## 1.分析需求

1. 注册新用户
2. 登录已有用户
3. 显示博客列表，每一项包含博客标题，作者，（分布时间，博客类型，分页功能暂不考虑）
4. 点击标题跳转到文章详情页
5. 文章详情页中，显示文章标题，作者，内容（评论功能，分类功能，推荐相关文章，暂不考虑）
6. 发布新文章（不考虑富文本编辑）
7. 删除博客

## 2.创建项目——maven webapp

加入依赖，Servlet API，MySQL Connector

## 3.数据库设计

考虑当前代码中要用到几张表，每张表有哪些字段，那些字段之间有关联关系

1. 提取需求中的实体有那些，实体既关键性名词
2. 实体需要那些属性（用户，文章/博客），就确定了两张表各需要那些字段.

User用户：userId用户Id，name名字，password密码

Article文章：articleId文章Id，title标题，content内容，userId用户Id

1. 考虑实体之间的关系，一对一，一对多，多对多

User表中的userId表示当前用户的标识

Article表中的userId表示当前博客时那个用户发布的，一个用户可以发布多个博客，它们之间的关系是多对多。

## 代码设计

1. 实现数据库的一些操作

a).创建一个类，实现JDBC来管理数据库连接

b).创建实体类User，Article

c).实现数据库的基本增删改查UserdDao，ArticleDao

1. 对UserDao类和ArticleDao类进行一些单元测试

## 前后端接口设计

约定——服务器都能接收啥样的请求，每个请求的具体格式如何，对应的响应结果又是如何。是一种自定义协议（基于HTTP的基础上进行扩展）

1. 获取注册页面

请求：GET/register.html

响应：注册页面，页面上有两个输入框（输入用户名和密码），一个提交按钮

1. 实现注册功能

请求：POST/register

约定格式：name=dwt&password=123456

响应：返回一个提示页面，提示用户注册成功还是失败，并能跳转到登录页面

1. 获取登录页面

请求：GET/login.html

约定格式：name=dwt&password=123456

响应：返回一个登录页面，包含两个输入框和一个提交按钮

1. 实现登录功能

请求：POST/login

响应：返回一个提示页面，告诉用户是登录成功还是登陆失败

1. 获取文章列表

请求：GET/article

响应：返回文章详情页（包含文章标题，点击标题，进入详情页）

1. 获取文章详情页

请求：GET/article?articleId=1

响应：返回文章的详情页（包含文章内容）

1. 新增/发布文章

请求：POST/article

Title=xxx&content=xxxxx

响应：返回一个提示页面，告诉用户发布成功还是失败

1. 删除文章

请求：GET/deleteArticle?articleId=1

响应：返回提示页面，提示用户删除成功还是失败

## 6.实现接口