

项目逻辑

我们从搭建好的spring框架说起

webconfig

webconfig类里边写了我们访问网络的路径，创建了网页实例化，并赋予了网络地址。我们认为是

<http://localhost:9091/api/> 后面跟着我们的具体要访问的方法，每个方法返回的是一个我们从数据库里查表得到的结果

Stackoverflowthread

这个类是我们后端和数据库交互的类，也就是在java中为数据库规范一个表格的内容，并写出getter和setter

目前我们只有这一个表，如果需要其他的表格，我们可以后加

ThreadRepository

- `ThreadRepository`：这是你定义的接口，通常用于访问与 `StackoverflowThread` 实体相关的数据库表。
- `JpaRepository<StackoverflowThread, Long>`：`JpaRepository` 是 Spring Data JPA 提供的一个接口，它定义了与数据库交互的方法，常用于实现 CRUD（增、删、改、查）操作。
 - `StackoverflowThread`：实体类，表示数据库中的 `stackoverflow_thread` 表。每个 `StackoverflowThread` 对象代表一行数据。
 - `Long`：表示 `StackoverflowThread` 实体的主键类型，这里的主键是 `Long` 类型。

也就是说，我们通过 `JpaRepository` 接口，连接了我们的数据库，可以从写好的接口里调用方法对数据库查询。

DemoApplication

启动我们项目的类，可以认为是启动了整个网络服务器，init类可以用来使用我们的方法收集数据等

接下来是最重要的两个类：

Dataservice

这个类实现了我们想要与数据库交互的方法：如增删改查在这里的方法实现了与数据库交互后返回结果。

DemoController

这个类是我们与前端的交互，通过getmapping接口，让我们在访问网页的时候自动启动类和方法，返回给html数据随后显示

这个类连接到了我们的index.html文件

```
fetch('http://localhost:9091/api/topic-frequency')
```

这个在index中表示访问这个网页，调用这个方法便可以接受response后根据收到的数据进行绘图渲染。