

# homepage 第三期：服务模块的通用模块开发及数据表的相关操作

作者：[ChenforCode](#)

原文链接：<https://ld246.com/article/1573461510210>

来源网站：[链滴](#)

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



1.建立服务模块，他同样是一个父模块，在pom文件中的打包方式处仍然是pom，并且由于他是一个模块，所以不需要src目录，删除。

2.建立服务模块的自模块：通用模块，用来写course服务和user服务可能用到的代码在他的pom文件中添加相关依赖（打包方式为jar）

```
<dependencies>
  <!-- JSON处理工具 -->
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.31</version>
  </dependency>

  <!-- apache提供的工具类 -->
  <dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.9</version>
  </dependency>
```

3.在通用模块中编写相关服务

@1.完成courseInfo类

```
package cn.chenforcode.homepage;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```

/**
 * @author <a href="http://www.chenforcode.cn">PKUCoder</a>
 * @date 2019/11/11 12:57 上午
 * @description 基本课程信息
 */
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class CourseInfo {
    private Long id;
    private String courseName;
    private String courseType;
    private String counselcon;
    private String counselIntro;

    public static CourseInfo invalid() {
        return new CourseInfo(-1L, "", "", "", "");
    }
}

```

## @2.完成userInfo类

```

package cn.chenforcode.homepage;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

/**
 * @author <a href="http://www.chenforcode.cn">PKUCoder</a>
 * @date 2019/11/11 12:53 上午
 * @description 基本的用户信息
 */
@Data
@NoArgsConstructor
@AllArgsConstructor
public class UserInfo {
    private Long id;
    private String username;
    private String email;

    public static UserInfo invalid() {
        return new UserInfo(-1L, "", "");
    }
}

```

## @3.完成courseInfoRequest类

```

package cn.chenforcode.homepage;

import java.util.List;

```

```

/**
 * @author <a href="http://www.chenforcode.cn">PKUCoder</a>
 * @date 2019/11/11 1:01 上午
 * @description 课程信息请求对象
 */
public class CourseInfosRequest {
    private List<Long> ids;
}

```

4.建立课程服务模块，完成pom文件，注意这里是一个web服务，所以需要加入web相关的依赖

```

<?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">
    <parent>
        <artifactId>imooc-homepage-service</artifactId>
        <groupId>cn.chenforcode.homepage</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>homepage-course</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.18</version>
            <scope>runtime</scope>
        </dependency>

        <!-- 通用模块 -->
        <dependency>
            <groupId>cn.chenforcode.homepage</groupId>
            <artifactId>homepage-common</artifactId>

```

```

        <version>1.0-SNAPSHOT</version>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

5.编写启动类，这里的@enableJpaAuditing的作用是自动管理每一条数据的createtime和updateTime不需要自己管理

```

package cn.chenforcode.homepage;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
import org.springframework.data.jpa.repository.config.EnableJpaAuditing;

/**
 * @author <a href="http://www.chenforcode.cn">PKUCoder</a>
 * @date 2019/11/11 3:55 下午
 * @description homepageCourse服务启动类
 */
@EnableJpaAuditing
@EnableEurekaClient
@SpringBootApplication
public class HomepageCourseApplication {
    public static void main(String[] args) {
        SpringApplication.run(HomepageCourseApplication.class, args);
    }
}

```

## 6.完成配置文件

```

server:
  port: 7001
servlet:
  context-path: /homepage-course
spring:
  application:
    name: eureka-client-homepage-course
  jpa:
    hibernate:
      ddl-auto: none
    show-sql: true
  properties:
    hibernate.format_sql: true

```

```

    open-in-view: false
datasource:
  url: jdbc:mysql://127.0.0.1:3306/imooc_homepage_sc?autoReconnect=true&useUnicode=true&characterEncoding=utf8&useSSL=false
  username: root
  password: yourpassword
  driver-class-name: com.mysql.cj.jdbc.Driver
tomcat:
  max-active: 4
  min-idle: 2
  initial-size: 2
eureka:
  client:
    service-url:
      #如果是一个集群，那么把所有的eureka的地址都写到这里
      defaultZone: http://server1:8000/eureka/

```

## 7.建立homepageCourse实体表

```

package cn.chenforcode.homepage.entity;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.hibernate.annotations.UpdateTimestamp;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;

import javax.persistence.*;
import java.util.Date;

/**
 * @author <a href="http://www.chenforcode.cn">PKUCoder</a>
 * @date 2019/11/11 4:12 下午
 * @description homepage_course表映射实体
 */
@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "homepage_course")
@EntityListeners(AuditingEntityListener.class)
public class HomepageCourse {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Long id;

    //课程名称
    @Basic
    @Column(name = "course_name", nullable = false)
    private String courseName;

```

```

//课程类型 0(免费课),1(实战课)
@Basic
@Column(name = "course_type", nullable = false)
private Integer courseType;

//课程图标
@Basic
@Column(name = "course_icon", nullable = false)
private String courseIcon;

//课程介绍
@Basic
@Column(name = "course_intro", nullable = false)
private String courseIntro;

//创建时间
@Basic
@Column(name = "create_time", nullable = false)
@CreatedDate
private Date createTime;

//更新时间
@Basic
@Column(name = "update_time", nullable = false)
@LastModifiedDate
private Date updateTime;

public HomepageCourse(String courseName, Integer courseType, String courseIcon, String
ourseIntro) {
    this.courseName = courseName;
    this.courseType = courseType;
    this.courseIcon = courseIcon;
    this.courseIntro = courseIntro;
}

/**
 * @Author <a href="http://www.chenforcode.cn">PKUCoder</a>
 * @Date 2019/11/11 4:27 下午
 * @Param []
 * @Return cn.chenforcode.homepage.entity.HomepageCourse
 * @Description 返回一个无效的course
 */
public static HomepageCourse invalid() {
    HomepageCourse invalid = new HomepageCourse("", 0, "", "");
    invalid.setId(-1L);
    return invalid;
}
}

```

## 8.建立对应的仓库

```
package cn.chenforcode.homepage.repository;
```

```
import cn.chenforcode.homepage.entity.HomepageCourse;
import org.springframework.data.jpa.repository.JpaRepository;

public interface HomepageCourseRepository extends JpaRepository<HomepageCourse, Long> {
}
```