

## 设计思路

- 使用mybatis连接mysql数据库，model层将表mn\_buy\_ticket的数据结构化，mapper层配置数据库查询操作
- kafka生产者通过service层操作mapper层获取mysql中的数据，对于获取的每一条数据转化成json字符串放入KafkaProducer[String, String]中

## 测试入口

- job3\src\test\java\com\bigdata\week3\job3路径下Job3ApplicationTests.java执行测试

## 核心代码

### model

```
public class BuyInfo {
    private String username;
    private Date buy_time;
    private String buy_address;
    private String origin;
    private String destination;
    //getter & setter
    //constructor
    //toString
}
```

### mapper

```
@Mapper
@Repository
public interface BuyInfoMapper {
    public List<BuyInfo> listBuyInfos();
}

<mapper namespace="com.bigdata.week3.job3.Mapper.BuyInfoMapper">
    <select id="listBuyInfos" resultType="com.bigdata.week3.job3.Model.BuyInfo">
        select * from mn_buy_ticket
    </select>
</mapper>
```

### service

```
@Service
public class BuyInfoService {

    @Autowired
    private BuyInfoMapper buyInfoMapper;

    public List<BuyInfo> getBuyInfoList(){
        return buyInfoMapper.listBuyInfos();
    }
}
```

## kafkaProducer

```
def produceToKafka(s3Content: String): Unit = {
    val props = new Properties
    props.put("bootstrap.servers", bootstrapServers)
    props.put("acks", "all")
    props.put("key.serializer",
"org.apache.kafka.common.serialization.StringSerializer")
    props.put("value.serializer",
"org.apache.kafka.common.serialization.StringSerializer")
    val producer = new KafkaProducer[String, String](props)
    val dataArr = s3Content.split("\n")
    println("开始传输")
    for (s <- dataArr) {
        if (!s.trim.isEmpty) {
            val record = new ProducerRecord[String, String](topic, null, s)
            println("开始生产数据: " + s)
            producer.send(record)
        }
    }
    producer.flush()
    producer.close()
}
```

## @test

```
@SpringBootTest
class Job3ApplicationTests {
    @Autowired
    BuyInfoService buyInfoService;
    @Test
    void contextLoads() {
        List<BuyInfo> buyInfoList=buyInfoService.getBuyInfoList();
        for(BuyInfo buyInfo:buyInfoList){
            Producer.produceToKafka(JSON.toJSONString(buyInfo));
        }
    }
}
```

## 最终效果

开始传输

开始生产数据: {"buy\_address": "阳江市东华北路168号", "buy\_time": 1567507737000, "desti

开始传输

开始生产数据: {"buy\_address": "济南市沿江东路451号", "buy\_time": 1553254516000, "desti

开始传输

开始生产数据: {"buy\_address": "雅安市天河路188号", "buy\_time": 1555383954000, "destina

开始传输

开始生产数据: {"buy\_address": "珠海市东华北路168号", "buy\_time": 1572753759000, "desti

开始传输

开始生产数据: {"buy\_address": "兰州市环市东路412号", "buy\_time": 1561456157000, "desti

开始传输

开始生产数据: {"buy\_address": "常德市大道中明月一路9号", "buy\_time": 1555311960000, "de

开始传输

开始生产数据: {"buy\_address": "邵阳市三育路44号", "buy\_time": 1551855615000, "destinat

开始传输

开始生产数据: {"buy\_address": "揭阳市大道中明月一路9号", "buy\_time": 1547519237000, "de

开始传输

开始生产数据: {"buy\_address": "达州市东风东路552号", "buy\_time": 1551067888000, "desti

开始传输

开始生产数据: {"buy\_address": "茂名市流花路122号", "buy\_time": 1575684484000, "destina

开始传输

开始生产数据: {"buy\_address": "乌鲁木齐市天河路188号", "buy\_time": 1572590926000, "des

开始传输

开始生产数据: {"buy\_address": "福州市东湖珠江宾馆(寺右一马路店)", "buy\_time": 1577706912

开始传输

开始生产数据: {"buy\_address": "怀化市东湖珠江宾馆(寺右一马路店)", "buy\_time": 1565355418

开始传输

开始生产数据: {"buy\_address": "自贡市小北路198号", "buy\_time": 1547029685000, "destina