

# KYLIN-5529 Support adding base indexes by selecting "Base Aggregate Index" or "Base Detail Index".

## 1. 背景

希望在添加基础索引时可单选“基础聚合索引”，且同时支持页面和API的添加方式。

## 2. 验收标准

- 1. 页面添加基础索引时支持单独添加基础聚合索引或基础明细索引
- 2. 创建模型API支持单独添加基础聚合索引或基础明细索引

## 3. 详细设计

### 3.1 页面添加基础索引时支持单独添加基础聚合索引或基础明细索引



## 1. 接口

POST kylin/api/index\_plans/base\_index

2. 现状

接口可以满足要求: 页面添加基础索引时支持单独添加基础聚合索引或基础明细索引

3. 参数使用示例:

```
1 {"model_id":"1cd15c0a-b857-f6af-615e-8f3c33e47018","project":"test","load_data":
2 "source_types":["BASE_TABLE_INDEX"]}
3 }
```

4. 测试

source_types参数	结果	接口截图
source_types 只传 BASE_AGG_INDEX	调用接口成功只添加基础聚合索引	
source_types 只传 BASE_TABLE_INDEX	调用接口成功只添加基础明细索引	
source_types 传 BASE_AGG_INDEX 和 BASE_TABLE_INDEX	调用接口成功添加基础聚合索引和基础明细索引	
source_types 不传	调用接口成功添加基础聚合索引和基础明细索引	

5. 修改

暂无

5.2 创建模型API支持单独添加基础聚合索引或基础明细索引

1. 接口

- POST http://host:port/kylin/api/models  
[https://docs.kyligence.io/books/v4.6/zh-cn/developer/v4/model\\_api/model\\_management\\_api.cn.html#%E5%88%9B%E5%BB%BA%E6%A8%A1%E5%9E%8B](https://docs.kyligence.io/books/v4.6/zh-cn/developer/v4/model_api/model_management_api.cn.html#%E5%88%9B%E5%BB%BA%E6%A8%A1%E5%9E%8B)

2. 现状

- with\_base\_index: true : 基础聚合索引和基础明细索引都创建
- with\_base\_index: false : 基础聚合索引和基础明细索引都不创建

```

1 //调用创建基础索引会同时创建
2 public void createAndAddBaseIndex(NDataModel model) {
3     checkIsNotCachedAndShared();
4     LayoutEntity agg = createBaseAggIndex(model);
5     LayoutEntity table = createBaseTableIndex(model);
6     List<LayoutEntity> baseLayouts = Lists.newArrayList();
7     baseLayouts.add(agg);
8     if (table != null) {
9         baseLayouts.add(table);
10    }
11    createAndAddBaseIndex(baseLayouts);
12 }

```

### 3. 修改

1. 为了支持单独控制基础聚合索引和基础明细索引是否创建

`IndexPlan` 中新建方法 `createAndAddBaseIndex(NDataModel model, List<IndexEntity.Source> sources)`, 添加参数用来单独控制创建基础索引

- `IndexEntity.Source.BASE_AGG_INDEX` : 单独控制创建基础聚合索引
- `IndexEntity.Source.BASE_TABLE_INDEX` : 单独控制创建基础明细索引

2. 为了保持与修改前一致, `createAndAddBaseIndex(NDataModel model)` 调用 `createAndAddBaseIndex(model, Lists.newArrayList(IndexEntity.Source.BASE_AGG_INDEX, IndexEntity.Source.BASE_TABLE_INDEX))`; 默认都创建

```

1 public void createAndAddBaseIndex(NDataModel model, List<IndexEntity.Source> sources) {
2     checkIsNotCachedAndShared();
3     List<LayoutEntity> baseLayouts = Lists.newArrayList();
4     if (sources.contains(IndexEntity.Source.BASE_AGG_INDEX)) {
5         LayoutEntity agg = createBaseAggIndex(model);
6         baseLayouts.add(agg);
7     }
8
9     if (sources.contains(IndexEntity.Source.BASE_TABLE_INDEX)) {
10        LayoutEntity table = createBaseTableIndex(model);
11        if (table != null) {
12            baseLayouts.add(table);
13        }
14    }
15    createAndAddBaseIndex(baseLayouts);
16 }

```

```

14     }
15     if (!baseLayouts.isEmpty()) {
16         createAndAddBaseIndex(baseLayouts);
17     }
18 }
19
20 public void createAndAddBaseIndex(NDataModel model) {
21     ArrayList<IndexEntity.Source> sources = Lists.newArrayList(IndexEntity.Source
22         IndexEntity.Source.BASE_TABLE_INDEX);
23     createAndAddBaseIndex(model, sources);
24 }

```

### 3. 为了满足PRD约定的规则:

现有参数保留:

