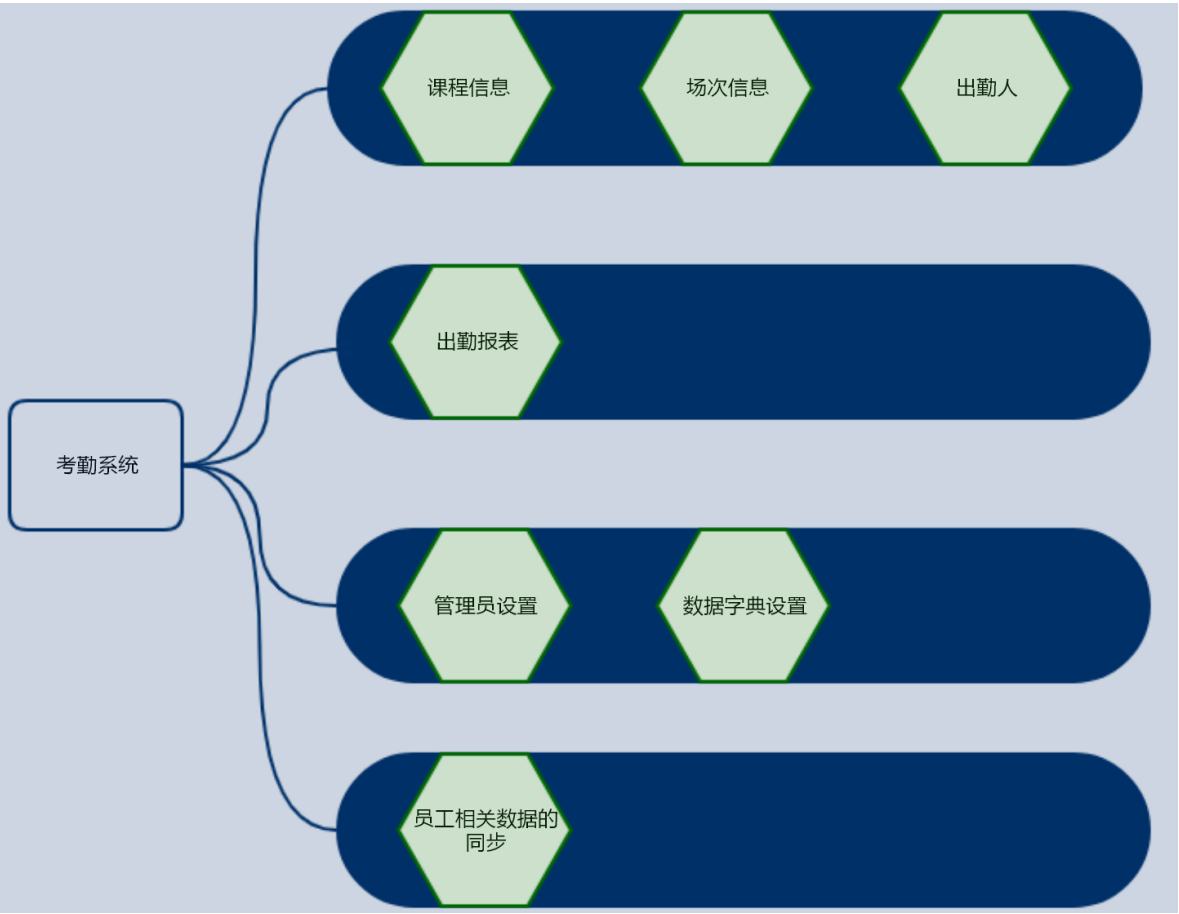


内部系统—— 培训系统后端交接文档

概述

培训系统主要负责记录Manulife员工的培训信息以及提供考勤报表，核心模块包括课程、场次、人员出勤率、管理员信息的维护以及考勤报表的展示。

一个高层次结构图如下

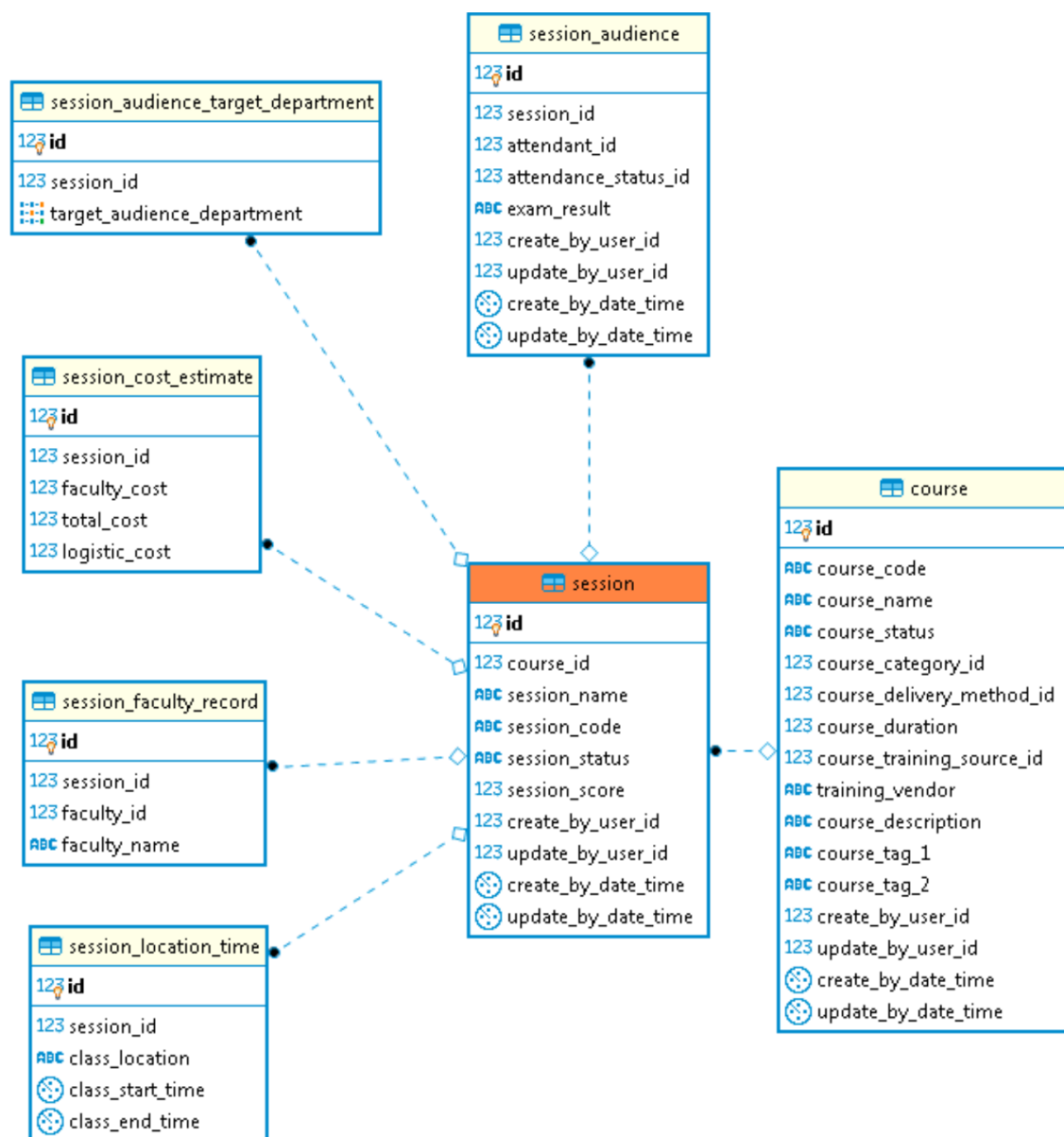


- 课程信息、场次信息、出勤人代表着完整的业务数据
- 出勤报表负责提供员工、团队、部门培训的统计信息
- 管理员设置为权限相关的设置、数据字典的设置是为了动态修改业务中需要用到的业务数据相关处理
- 培训系统中的一部分基础数据如员工，团队，部门信息是从员工系统中同步而来，这样方便了公司对员工信息的集中化管理

