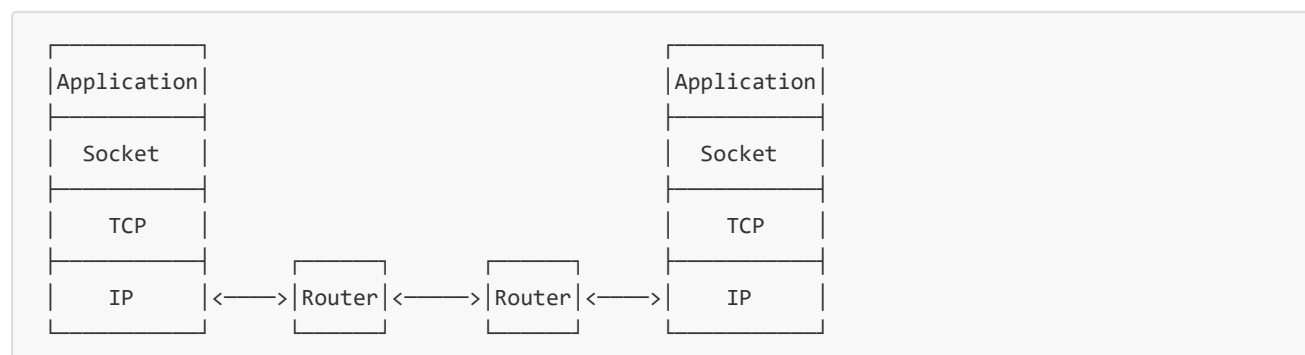
首先的话需要在自己的系统平台（Windows or MacOS）搭建好一系列的后端环境，并熟悉对应语言IDE的基本操作：

- Java JRE/JDK
- JetBrains IDEA Community（如果用户拥有Pro的License，那更好）
- Maven
- MySQL

TCPIP/HTTP通信基础

当今Web开发的软件框架已经完全将TCPIP/HTTP的基础内容封装在现成的API和方法内。不接触深度的web开发一般会在代码层面直接面对这些内容的处理。但是我们依旧建议复习/了解七层结构，并且熟记TCP的握手流程。

在开发网络应用程序的时候，我们又会遇到Socket这个概念。Socket是一个抽象概念，一个应用程序通过一个Socket来建立一个远程连接，而Socket内部通过TCP/IP协议把数据传输到网络：



Socket、TCP和部分IP的功能都是由操作系统提供的，不同的编程语言只是提供了对操作系统调用的简单的封装。例如，Java提供的几个Socket相关的类就封装了操作系统提供的接口。

为什么需要Socket进行网络通信？因为仅仅通过IP地址进行通信是不够的，同一台计算机同一时间会运行多个网络应用程序，例如浏览器、QQ、邮件客户端等。当操作系统接收到一个数据包的时候，如果只有IP地址，它没法判断应该发给哪个应用程序，所以，操作系统抽象出Socket接口，每个应用程序需要各自对应到不同的Socket，数据包才能根据Socket正确地发到对应的应用程序。

一个Socket就是由IP地址和端口号（范围是0~65535）组成，可以把Socket简单理解为IP地址加端口号。端口号总是由操作系统分配，它是一个0~65535之间的数字，其中，小于1024的端口属于特权端口，需要管理员权限，大于1024的端口可以由任意用户的应用程序打开。

- 101.202.99.2:1201
- 101.202.99.2:1304
- 101.202.99.2:15000

使用Socket进行网络编程时，本质上就是两个进程之间的网络通信。其中一个进程必须充当服务器端，它会主动监听某个指定的端口，另一个进程必须充当客户端，它必须主动连接服务器的IP地址和指定端口，如果连接成功，服务器端和客户端就成功地建立了一个TCP连接，双方后续就可以随时发送和接收数据。

USER

因此，当Socket连接成功地在服务器端和客户端之间建立后：

- 对服务器端来说，它的Socket是指定的IP地址和指定的端口号；
- 对客户端来说，它的Socket是它所在计算机的IP地址和一个由操作系统分配的随机端口号。

