

边栏搜索框下方是动态分类类型的词汇列表，在TEEBB控制台中，我们已经创建了动态分类类型，把动态分类类型的词汇在这里遍历显示出来就可以了。

在TEEBB中所有可以使用字段的类型`EntityType`都可以使用`teebb.core.block.contents`block类型获取对应的Entity对象。打开`TeebbCoreBundle Services`目录下的`Types`目录，分类类型对应的Entity类是`Taxonomy::class`，Doctrine使用`Taxonomy`类存储分类词汇数据。

在`side.html.twig`文件分类词汇代码位置使用`sonata_block_render`渲染分类词汇列表，`entity_class`输入`Taxonomy`全类名，全类名的\要使用双\\来进行转义，这里只想显示动态分类类词的词汇，打开`Taxonomy`类，`Taxonomy`使用`taxonomyType`属性存储分类类型的别名，查询条件这里输入`{taxonomyType: 'news_tags'}`，`news_tags`是动态分类类型的机读别名。这里我们同样需要自定义block模板，输入`template`参数，并且对应创建模板文件。`teebb.core.block.contents`同样给模板文件传入了`contents`参数，这里`contents`是`Taxonomy`类对象的列表。使用`{{dump(contents)}}`进行查看。对`contents`参数循环遍历，复制模板文件代码进行替换。

```
{% for taxonomy in contents %}
    <a href="{{ path('teebb_taxonomy_contents',{slug:taxonomy.slug }) }}"
        class="list-group-item list-group-item-action py-3 h4 mb-0 {% if
app.request.get('_route') == 'teebb_taxonomy_contents' and
        app.request.get('slug') == taxonomy.slug %}active {% endif
%}">
        {{ taxonomy.term }}
    </a>
{% endfor %}
```

`a`标签链接到`teebb_taxonomy_contents`路由，`teebb_taxonomy_contents`路由是TEEBB自带的用于显示某个分类词汇下所有内容的列表页路由，`app`是Symfony提供的Twig全局变量，使用`app.request`对象可以获取当前页面请求数据，我们判断当前页面的路由和参数来判断当前是否需要添加`active`样式类。

回到动态列表页面，刷新，分类词汇已经显示了。词汇列表下方是精选内容列表，这里我们只想获取置顶的内容，同样使用`teebb.core.block.contents`block类型，`entity_class`使用`Content`类可以省略，过滤条件这里使用`Content`类的`boolTop`属性为`true`作为查询条件。同样需要自定义模板传入`template`参数，创建模板文件：

```
{% for content in contents %}
    <div class="bg-light p-3 mb-3">
        <div class="row row-0">
            <div class="col-4">
                {% set imageFields =
get_specify_type_field(content,'content', content.typeAlias,
'referenceImage') %}
                {% for imageFieldData in imageFields %}
                    {% if loop.first %}
                        {% if imageFieldData[0].value %}
                            <a href="{{ path('teebb_content_show', {slug:
content.slug}) }}">
                                {% set imagePath =
imageFieldData[0].value.filePath %}
                                
                            </a>
                        {% endif %}
                    {% endif %}
                {% endfor %}
            </div>
        </div>
    </div>
{% endfor %}
```

```

                <img src={{
imagePath|imagine_filter('list_poster') }} alt="{{ content.title }}">
                </a>
            {% else %}
                <div class="ratio ratio-16x9">
                    <div class="skeleton-image"></div>
                </div>
            {% endif %}
        {% endif %}
    {% endfor %}
</div>
<div class="col-8">
    <div class="card-body">
        <a href="{{ path('teebb_content_show',
{slug:content.slug}) }}" class="text-dark">
            <h4>{{ content.title }}</h4>
        </a>
    </div>
</div>
</div>
</div>
{% endfor %}

```

使用`get_specify_type_field`获取封面图像的第一张作为封面，第一个参数传入`content`对象，第二个参数传入`Content`类的`bundle`名称，第三个参数是`content`的类型别名，第四个传入字段的类型`id`，如果没有封面图像使用一张原型图片进行占位。

在TEEBB控制台上修改一篇内容勾选内容置顶，回到动态列表页面刷新。边栏显示正常了。在左侧内容列表中，分类词汇的链接页面显示当前分类词汇下的所有内容，这个页面也需要修改。和动态列表页面的显示相同，你可以参考动态列表页面，复制`TeebbCoreBundle`中对应的模板进行模板覆盖并在对应代码位置引入`content_list.html.twig`和`side.html.twig`模板文件。

在下一节，我们分析下`teebb.com`网站的课程列表功能，我们将学会使用另一个`block`类型来获取某分类词汇下的所有内容。