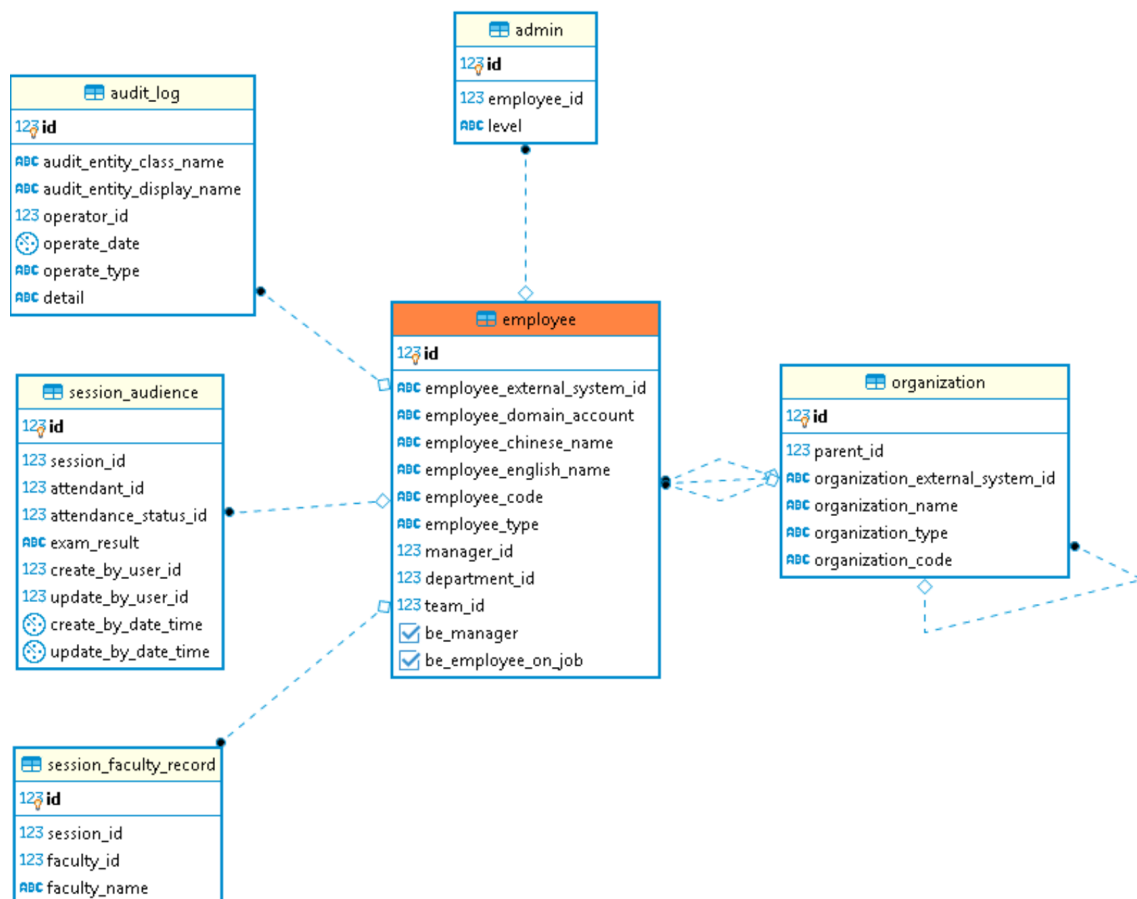
后端结构图

数据库表关系图

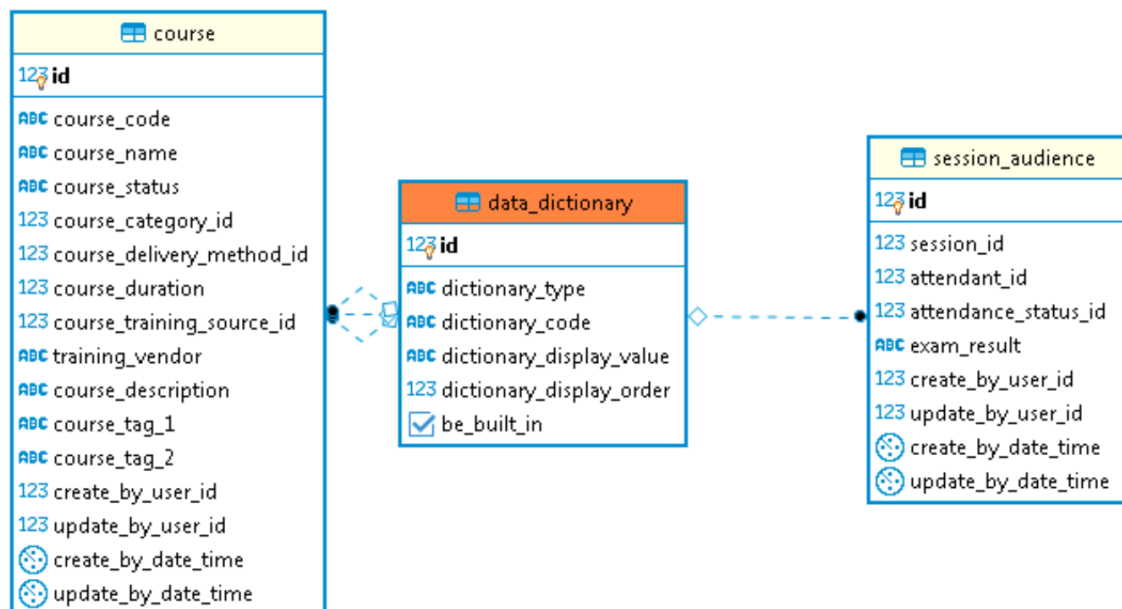
课程【course】、场次【session】以及相关依赖表的



组织关系



数据字典



Excel导入

Excel文件设计

Excel模板放置在 `resources.templates.excel` 包下，Excel的文件格式为 `{字段中文名称}\r\n({字段英文名称}:{字段类型})`，Cell的代表类 `ExcelCellMetadata` 以及接口 `ExcelTypeReader`，`ExcelRowReader` 共同完成了数据行的解析