任务要求

设计一个基于最基本Socket的P2P“聊天软件”（文本互传），要求在同一个网关内可以通过任意常用端口建立会话并进行socket通讯、双向文本收发。

SQL入门

在这里收录一些SQL基础知识的参考链接：

数据库的基础知识，以及为什么需要数据库：<https://www.jianshu.com/p/9e8d8f45c07d>

数据库核心概念（范式、事务、视图、表）：<https://zhuanlan.zhihu.com/p/42925724>

介绍RDBMS术语：<https://www.runoob.com/mysql/mysql-tutorial.html>

介绍MySQL的基本数据类型：<https://pdai.tech/md/db/sql-mysql/sql-mysql-theory.html>

介绍通过SSL Terminal连接MySQL的方法

介绍MySQL创建、删除、选择Database的方法：<https://www.runoob.com/mysql/mysql-create-database.html>

<https://www.runoob.com/mysql/mysql-drop-database.html>

<https://www.runoob.com/mysql/mysql-select-database.html>

介绍RDBMS术语：<https://www.runoob.com/mysql/mysql-tutorial.html>

介绍MySQL的基本数据类型：<https://pdai.tech/md/db/sql-mysql/sql-mysql-theory.html>

介绍通过SSL Terminal连接MySQL的方法

介绍MySQL创建、删除、选择Database的方法：<https://www.runoob.com/mysql/mysql-create-database.html>

<https://www.runoob.com/mysql/mysql-drop-database.html>

<https://www.runoob.com/mysql/mysql-select-database.html>

任务要求

在这里，需要设计一套数据库表结构。

其中必须包含的表有：用户表、内容表、评论表、广告表。

用户表需要记录用户的基本信息，例如：uid、绑定的手机号、头像、昵称、收藏列表、新闻偏好、浏览历史记录、评论历史记录、最后一次登录时间等。

内容表用来记录门户新闻内容。必须包含以下字段：新闻id、标题、封面图、投放日期时间、作者、新闻正文、浏览量、收藏量、转发量、付费推广标记等。

评论表用来记录新闻下的评论内容，必须包含以下字段：用户id、评论id、新闻id、评论内容、评论时间、引用评论id、点赞数、点踩数等。

广告是平台营收的最重要来源之一。该表用来记录平台各处投放的广告内容，必须包含以下字段：标题、文本内容、图像链接、投放位置（splash、弹窗、首页列表内嵌广告、评论内嵌广告、新闻文章内嵌广告、新闻文章末尾广告、相关新闻推荐内嵌广告）、投放方（甲方）标识符、投放有效期、地区限制、设备限制、点击量、曝光量、投放权重等。

数据类型和元素名需要自行判断，并且可以参考互联网上成熟的设计方案进行设计。

进阶：评论表的内容适合使用非关系型数据库进行存储，推荐同时使用MongoDB进行配合维护。

SQL指令

排序: <https://www.runoob.com/mysql/mysql-order-by.html>

分组: <https://www.runoob.com/mysql/mysql-group-by-statement.html>

连接: <https://www.runoob.com/mysql/mysql-join.html>

空处理: <https://www.runoob.com/mysql/mysql-null.html>

Alter: <https://www.runoob.com/mysql/mysql-alter.html>

索引: <https://www.runoob.com/mysql/mysql-index.html>

重复处理: <https://www.runoob.com/mysql/mysql-handling-duplicates.html>

运算符: <https://www.runoob.com/mysql/mysql-operator.html>

任务1

我们要针对上面的新闻门户数据表设计CRUD SQL指令，例如接下来的几个经典业务场景：

1. 通过邮箱创建用户
2. 用户编辑自己的昵称等账号信息
3. 用户登陆
4. 用户获取主页新闻列表
5. 用户按分区浏览新闻列表
6. 用户按标题和内容搜索新闻列表
7. 用户访问新闻，并且加载新闻下对应的评论
8. 用户给新闻进行收藏、分享
9. 用户对新闻评论
10. 用户对评论进行点赞/点踩
11. 用户删除自己的评论
12. 用户在个人页中观看自己的评论历史
13. 用户在个人页中观看自己的浏览历史
14. （模拟）投放新闻
15. 按照内容类型获取广告
16. 投放广告

任务2

我们的后端构建在Java的基础上，所以Java的数据库驱动名为jdbc，具体可以通过以下资料了解：

<https://www.yiibai.com/jdbc/>

将上一任务的SQL指令迁移到JDBC实现，运行jar包并观察数据库处理结果。