



链滴

# Spring Data JPA

作者: [xsx](#)

原文链接: <https://ld246.com/article/1482734996837>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## pom.xml

```
version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
□
```

```
<groupId>com.ejtgroupld</groupId>
```

```
<artifactId>JerseyDemo</artifactId>
```

```
<version>1.0-SNAPSHOT</version>
```

```
□
```

```
<packaging>jar</packaging>
```

```
□
```

```
<name>rest-versioning</name>
```

```
<description>Demo project for spring boot jersey</description>
```

```
□
```

```
<parent>
```

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

```
<artifactId>spring-boot-starter-parent</artifactId>
```

```
<version>1.4.2.RELEASE</version>
```

```
<relativePath/>
```

```
</parent>
```

```
□
```

```
<properties>
```

```
<java.version>1.8</java.version>
```

```
</properties>
```

[]

<dependencies>

[]

<dependency>

<groupId>org.springframework.bootgroupId>

<artifactId>spring-boot-starter-jerseyartifactId>

dependency>

<dependency>

<groupId>org.springframework.datagroupId>

<artifactId>spring-data-jpaartifactId>

<version>1.10.5.RELEASEversion>

dependency>

<dependency>

<groupId>org.springframework.datagroupId>

<artifactId>spring-data-rest-webmvcartifactId>

<version>2.5.5.RELEASEversion>

dependency>

<dependency>

<groupId>org.springframework.bootgroupId>

<artifactId>spring-boot-starter-data-jpaartifactId>

dependency>

<dependency>

<groupId>mysqlgroupId>

<artifactId>mysql-connector-javaartifactId>

dependency>

dependencies>

[]

```
<build>
<plugins>
<plugin>
<groupId>org.springframework.bootgroupId>
<artifactId>spring-boot-maven-pluginartifactId>
plugin>
<plugin>
<groupId>org.apache.maven.pluginsgroupId>
<artifactId>maven-compiler-pluginartifactId>
<configuration>
<source>1.7source>
<target>1.7target>
configuration>
plugin>
plugins>
build>
project>
```

## 实体类

```
package com.ejt.jersey.entity;
[]
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
[]
/**
```

- Created by xushixin on 2016/12/8.

```
*/
```

```
@Entity
```

```
public class User {
```

```
    []
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    private int id;
```

```
    private String name;
```

```
    private int age;
```

```
    []
```

```
    protected User() {
```

```
    }
```

```
    []
```

```
    public User(String name, int age) {
```

```
        this.name = name;
```

```
        this.age = 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;
}
}

```

这里有一个Customer类有三个属性，id，firstName和lastName。你也有两个构造函数。默认构造函数只存在为了JPA的目的。你不会直接使用它，所以它被指定为protected。另一个构造函数是用于建要保存到数据库的Customer实例的构造函数。

Customer类用@Entity注释，表示它是一个JPA实体。对于缺少@Table注释，假设此实体将映射到为Customer的表。

该Customer的id属性与注解@Id使JPA将其识别为对象的ID。使用注解@GeneratedValue表明，应自动生成ID的策略。

另外两个属性name和age没有留下注释。假设它们将被映射到与属性本身是相同名称的列。

## Controller类

```

package com.ejt.jersey.cotroller;
[]
/**
• Created by xushixin on 2016/12/9.
*/
[]
import java.util.List;
[]
import com.ejt.jersey.entity.User;
import org.springframework.data.repository.PagingAndSortingRepository;
import org.springframework.data.repository.query.Param;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;
[]
@RepositoryRestResource(collectionResourceRel = "people", path = "people")

```

```

public interface PersonRepository extends PagingAndSortingRepository<User, Long> {
    []
    List<User> findByName(@Param("name") String name);
    []
}

```

此存储库是一个接口，将允许您执行涉及Person对象的各种操作。它通过扩展Spring Data Commons中定义的PagingAndSortingRepository接口来获取这些操作。

在运行时，Spring Data REST将自动创建此接口的实现。然后，它将使用@RepositoryRestResource注释来指示Spring MVC在/people处创建RESTful端点。

要导出存储库，不需要@RepositoryRestResource。它仅用于更改导出详细信息，例如使用/people而不是/users的默认值。

在这里，您还定义了一个自定义查询，以基于Name检索User对象的列表。您将在本指南中了解如何一步调用。

```

package com.ejt.jersey;

[]

/**
 * Created by xushixin on 2016/12/8.
 */

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

[]

@SpringBootApplication

public class SpringBootJerseyApplication {

    []

    public static void main(String[] args) {
        SpringApplication.run(SpringBootJerseyApplication.class, args);
    }

}

```

@SpringBootApplication是一个方便的注释，添加以下所有内容：

- @Configuration将类标记为应用程序上下文的bean定义的源。

- @EnableAutoConfiguration告诉Spring Boot根据类路径设置, 其他bean和各种属性设置开始添bean。
  - 通常你会为Spring MVC应用程序添加@EnableWebMvc, 但是Spring Boot在类路径上看到pring-webmvc时会自动添加它。这将应用程序标记为Web应用程序, 并激活关键行为, 如设置DispatcherServlet。
  - @ComponentScan告诉Spring在jersey包中查找其他组件, 配置和服务, 使它能够找到控器。

main () 方法使用Spring Boot的SpringApplication.run () 方法来启动应用程序。你有没有注意没有一行XML? 也没有web.xml文件。这个Web应用程序是100%纯Java, 你不必处理配置任何管或配置文件。

Spring Boot自动启动Spring Data JPA以创建PersonRepository的具体实现, 并将其配置为使用JP与后端数据库通信。

Spring Data REST构建在Spring MVC之上。它创建了一个Spring MVC控制器, JSON转换器以及供RESTful前端所需的其他bean的集合。这些组件链接到Spring Data JPA后端。使用Spring Boot是所有自动配置; 如果你想研究它是如何工作的, 你可以从Spring Data REST中的RepositoryRestMvc Configuration开始。

## 配置文件application.yml

```
server:  
  
port: 8080  
  
[]  
  
spring:  
  
datasource:  
  
name: dataSource  
  
driver-class-name: com.mysql.jdbc.Driver  
  
url: jdbc:mysql://192.168.1.201:3306/xsxdb  
  
username: xushixin  
  
password: ejtone123  
  
initial-size: 1  
  
min-idle: 1  
  
max-wait: 6000  
  
time-between- eviction-runs-millis: 60000  
  
min- evictable-idle-time-millis: 300000
```



validation-query: select \*

test-while-idle: true

test-on-borrow: false

test-on-return: false

**Service层为调用，省略.**

**Jersey不能用，路径全部被JPA拦截，故Jersey代码不再贴来，请自行查看。**

因为路径全部被拦截，还需要某些准备工作，暂时抛开这个问题。注：JerseyConfig.java里面的@ApplicationPath注解是指定Jersey的路径，如果不设置即为默认根目录，将导致所有路径均不可用，状态码204（访问成功但是没有任何数据，不做任何操作，也不导向新的地址）。