# 2023.8.1

## 1、 请求参数x-www-form-urlencoded 、form-data和raw

这些是常见的 HTTP 请求参数传递方式，每种方式都有自己的特点和用途。

1. `x-www-form-urlencoded`：
   - 这是一种常见的参数传递方式，适用于在 POST 请求中使用表单数据。
   - 参数以键值对的形式出现，通过使用 URL 编码将键和值进行编码，然后将它们连接成字符串。
   - 这种方式适用于简单的键值对参数，并且可以直接在请求体中发送。
2. `form-data`：
   - 这是一种常用的参数传递方式，适用于上传文件或者包含二进制数据的场景。
   - 参数被视为一个单独的部分，每个参数都有一个唯一的标识符（boundary）。
   - 每个参数都会被封装为一个独立的部分，可以包括文件、文本等不同类型的数据。
   - 在发送请求时，参数会被封装为多个不同的部分，并按照一定的格式进行传输。
3. `raw`：
   - 这是一种直接在请求体中发送原始数据的方式，适用于发送任意类型的数据，如 JSON、XML 或其他自定义格式的数据。
   - 数据以原始的形式出现，没有经过编码或封装的处理。
   - 通常需要手动设置请求头（Content-Type）来指定数据的类型和编码。

根据具体的场景和需求，选择适当的参数传递方式。如果需要发送表单数据或者键值对参数，可以使用 `x-www-form-urlencoded` 或者 `form-data`。如果需要发送原始的数据格式，可以使用 `raw`。

希望这个解答对你有所帮助！如果还有其他问题，请随时提问。

## 2、理解

```
<select id="getContSignApproving"
resultType="com.nbnfsoft.contract.domain.output.ContractListQueryOutput">
        SELECT * FROM(
        SELECT
        ID,
        STATUS,
        CODE,
        NAME,
        TYPE,
        TAG,
        BUSINESS_ID,
        <if test="flag == 1 or flag == 3">
            SUP_B AS SUPPLIER,
        </if>
        <if test="flag == 2">
            (select a.name from nfcont_supplier a where a.id = ci.sup_b) AS
SUPPLIER,
            case status when '1' then '未提交' when '2' then
            (
            SELECT listagg ( NE.EMP_NAME, ',' ) within GROUP ( ORDER BY
NE.EMP_NAME )
            FROM NFDIC_EMPLOYEE NE
            WHERE NE.ID IN(
```

```
            select regexp_substr(CI.current_apr_emp, '[^,]+', 1, level ) as ids
            from dual connect by regexp_substr(CI.current_apr_emp, '[^,]+', 1,
level ) is not null
            )) else '审批结束' end current_emp_name,
        </if>
        PRICE,
        (
        SELECT
        listagg ( ND.DEPT_NAME, ',' ) within GROUP ( ORDER BY ND.DEPT_NAME )
        FROM
        NFDIC_DEPT ND
        WHERE
        ND.ID IN

        (
        SELECT
        REGEXP_SUBSTR( CI.CHARGE_DEPT, '[^,]+', 1, LEVEL ) AS deptIds
        FROM
        DUAL CONNECT BY REGEXP_SUBSTR( CI.CHARGE_DEPT, '[^,]+', 1, LEVEL ) IS NOT
NULL
        )
        )AS DEPT_NAME,
        (SELECT EMP_NAME FROM NFDIC_EMPLOYEE WHERE ID = CI.CREATE_ID) AS
EMP_NAME,
        COMMIT_DATE,
        SIGN_DATE,
        CONT_STYLE
        FROM NFCONT_INFO CI
        ORDER BY CREATE_DATE DESC
        )
        WHERE STATUS=2
        <if test="input.code != null and input.code != ''">
            AND CODE LIKE CONCAT('%',CONCAT(#{input.code},'%'))
        </if>

        <if test="input.business_id != null and input.business_id != ''">
            AND BUSINESS_ID LIKE CONCAT('%',CONCAT(#{input.business_id},'%'))
        </if>

        <if test="input.name != null and input.name != ''">
            AND NAME LIKE CONCAT('%',CONCAT(#{input.name},'%'))
        </if>

        <if test="input.type != null and input.type != ''">
            AND TYPE LIKE CONCAT('%',CONCAT(#{input.type},'%'))
        </if>

        <if test="input.supplier != null and input.supplier != ''">
            AND SUPPLIER LIKE CONCAT('%',CONCAT(#{input.supplier},'%'))
        </if>

        <if test="input.price != null">
            AND PRICE = #{input.price}
        </if>

        <if test="input.commit_start_date != null">
            AND TO_CHAR(COMMIT_DATE,'yyyy-MM-dd') >= TO_CHAR(#
{input.commit_start_date},'yyyy-MM-dd')
```

```
        </if>

        <if test="input.commit_end_date != null">
            AND TO_CHAR(COMMIT_DATE,'yyyy-MM-dd') &lt;= TO_CHAR(#
{input.commit_end_date},'yyyy-MM-dd')
        </if>
    </select>
```

