01_SpringBoot导出Excel?

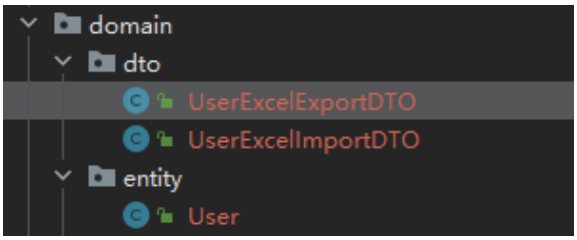
时间：2023年02月12日 22:47:57

文档：

- https://easyexcel.opensource.alibaba.com/docs/current/quickstart/write

---

一、导出DTO

- 

约定：

- 导入时，使用index将excel列映射到POJO属性上
- 导出时，使用name+index，将POJO属性映射到excel列上

二、导出接口

1. 使用BeanUtils完成DTO的生成

```
/**
 * 导出Excel
 * 1. 设置响应头
 * 2. 获取原始数据，转换为对应DTO
 * 3. 将DTO数据写入响应输出流
 */
@GetMapping("/download")
public void excelDownload(HttpServletResponse response) {
    final String filename = "default.xlsx";
    response.setHeader(HttpHeaders.ACCESS_CONTROL_EXPOSE_HEADERS, "Content-Disposition");
    response.setHeader(HttpHeaders.CONTENT_DISPOSITION, "attachment;filename=" +
            URLEncoder.encode(filename, StandardCharsets.UTF_8));

    List<User> users = new ArrayList<>();
    users.add(new User(UUID.randomUUID().toString().replaceAll("-", ""), "AAA",
            "0001", 12, LocalDateTime.now(), ""));
    users.add(new User(UUID.randomUUID().toString().replaceAll("-", ""), "BBB",
            "0002", 13, LocalDateTime.now(), ""));

    List<UserExcelExportDTO> data = users.stream().map(x -> {
        UserExcelExportDTO t = new UserExcelExportDTO();
        BeanUtils.copyProperties(x, t);
        return t;
```

```
        }).collect(Collectors.toList());

        try {
            EasyExcel.write(response.getOutputStream(), UserExcelExportDTO.class).sheet("模板").doWrite(data);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
```

2. 引入mapstruct，完成DTO的生成

```
    /**
     * 导出Excel
     * 1. 设置响应头
     * 2. 获取原始数据，转换为对应DTO
     * 3. 将DTO数据写入响应输出流
     */
    @GetMapping("/download")
    public void excelDownload(HttpServletResponse response) {
        final String filename = "default.xlsx";
        response.setHeader(HttpHeaders.ACCESS_CONTROL_EXPOSE_HEADERS, "Content-Disposition");
        response.setHeader(HttpHeaders.CONTENT_DISPOSITION, "attachment;filename=" +
                URLEncoder.encode(filename, StandardCharsets.UTF_8));

        List<User> users = new ArrayList<>();
        users.add(new User(UUID.randomUUID().toString().replaceAll("-", ""), "AAA",
                "0001", 12, LocalDateTime.now(), ""));
        users.add(new User(UUID.randomUUID().toString().replaceAll("-", ""), "BBB",
                "0002", 13, LocalDateTime.now(), ""));

        /* 使用BeanUtils完成DTO的生成 */
        // List<UserExcelExportDTO> data = users.stream().map(x -> {
        //     UserExcelExportDTO t = new UserExcelExportDTO();
        //     BeanUtils.copyProperties(x, t);
        //     return t;
        // }).collect(Collectors.toList());

        /* 引入mapstruct完成DTO的生成 */
        List<UserExcelExportDTO> data = UserExcelExportDTO.UserExcelExportDTOStruct.copy.from(users);

        try {
            EasyExcel.write(response.getOutputStream(), UserExcelExportDTO.class).sheet("模板").doWrite(data);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
```

（*依赖+新的DTO*）

```
<!-- mapstruct -->
<dependency>
    <groupId>org.mapstruct</groupId>
    <artifactId>mapstruct</artifactId>
    <version>1.5.3.Final</version>
</dependency>
<dependency>
```

```
    <groupId>org.mapstruct</groupId>
    <artifactId>mapstruct-processor</artifactId>
    <version>1.5.3.Final</version>
</dependency>
```

```
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class UserExcelExportDTO {

    @ExcelProperty(value = ”姓名”, index = 0)
    private String name;
    @ExcelProperty(value = ”编号”, index = 1)
    private String code;

    @ExcelProperty(value = ”年龄”, index = 2)
    private Integer age;

    @ExcelProperty(value = ”创建时间”, index = 3)
    @DateTimeFormat(”yyyy-MM-dd HH:mm:ss”)
    @ColumnWidth(25)
    private LocalDateTime createTime;

    @ExcelIgnore
    private String description;


    @Mapper
    public interface UserExcelExportDTOStruct {
        UserExcelExportDTOStruct copy = Mappers.getMapper(UserExcelExportDTOStruct.class);
        UserExcelExportDTO from(User user);
        List<UserExcelExportDTO> from(List<User> userList);

    }
}
```

三、时间转换与列宽度