转载请标明出处：   
[http://blog.csdn.net/forezp/article/details/61472783](http://blog.csdn.net/forezp/article/details/61472783" \t "_blank)   
本文出自[方志朋的博客](http://blog.csdn.net/forezp" \t "_blank)

一.什么是spring boot

Takes an opinionated view of building production-ready Spring applications. Spring Boot favors convention over configuration and is designed to get you up and running as quickly as possible.

摘自官网

翻译：采纳了建立生产就绪Spring应用程序的观点。 Spring Boot优先于配置的惯例，旨在让您尽快启动和运行。

spring boot 致力于简洁，让开发者写更少的配置，程序能够更快的运行和启动。它是下一代javaweb框架，并且它是spring cloud（微服务）的基础。

二、搭建第一个sping boot 程序

可以在start.spring.io上建项目，也可以用idea构建。本案列采用idea.

具体步骤：

new prpject -> spring initializr ->{name :firstspringboot , type: mavenproject,packaging:jar ,..} ->{spring version :1.5.2 web: web } -> ....

应用创建成功后，会生成相应的目录和文件。

其中有一个Application类,它是程序的入口:

@SpringBootApplication

public class FirstspringbootApplication {

public static void main(String[] args) {

SpringApplication.run(FirstspringbootApplication.class, args);

}

}

在resources文件下下又一个application.yml文件，它是程序的配置文件。默认为空，写点配置 ,程序的端口为8080,context-path为 /springboot：

server:

port: 8080

context-path: /springboot

写一个HelloController：

@RestController //等同于同时加上了@Controller和@ResponseBody

public class HelloController {

//访问/hello或者/hi任何一个地址，都会返回一样的结果

@RequestMapping(value = {"/hello","/hi"},method = RequestMethod.GET)

public String say(){

return "hi you!!!";

}

}

运行 Application的main(),呈现会启动，由于springboot自动内置了servlet容器，所以不需要类似传统的方式，先部署到容器再启动容器。只需要运行main()即可，这时打开浏览器输入网址：localhost:8080/springboot/hi ，就可以在浏览器上看到: hi you!!!

三.属性配置

在appliction.yml文件添加属性：

server:

port: 8080

context-path: /springboot

girl:

name: B

age: 18

content: content:${name},age:${age}

在java文件中，获取name属性，如下：

@Value("${name}")

private String name;

也可以通过ConfigurationProperties注解，将属性注入到bean中，通过Component注解将bean注解到spring容器中：

@ConfigurationProperties(prefix="girl")

@Component

public class GirlProperties {

private String name;

private int age;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

}

另外可以通过配置文件制定不同环境的配置文，具体见源码：

spring:

profiles:

active: prod

四.通过jpa方式操作数据库

导入jar ，在pom.xml中添加依赖:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

在appilication.yml中添加数据库配置：

spring:

profiles:

active: prod

datasource:

driver-class-name: com.mysql.jdbc.Driver

url: jdbc:mysql://localhost:3306/dbgirl?useUnicode=true&characterEncoding=utf8&characterSetResults=utf8

username: root

password: 123

jpa:

hibernate:

ddl-auto: create

show-sql: true

这些都是数据库常见的一些配置没什么可说的，其中ddl\_auto: create 代表在数据库创建表，update 代表更新，首次启动需要create ,如果你想通过hibernate 注解的方式创建数据库的表的话，之后需要改为 update.

创建一个实体girl，这是基于hibernate的:

@Entity

public class Girl {

@Id

@GeneratedValue

private Integer id;

private String cupSize;

private Integer age;

public Girl() {

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public Integer getAge() {

return age;

}

public void setAge(Integer age) {

this.age = age;

}

public String getCupSize() {

return cupSize;

}

public void setCupSize(String cupSize) {

this.cupSize = cupSize;

}

}

创建Dao接口, springboot 将接口类会自动注解到spring容器中，不需要我吗做任何配置，只需要继承JpaRepository 即可：

//其中第二个参数为Id的类型

public interface GirlRep extends JpaRepository<Girl,Integer>{ }

创建一个GirlController，写一个获取所有girl的api和添加girl的api ，自己跑一下就可以了:

@RestController

public class GirlController {

@Autowired

private GirlRep girlRep;

/\*\*

\* 查询所有女生列表

\* @return

\*/

@RequestMapping(value = "/girls",method = RequestMethod.GET)

public List<Girl> getGirlList(){

return girlRep.findAll();

}

/\*\*

\* 添加一个女生

\* @param cupSize

\* @param age

\* @return

\*/

@RequestMapping(value = "/girls",method = RequestMethod.POST)

public Girl addGirl(@RequestParam("cupSize") String cupSize,

@RequestParam("age") Integer age){

Girl girl = new Girl();

girl.setAge(age);

girl.setCupSize(cupSize);

return girlRep.save(girl);

}

}

如果需要事务的话，在service层加@Transaction注解即可。已经凌晨了，我要睡了.

源码；[http://download.csdn.net/detail/forezp/9778235](http://download.csdn.net/detail/forezp/9778235" \t "_blank)

关注我的专栏《史上最简单的 SpringCloud 教程 》[http://blog.csdn.net/column/details/15197.html](http://blog.csdn.net/column/details/15197.html" \t "_blank)