## 3、分析

```
<if test="flag == 2">
    (select a.name from nfcont_supplier a where a.id = ci.sup_b) AS SUPPLIER,
        case status when '1' then '未提交' when '2' then
        (
        SELECT listagg ( NE.EMP_NAME, ',' ) within GROUP ( ORDER BY NE.EMP_NAME
)
        FROM NFDIC_EMPLOYEE NE
        WHERE NE.ID IN(
        select regexp_substr(CI.current_apr_emp, '[^,]+', 1, level ) as ids
        from dual connect by regexp_substr(CI.current_apr_emp, '[^,]+', 1, level
) is not null
        )) else '审批结束' end current_emp_name,
</if>
```

### 1.case when

在 Oracle 数据库中，`CASE WHEN` 是一种用于条件判断的表达式。它可以根据给定的条件返回不同的值。

以下是 `CASE WHEN` 表达式的基本语法：

```
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    ...
    ELSE default_result
END
```

在这个语法结构中，每个 `WHEN` 子句都包含一个条件和与之关联的结果。当满足某个条件时，相应的结果将被返回。如果没有任何条件满足，那么会返回 `ELSE` 后面指定的默认结果。

下面是一个示例，演示如何使用 `CASE WHEN` 表达式：

```
SELECT employee_id, first_name, last_name,
    CASE
        WHEN salary >= 5000 THEN 'High'
        WHEN salary >= 3000 THEN 'Medium'
        ELSE 'Low'
    END AS salary_category
FROM employees;
```

在这个示例中，根据员工的薪水（salary）划分了不同的薪资类别。如果薪水大于等于 5000，则返回 "High"；如果薪水大于等于 3000，则返回 "Medium"；否则返回 "Low"。使用 `AS` 关键字可以为 `CASE WHEN` 表达式指定一个别名作为查询结果的列名。

需要注意的是，`CASE WHEN` 表达式在 SELECT 查询、UPDATE 语句或其他支持表达式的地方均可使用，以便在不同的条件下返回不同的结果。

希望这个解答对你有所帮助！如果还有其他问题，请随时提问。

## 4、分析2

```
SELECT
    REGEXP_SUBSTR( #{input.status}, '[^,]+', 1, LEVEL ) AS LIST
    FROM
    DUAL CONNECT BY REGEXP_SUBSTR( #{input.status}, '[^,]+', 1, LEVEL ) IS NOT
NULL
```

这是一个使用 Oracle 数据库的 SQL 查询语句，其中使用了正则表达式函数 `REGEXP_SUBSTR` 和连接查询（`CONNECT BY`）来将字符串中的逗号分隔的值拆分为多行。

以下是对这个 SQL 查询语句的解析和说明：

```
sql复制代码SELECT
    REGEXP_SUBSTR( #{input.status}, '[^,]+', 1, LEVEL ) AS LIST
FROM
    DUAL
CONNECT BY
    REGEXP_SUBSTR( #{input.status}, '[^,]+', 1, LEVEL ) IS NOT NULL;
```

- `REGEXP_SUBSTR( #{input.status}, '[^,]+', 1, LEVEL )`：这是一个正则表达式函数，用于从输入的 `#{input.status}` 字符串中提取逗号分隔的值。`'[^,]+'` 表示匹配除逗号外的任意字符，`1` 表示起始位置，`LEVEL` 表示循环的深度。
- `AS LIST`：使用 `AS` 关键字给提取的值指定别名为 `LIST`，作为查询结果的列名。
- `FROM DUAL`：`DUAL` 是 Oracle 的一个虚拟表，用于在没有实际表的情况下执行一些简单的查询。在这里，我们使用 `DUAL` 作为查询的源。
- `CONNECT BY REGEXP_SUBSTR( #{input.status}, '[^,]+', 1, LEVEL ) IS NOT NULL`：`CONNECT BY` 是 Oracle 的连接查询语法，在这里被用来实现循环，并在每个循环中提取字符串中的逗号分隔值。`IS NOT NULL` 部分用于终止连接查询，以确保所有的拆分值都被提取。

请注意，`#{input.status}` 是 MyBatis 中的占位符语法，表示这个查询是一个预编译的参数化查询，具体的值将在执行时动态替换。

希望对你有所帮助！如果还有其他问题，请随时提问。