- `with_base_index` - 可选 `boolean`，是否添加基础索引，基础索引包含模型全部维度和度量，随着模型变化自动更新，默认值: `false`

新增参数用于选择要添加的索引类型:

- `base_index_type` - 可选 `array[string]`，选择要添加的基础索引类型，可选值 `base_agg_index`，`base_table_index`
  - `[base_agg_index, base_table_index]` 两个都添加
  - `[base_agg_index]` 只添加基础聚合索引
  - `[base_table_index]` 只添加基础明细索引
  - `[]` 不添加基础索引

新参数 `base_index_type` 不与旧参数 `with_base_index` 强制绑定，可单独使用，当 `base_index_type` 存在时，以新参数为准，如下表:

	没有配置 <code>base_index_type</code>	配置 <code>base_index_type</code>
<code>with_base_index = true</code>	两个都添加	按 <code>base_index_type</code> 添加
<code>with_base_index = false</code>	两个都不添加	按 <code>base_index_type</code> 添加
没有配置 <code>with_base_index</code>	两个都不添加	按 <code>base_index_type</code> 添加







手册不需要去掉旧参数 `with_base_index`，只需要解释说明新参数。

在 `ModelService` 中, `modelRequest.isWithBaseIndex()` 替换为分别判断基础聚合索引和基础明细索引



### 4. 测试

参数	结果	接口截图
----	----	------

### 单参数

"with_base_index":true	基础聚合索引和基础明细索引都创建,与之前保持一致,符合预期	
"with_base_index":false	基础聚合索引和基础明细索引都未创建,与之前保持一致,符合预期	
"base_index_type":["BASE_TABLE_INDEX"]	基础明细索引创建,符合预期	
"base_index_type":["BASE_AGG_INDEX"]	基础聚合索引创建,符合预期	
"base_index_type":["BASE_TABLE_INDEX","BASE_AGG_INDEX"]	基础聚合索引和基础明细索引都创建,符合预期	
"base_index_type":[]	基础聚合索引和基础明细索引都未创建,符合预期	

### 双参数

"with_base_index":true,"base_index_type":["BASE_TABLE_INDEX"]	基础明细索引创建,符合预期	
"with_base_index":false,"base_index_type":["BASE_TABLE_INDEX"]	基础明细索引创建,符合预期	

## 5.3 编辑模型API支持单独添加基础聚合索引或基础明细索引

### 1. 验收标准

编辑模型修改逻辑: 参照PRD中提到的, 也需要支持: 单独添加“基础聚合索引”或“基础明细索引”

产品逻辑和当前保持一致:

- 初次保持模型, 默认添加“基础聚合索引”+“基础明细索引”, 用户可以分别取消勾选
- 再次编辑模型, 不可删除已经添加过的的基础索引, 可以添加未添加过的基础索引
- 开启分层存储时, 必须添加基础明细索引, 不推荐添加基础聚合索引

### 2. 接口

PUT /api/models/semantic

### 3. 现状

修改基础索引存在两个开关

- createIfNotExistTableLayout: 基础明细索引
- createIfNotExistAggLayout:基础聚合索引

```
@Transaction(project = 0)
public BuildBaseIndexResponse updateBaseIndex(String project, CreateBaseIndexRequest request,
        boolean createIfNotExistTableLayout, boolean createIfNotExistAggLayout, boolean isAuo) {
    aclEvaluate.checkProjectWritePermission(project);
    // update =
```

- createIfNotExist 同时控制 两个开关的赋值

<pre>82 83     long curBaseTableLayout = getBaseTableLayout(); 84 -    boolean needCreateBaseTable = createIfNotExist; 85 86     if (exist(preBaseTableLayout) &amp;&amp; notExist(curBaseTableLayout)) { 87         needCreateBaseTable = true; 88     } 89 90     Long curExistBaseAggLayout = getBaseAggLayout(); 90 -    boolean needCreateBaseAgg = createIfNotExist; 91 92     if (exist(preBaseAggLayout) &amp;&amp; notExist(curExistBaseAggLayout)) { 93         needCreateBaseAgg = true; 94     } 95 }</pre>	<pre>95 96     long curBaseTableLayout = getBaseTableLayout(); 97 +    boolean needCreateBaseTable = 98         updateBaseIndexTypes.contains(IndexEntity.Source.BASE_TABLE_INDEX); 98     if (exist(preBaseTableLayout) &amp;&amp; notExist(curBaseTableLayout)) { 99         needCreateBaseTable = true; 100     } 101 102     Long curExistBaseAggLayout = getBaseAggLayout(); 103 +    boolean needCreateBaseAgg = 104         updateBaseIndexTypes.contains(IndexEntity.Source.BASE_AGG_INDEX); 104     if (exist(preBaseAggLayout) &amp;&amp; notExist(curExistBaseAggLayout)) { 105         needCreateBaseAgg = true; 106     } 107 }</pre>
---	--

