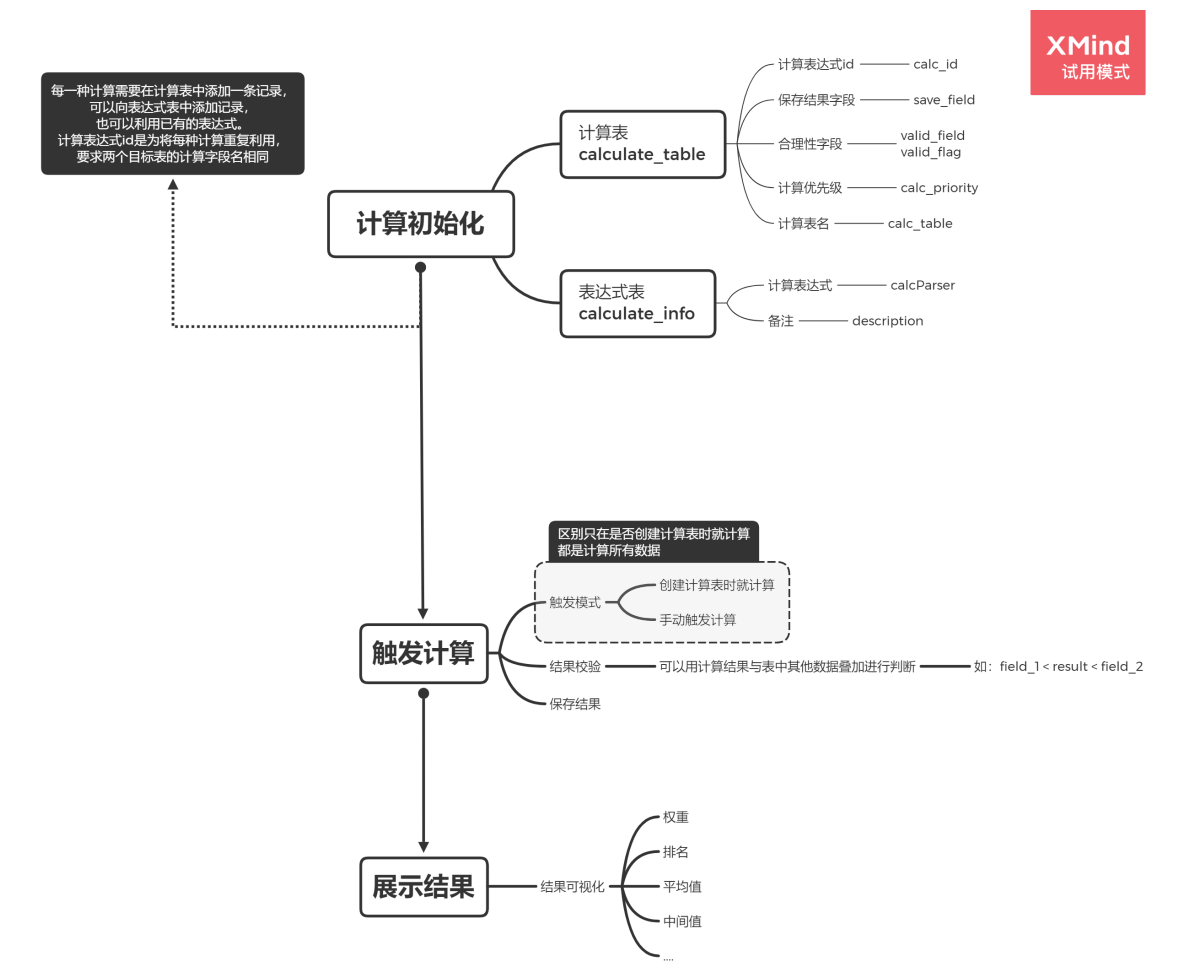


计算模块 设计概览



暂时只能做到行内数据计算，数据保存在行内。。

最多可以做到在同表中列的运算(求和、平均等等)，但运算结果也只能保存在行内。

现有 API


基于如下数据进行API测试：

测试数据

表 calc_test

	id	test_field_1	test_field_2	test_field_3	result	result_2	result_valid	result_2_valid
▶	1	100	9	8	117	908	1	0

表 calculate_info

名	类型	长度	小数点	不是 null	虚拟	键	注释
▶ id	int	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	
calcParser	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>		
description	varchar	999		<input type="checkbox"/>	<input type="checkbox"/>		
calcType	tinyint	1		<input type="checkbox"/>	<input type="checkbox"/>		

id	calcParser	description	calcType
▶ 3	test_field_1 + test_field_2 + test_field_3	测试用	(Null)
4	1+1	(Null)	(Null)
5	test_field_1 * test_field_2 + test_field_3	测试2	(Null)

表 calculate_table

名	类型	长度	小数点	不是 null	虚拟	键	注释
▶ id	int	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1	
calc_table	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
calc_id	int	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
calc_priority	int	10		<input type="checkbox"/>	<input type="checkbox"/>		
save_field	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
valid_flag	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>		
valid_field	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>		

id	calc_table	calc_id	calc_priority	save_field	valid_flag	valid_field
▶ 1	calc_test	3	99	result	100 < result	result_valid
2	calc_test	5	98	result_2	result_2 < 0	result_2_valic

