手把手的SpringBoot教程，SpringBoot创建web项目（五）

这一节，我们来演示如何在SpringBoot项目中连接数据库，并且自动创建一张表。

按照惯例，数据库我们依然使用mysql，至于什么是jpa呢？

jpa是sun推出的持久化规范（java persistens api），JPA通过JDK 5.0注解或XML描述对象－关系表的映射关系，并将运行期的实体对象持久化到数据库中。JPA 的目标之一是制定一个可以由很多供应商实现的API，并且开发人员可以编码来实现该API，而不是使用私有供应商特有的API。

实现JPA规范的框架，比较出名的是hibernate。

现在，我们需要在pom文件中引入两个依赖：

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
</dependency>
```

分别为spring-data-jpq和mysql驱动。

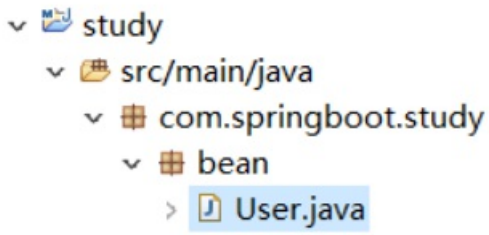
Spring Data是一个用于简化数据库访问，并支持云服务的开源框架。其主要目标是使得对数据的访问变得方便快捷，并支持map-reduce框架和云计算数据服务。 Spring Data 包含多个子项目，spring-data-jpq就是其中的一个。

修改yml文件：

```
server:
  port: 8088
  context-path: /demo
spring:
  datasource:
    driver-class-name: com.mysql.jdbc.Driver
    url: jdbc:mysql://127.0.0.1:3306/crud
    username: root
    password: 123456
  jpa:
    hibernate:
      ddl-auto: create
      show-sql: true
```

其中，设置ddl-auto: create的目的就是在项目启动的时候，就创建表。

接着，我们去新建一个JavaBean：



代码：

```
package com.springboot.study.bean;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;

@Entity
public class User {

    @Id
    @GeneratedValue
    private Integer id;

    private String username;
    private String password;
```

```
    //无参构造方法，这个必须要有，不然会报错
    public User() {

    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }



}
```

其中，主键ID设置为自增长。

然后，启动项目，发现数据库的表已经自动生成了。

名	类型	长度	小数点	不是 null	
id	int	11	0	☑	1
password	varchar	255	0	☐	
username	varchar	255	0	☐	

源码下载地址：http://java520.top/article/3489.html