### 4. 修改逻辑

#### 1. BaseIndexUpdateHelper中的createIfNotExist 修改为 updateBaseIndexTypes

<pre>43 44 private String project; 45 private String modelId; 46 - private boolean createIfNotExist; 47 + private boolean needUpdate; 48 private boolean secondStorageEnabled = false;</pre>	<pre>48 49 private String project; 50 private String modelId; 51 + private List&lt;IndexEntity.Source&gt; updateBaseIndexTypes; 52 private boolean needUpdate; 53 private boolean secondStorageEnabled = false;</pre>
--	---

#### 2. 复用5.2 创建模型API的 处理逻辑得到updateBaseIndexTypes 的值

<pre>2999     copyModel.init(modelManager.getConfig(), project, 3000         modelManager.getCCRelatedModels(copyModel)); 3001 3002     BaseIndexUpdateHelper baseIndexUpdater = new 3003         BaseIndexUpdateHelper(originModel, 3004             request.isWithBaseIndex()); 3005 3006     preProcessBeforeModelSave(copyModel, project);</pre>	<pre>3032     copyModel.init(modelManager.getConfig(), project, 3033         modelManager.getCCRelatedModels(copyModel)); 3034 3035     BaseIndexUpdateHelper baseIndexUpdater = new 3036         BaseIndexUpdateHelper(originModel, 3037             needHandleBaseIndexType(request)); 3038 3039     preProcessBeforeModelSave(copyModel, project);</pre>
--	---

- 修改BaseIndexUpdateHelper 的构造方法

<pre>54     NIndexPlanManager indexPlanManager = 55         NIndexPlanManager.getInstance(KylinConfig.getInstanceFromEnv(), 56             model.getProject()); 57     IndexPlan indexPlan = indexPlanManager.getIndexPlan(model.getId()); 58 59     @@ -61,7 +72,7 @@ public BaseIndexUpdateHelper(NDataModel model, boolean createIfNotExist) { 60     if (needUpdate) { 61         project = model.getProject(); 62         modelId = model.getId(); 63 64 -    this.createIfNotExist = createIfNotExist; 65     preBaseAggLayout = getBaseAggLayout(); 66     preBaseTableLayout = getBaseTableLayout(); 67     }</pre>	<pre>64 +    public BaseIndexUpdateHelper(NDataModel model, List&lt;IndexEntity.Source&gt; 65         updateBaseIndexTypes) { 66         NIndexPlanManager indexPlanManager = 67             NIndexPlanManager.getInstance(KylinConfig.getInstanceFromEnv(), 68                 model.getProject()); 69         IndexPlan indexPlan = indexPlanManager.getIndexPlan(model.getId()); 70 71         if (needUpdate) { 72             project = model.getProject(); 73             modelId = model.getId(); 74 75 +    this.updateBaseIndexTypes = updateBaseIndexTypes; 76     preBaseAggLayout = getBaseAggLayout(); 77     preBaseTableLayout = getBaseTableLayout(); 78     }</pre>
---	--



- 新增构造方法来兼容外部调用
  - createIfNotExist : true, 创建基础聚合索引和基础明细索引
  - createIfNotExist : false, 不创建基础聚合索引和基础明细索引

```

public BaseIndexUpdateHelper(NDataModel model, boolean createIfNotExist) {

```

```

57 public BaseIndexUpdateHelper(NDataModel model, boolean createIfNotExist) {
58 +     this(model,
59 +         createIfNotExist
60 +         ? Lists.newArrayList(IndexEntity.Source.BASE_AGG_INDEX,
          IndexEntity.Source.BASE_TABLE_INDEX)
61 +         : Lists.newArrayList());
62 +     }
63 +

```

## 5. 测试接口

参数	结果	接口截图
回归参数		
"with_base_index":true	基础聚合索引和基础明细索引都创建,符合预期	
"with_base_index":false	基础聚合索引和基础明细索引都未创建,符合预期	
新增的base_index_type 参数测试		
"base_index_type":["BASE_TABLE_INDEX"]	基础明细索引创建,符合预期	
"base_index_type":["BASE_AGG_INDEX"]	基础聚合索引创建,符合预期	
多次编辑模型测试		
在基础明细索引已存在的情况 下"base_index_type":["BASE_AGG_INDEX"]	基础聚合索引创建,并且已存在的基础明细索引不影响,符合预期	
在基础聚合索引已存在的情况 下"base_index_type":["BASE_TABLE_INDEX"]	基础明细索引创建,并且已存在的基础聚合索引不影响,符合预期	