/create

/calculate

描述

新建计算表达式。

```
METHOD: POST
REQUEST:
{
  "calcParser": "test_field_1 * test_field_2 + test_field_3",
  "description": "测试2"
}
RESPOND:
{
  "message": "创建计算信息成功",
  "statusCode": 200,
  "success": true,
  "data": {
    "id": 5,
    "calcParser": "test_field_1 * test_field_2 + test_field_3",
    "description": "测试2",
    "calcType": null
  }
}
```

/calctable

描述

新建计算表。

```
METHOD: POST
REQUEST:
{
  "calc_table": "calc_test",
  "calc_id": 5,
  "calc_priority": 98,
  "save_field": "result_2"
```

```

}
RESPOND:
{
  "message": "创建计算表成功",
  "statusCode": 200,
  "calc_stack": {
    "calc_msg": "不需要执行计算",
    "calc_info": {
      "id": 5,
      "calcParser": "test_field_1 * test_field_2 + test_field_3",
      "description": "测试2",
      "calcType": null
    }
  }
}
}

```

/calculate

/table

描述

根据表名触发计算表的计算。(根据在calculate_table里定义的优先级先后进行)

```

METHOD: GET
REQUEST:
{
  "calc_table": "calc_test"
}
RESPOND:
{
  "statusCode": 200,
  "message": "计算完成",
  "calc_stack": [
    {
      "calc_msg": "计算完成 共 0 个错误",
      "calc_info": {
        "id": 3,
        "calcParser": "test_field_1 + test_field_2 + test_field_3",
        "description": "测试用",
        "calcType": null
      }
    },
    {
      "calc_msg": "计算完成 共 0 个错误",
      "calc_info": {
        "id": 5,
        "calcParser": "test_field_1 * test_field_2 + test_field_3",
        "description": "测试2",
        "calcType": null
      }
    }
  ]
}

```

/test

测试表达式、返回结果的api

/data

返回目标表中的所有数据

```
METHOD: GET
REQUEST:
{
  "tablename": "calc_test"
}
RESPOND:
{
  "statusCode": 200,
  "success": true,
  "result": [
    {
      "id": 1,
      "test_field_1": 100,
      "test_field_2": 9,
      "test_field_3": 8,
      "result": "117",
      "result_2": "908",
      "result_valid": 1,
      "result_2_valid": 0,
      "result_3": "9"
    },
    {
      "id": 2,
      "test_field_1": 200,
      "test_field_2": 11,
      "test_field_3": 18,
      "result": "229",
      "result_2": "2218",
      "result_valid": 1,
      "result_2_valid": 0,
      "result_3": "161"
    }
  ]
}
```

/calctable

返回符合条件的所有计算信息 (calculate_table 表)

```
METHOD: GET
REQUEST:
{
  "calc_table": "calc_test"
}
RESPOND:
{
  "statusCode": 200,
  "success": true,
  "result": [
```

```

    {
      "id": 3,
      "calc_table": "calc_test",
      "calc_id": 6,
      "calc_priority": 100,
      "save_field": "result_3",
      "valid_flag": null,
      "valid_field": null,
      "calc_flag": "test_field_3 < 20"
    },
    {
      "id": 1,
      "calc_table": "calc_test",
      "calc_id": 3,
      "calc_priority": 99,
      "save_field": "result",
      "valid_flag": "100 < result",
      "valid_field": "result_valid",
      "calc_flag": null
    },
    {
      "id": 2,
      "calc_table": "calc_test",
      "calc_id": 5,
      "calc_priority": 98,
      "save_field": "result_2",
      "valid_flag": "result_2 < 0",
      "valid_field": "result_2_valid",
      "calc_flag": null
    }
  ]
}

```

/calc_info

返回符合条件的计算表达式(calculate_info)

```

METHOD: GET
REQUEST:
{}
RESPOND:
{
  "statusCode": 200,
  "success": true,
  "result": [
    {
      "id": 3,
      "calcParser": "test_field_1 + test_field_2 + test_field_3",
      "description": "测试用",
      "calcType": null
    },
    {
      "id": 4,
      "calcParser": "1+1",
      "description": null,
      "calcType": null
    }
  ],
}

```

```

    {
      "id": 5,
      "calcParser": "test_field_1 * test_field_2 + test_field_3",
      "description": "测试2",
      "calcType": null
    },
    {
      "id": 6,
      "calcParser": "average(column('test_field_1','test_field_1 > 100'))
+ test_field_2 + sum(column('test_field_3', 'test_field_3 > 20'))",
      "description": null,
      "calcType": 1
    }
  ]
}

```

/test_parse

测试列计算（只支持列计算）,不会保存结果

```

METHOD: POST
REQUEST:
{
  "calc_table": "calc_test",
  "parse": "average(column('test_field_1'))"
}
RESPOND:
{
  "statusCode": 200,
  "success": true,
  "result": {
    "error": null,
    "result": 150
  }
}

```

可能需要的 API

/select/calculate

筛选某个表已有的所有计算表达式

或者是筛选所有计算表达式？

是否需要对外表达式进行分门别类，供用户筛选？

/calculate/table&calc_id

可能需要跳过优先级直接触发特定计算？

...

