

## 1. API 생성

### 1. 월별 접속자 수

```
<select id="selectMonthLogin" parameterType="string" resultType="hashMap">
    select count(*) as totCnt
    from statistic.requestinfo ri
    where mid(ri.createDate, 3, 2) = #{month} and ri.requestCode = 'L';
</select>
```

### 2. 일별 접속자 수

```
<select id="selectDayLogin" parameterType="string" resultType="hashMap">
    select count(*) as totCnt
    from statistic.requestinfo ri
    where mid(ri.createDate, 5, 2) = #{day} and ri.requestCode = 'L';
</select>
```

### 3. 평균 하루 로그인 수

```
<select id="selectAverageLogin" resultType="hashMap">
    select CONCAT('20', mid(ri.createDate, 1, 2), '-', mid(ri.createDate, 3, 2)) yd,
    count(*) / (DATEDIFF(LAST_DAY(CONCAT('20',
                                                mid(ri.createDate, 1, 2),
                                                '-',
                                                mid(ri.createDate, 3, 2),
                                                '-01')),
                        CONCAT('20',
                                mid(ri.createDate, 1, 2),
                                '-',
                                mid(ri.createDate, 3, 2), '-01')) + 1)
    as totAverage
    from statistic.requestinfo ri
    where ri.requestCode = 'L'
    group by yd;
</select>
```

Group by 절을 이용해 년, 월별로 묶어 로그인 횟수를 구하고, MySQL의 DATEDIFF, LAST\_DAY 함수를 이용해 해당 월의 전체 일 수를 구해 이 값으로 로그인 횟수를 나눠줬습니다.

#### 4. 공휴일을 제외한 년, 월별 로그인 수

```
<select id= "selectExceptHolidayLogin" resultType= "hashMap">
  select count(*) as totCnt
  from statistic.requestinfo ri
<choose>
  <when test= 'holidayList.size() == 0'> <!-- 해당 년, 월에 공휴일이 없는 경우 -->
    where left(ri.createDate, 2) = #{year} and mid(ri.createDate, 3, 2) = #{month}
    and WEEKDAY(CONCAT('20',
                        mid(ri.createDate, 1, 2),
                        '-',
                        mid(ri.createDate, 3, 2),
                        '-',
                        mid(ri.createDate, 5, 2)))!=5
    and WEEKDAY(CONCAT('20',
                        mid(ri.createDate, 1, 2),
                        '-',
                        mid(ri.createDate, 3, 2),
                        '-',
                        mid(ri.createDate, 5, 2)))!=6
    and ri.requestCode = 'L';
  </when>
  <otherwise> <!-- 해당 년, 월에 공휴일이 있는 경우 -->
    where left(ri.createDate, 2) = #{year} and mid(ri.createDate, 3, 2) = #{month}
    and WEEKDAY(CONCAT('20',
                        mid(ri.createDate, 1, 2),
                        '-',
                        mid(ri.createDate, 3, 2),
                        '-',
                        mid(ri.createDate, 5, 2)))!=5
    and WEEKDAY(CONCAT('20',
                        mid(ri.createDate, 1, 2),
                        '-',
                        mid(ri.createDate, 3, 2),
                        '-',
                        mid(ri.createDate, 5, 2)))!=6
```

and left(ri.createDate, 6) NOT IN

```
<foreach collection="holidayList" item="item" open="(" separator="," close=")">
```

    <!-- collection : 리스트 파라미터의 이름을 넣는다.

        item : 리스트 내의 원소를 참조할 이름

        separator : 원소들을 구분하기 위해 출력할 문자열 -->

        #{item}

```
</foreach>
```

and ri.requestCode = 'L';

```
</otherwise>
```

```
</choose>
```

```
</select>
```

WEEKDAY 함수를 이용해 토요일, 일요일인 날짜는 제외하고, 공공 데이터 포털의 특일 정보 API를 이용해 공휴일 날짜를 불러와 List로 저장했습니다. 이 리스트를 statisticMapper.xml로 넘겨 foreach 문을 돌려서 공휴일에 속하는 날짜는 count에서 제외했습니다. 공휴일이 없는 달의 경우 List가 빈 상태여서 오류가 발생했으므로, mybatis의 choose 절을 이용해 list의 size가 0 일 때는 공휴일 처리를 하지 않도록 했습니다.

#### 5. 부서별, 월별 로그인 수

```
<select id="selectDepartmentMonthLogin" parameterType="string" resultType="hashMap">
```

    select count(\*) as totCnt

    from statistic.requestinfo ri INNER JOIN statistic.user usr ON (ri.userID = usr.userID)

    where usr.HR\_ORGAN = #{dep} and mid(ri.createDate, 3, 2) = #{month}

        and ri.requestCode = 'L';

```
</select>
```