一个例子如下

```
课程描述
(courseDescription:String)
```

支持的类型请参考 `ExcelCellDataType`

后台实现

`com.manulife.trainingrecord.application.excel` 主要用于Excel的文件下载和上传接口的定义，真正的Excel解析出现在 `com.manulife.trainingrecord.application.utils.excel`，大致处理如下，所有的Excel行数据由抽象类 `ExcelRowObject` 定义，该类提供行数、数据类型、数据类型名称等元数据从而便于异常消息的提示

一个调用示例

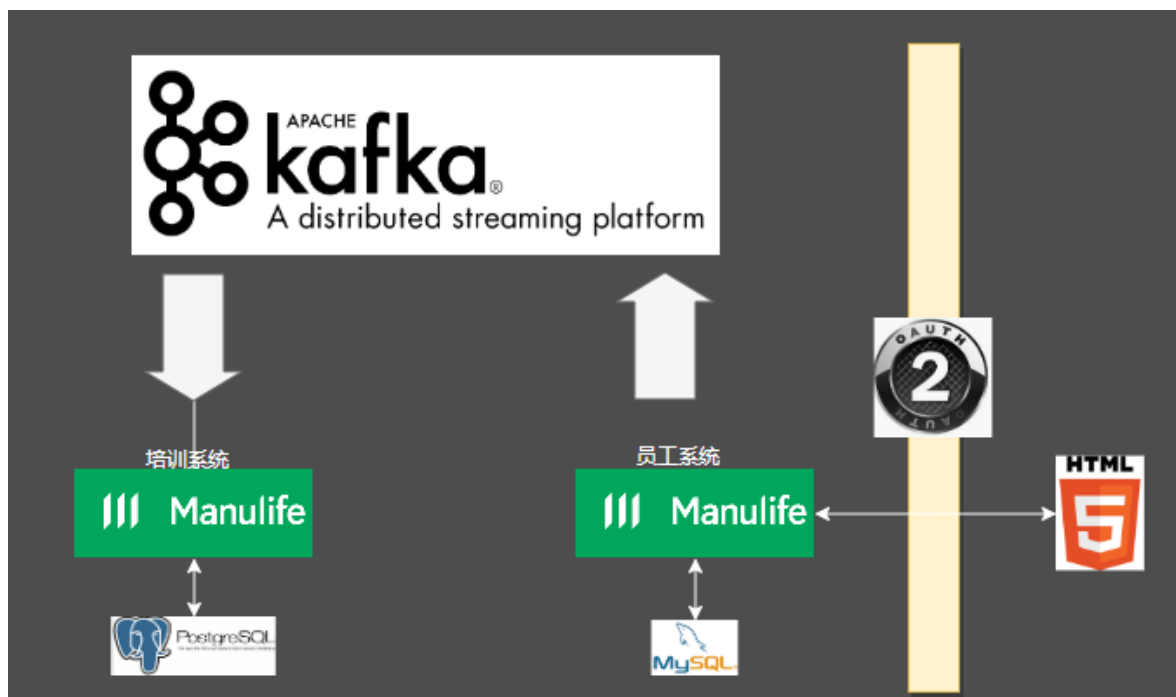
```
List<CourseExcelDTO> courseDtos = ExcelFileUtils.readFirstSheetFrom(
    excelFileInputStream,
    sheetName,
    fieldDefineRow,
    excelRowReader
);
```

- `sheetName` 用来验证Excel是否最新
- `fieldDefineRow` 表示字段行的位置
- `excelRowReader` 实例化一个`ExcelRowObject`对象

`ExcelRowReader` 默认实现是 `ExcelRowReaderBySetter` 通过提供赋值函数为对象字段赋值，在赋值函数中也可以抛出异常，解析类会自动捕获并转换成 `BusinessException`。

基于Kafka信息同步

概略图



实现细节

组织架构

manulife: 虚拟根节点

- |— deartment: 部门
- |— manager: 经理
- |— team: 团队

1、配置文件 `application.properties` 增加

```
spring.kafka.bootstrap-servers={xxx}
spring.kafka.consumer.group-id={xxx}
```

2、消费端监听

```
@KafkaListener(topics="{xxx}")
public void precessMessage(String kafkaMessage) {
    xxx
}
```

3、整体设计介绍

employeesystem : 员工系统集成

- |— client : 客户端域, 通过feign从员工系统中抓取员工, 部门, 团队信息
- |— consume: 负责消费kafka的消息, 这也是同步的核心处理
- |— log: 记录消息消费过程中出现的问题, 方便排查

细节介绍

- 消费入口 `MessageConsumeServiceDispatch.processMessage`

- **InstanceOrganization** 一个对实例化组织的高层次抽象，在培训系统中，组织包括部门，管理者，团队，**SHART_MGT**。
- 消息消费使用工厂服务定位模式，所有的消息处理者实现 **EmployeeSystemMessageConsumeService** 接口，需要注意的是，组织含有父级节点在实现类中需要特殊处理