关于占位符

- 模板精简，并减少 VScode全局修改文件这一步骤

打开Zotero设置编辑器，将模板用到的占位符对应的 Value 修改成如下形式

title

```
{"content":"#### {{field_contents}}", "field_contents": "{{content}}", "link_style": "no-links"}
```

author

```
{"content":"{{bullet}} Authors:: {{field_contents}}", "link_style": "wiki", "list_separator": ", "}
```

tags

```
{"content":"{{bullet}} Keywords:: {{field_contents}}", "field_contents": "{{content}}", "link_style": "wiki", "list_separator": ", ", "remove_spaces": "true"}
```

publicationTitle（新建）, New -> String -> extensions.mdnotes.placeholder.publicationTitle

```
{"content":"{{bullet}} Journal:: {{field_contents}}", "field_contents": "{{content}}", "link_style": "wiki", "list_separator": ", "}
```

date 新建 New -> String -> extensions.mdnotes.placeholder.date

```
{"content":"{{bullet}} Date:: {{field_contents}}", "field_contents": "{{content}}", "link_style": "no-links", "list_separator": ", "}
```

abstractNote

```
{"content":"#### Abstract\n\n{{field_contents}}\n", "field_contents": "{{content}}", "link_style": "no-links", "list_separator": ", "}
```

pdfAttachments、DOI可以重置

原模板：

```
#### {{title}}  #论文
***
#### Metadata
- Author:: {{author}}
- 作者机构::
- Keywords:: {{tags}}
- Journal:: {{publicationTitle}}
- Date:: {{date}}
- 状态:: #待读
- 对象::
- 方法::
- 分类::
- 内容::
- {{pdfAttachments}}

#### {{abstractNote}}
***

#### 理论
*
#### 内容
*
#### 方法
*
#### 结论
*
```

```
#### 亮点
*
#### 灵感
*


***
#### Zotero links
{{DOI}}
```

修改后:

```

{{title}}  #论文
***
#### Metadata
{{author}}
- 作者机构::
{{tags}}
{{publicationTitle}}
{{date}}
- 状态:: #待读
- 对象::
- 方法::
- 分类::
- 内容::
{{pdfAttachments}}

{{abstractNote}}
***


#### 理论
*
```

````
#### 内容
*
#### 方法
*
#### 结论
*
#### 亮点
*
#### 灵感
*


***
#### Zotero links
{{DOI}}
````

关于笔记可视化

利用dataviewjs显示具体的笔记内容

SQL

~~~
```dataviewjs
let quotesPromises = dv.pages("#论文").map(async (p) => {
    const link = p.file.link;
    const title_array = [
        "#### 理论",
        "#### 内容",
        "#### 方法",
        "#### 结论",
        "#### 亮点",
        "#### 灵感",
        "***",
    ];
~~~

```
    var quotes_array = [];
    return await dv.io.load(p.file.link).then((contents) => {
        for (let index = 0; index < title_array.length - 1; index++) {
            const element_1 = title_array[index];
            const element_2 = title_array[index + 1];

            const indexOfQuotes = contents.indexOf(element_1);

            if (indexOfQuotes > 0) {
                const quoteStart = contents.substr(
                    indexOfQuotes + title_array[0].length
                );
                const quotes = quoteStart
                    .substr(0, quoteStart.indexOf(element_2))
                    .trim();
                quotes_array.push(quotes);
            } else {
                break;
            }
        }
        if (quotes_array.join === "") {
            return [];
        } else {
            //console.log('------');
            //console.log(quotes_array);
            quotes_array.unshift(p.file.link);
            //console.log(quotes_array);
            return Array(quotes_array);
        }
    });
});
let quotes = (await Promise.all(quotesPromises))
    .flat()
```

~~~
    .filter((q) => q.length > 0);

dv.table(["文件", "理论", "内容", "方法", "结论", "亮点", "灵感"], quotes);
```
~~~