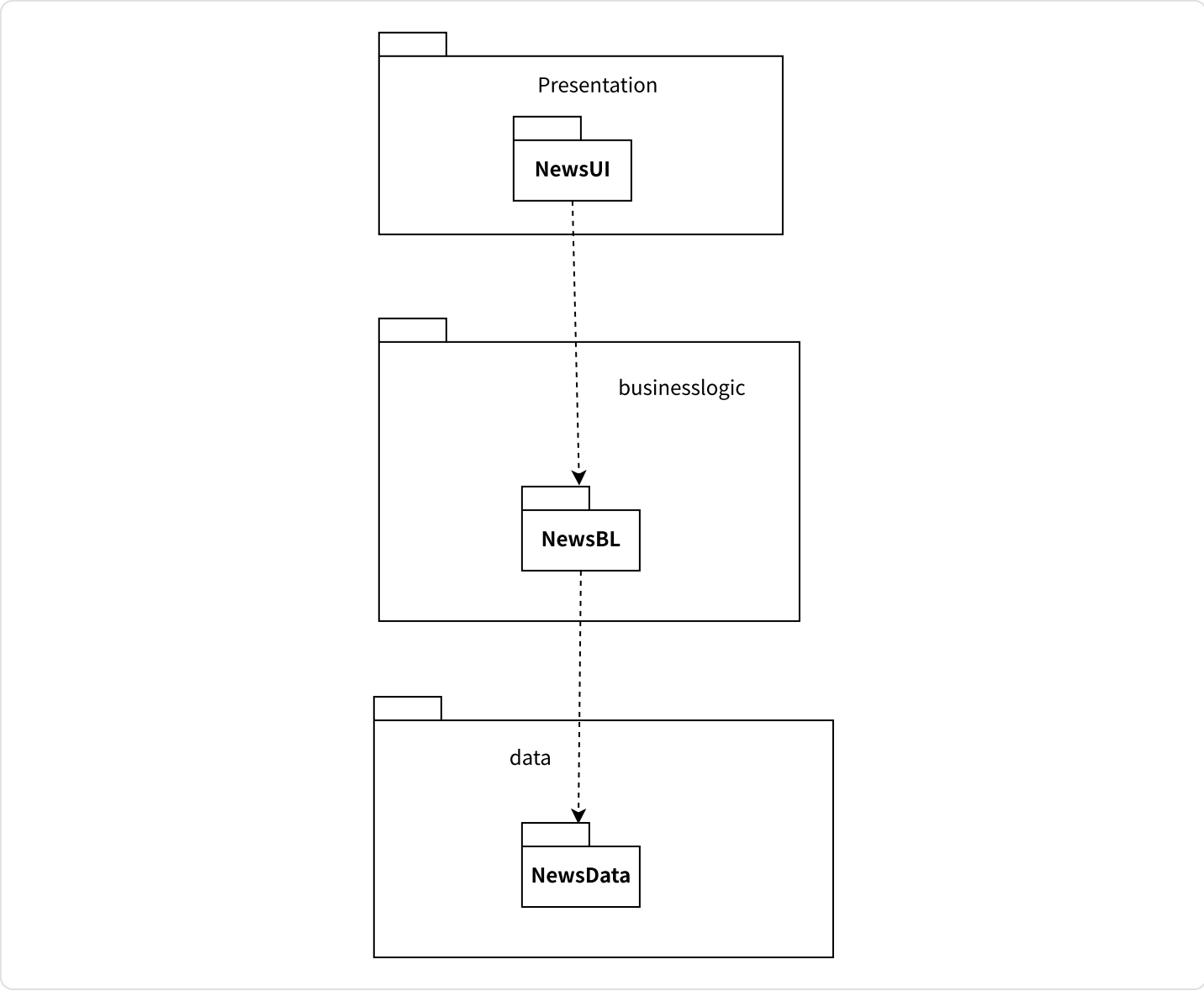


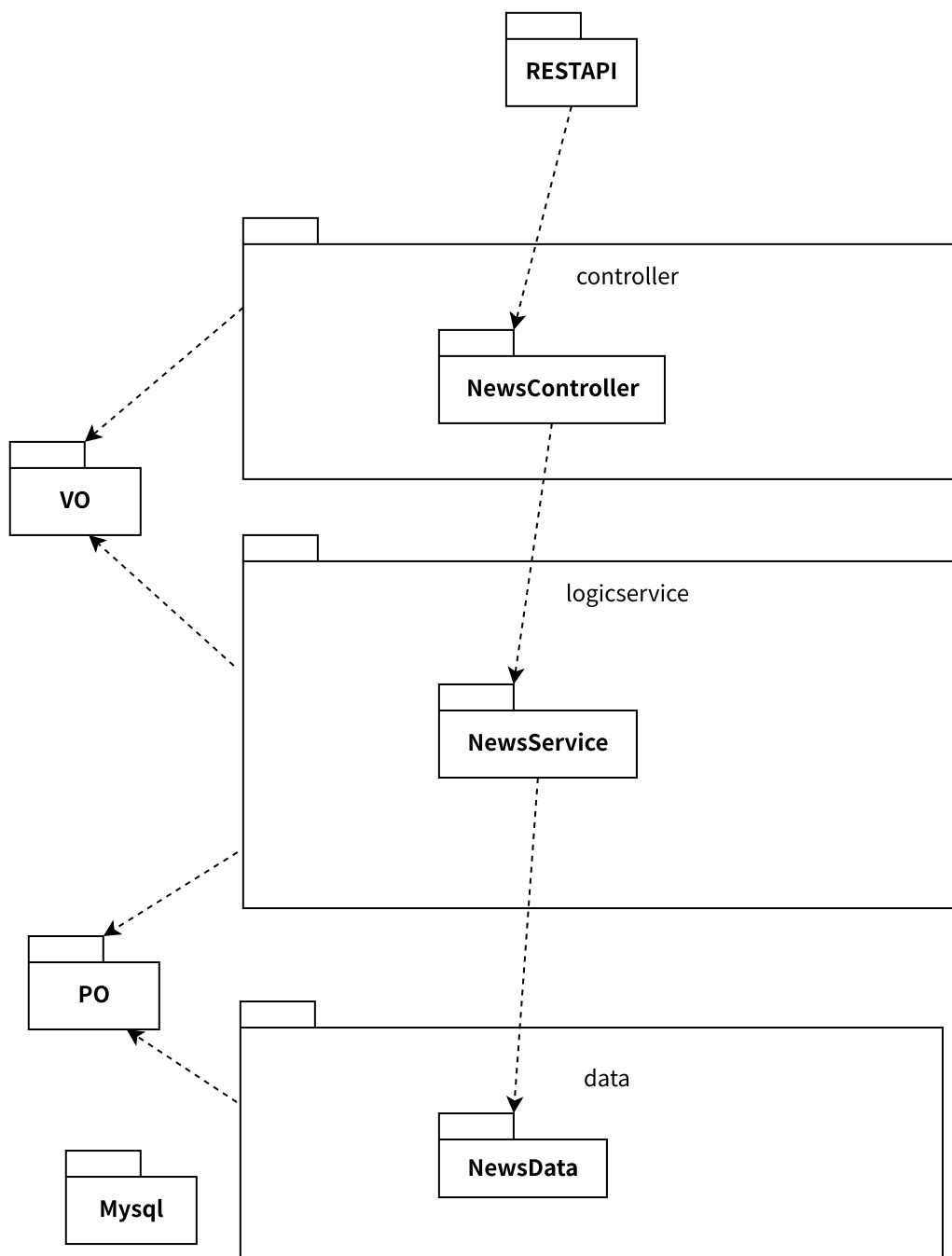
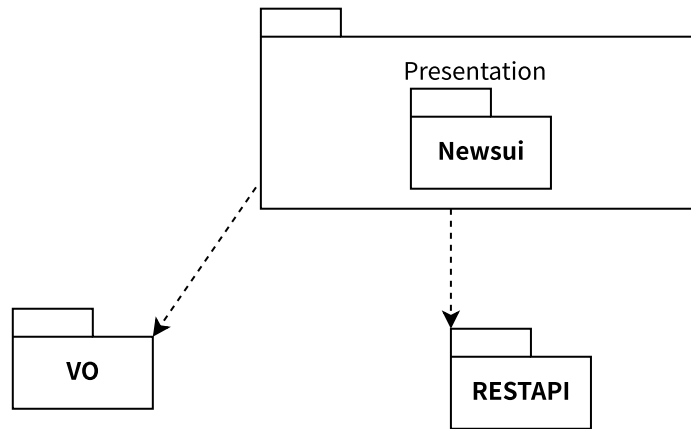
SpY项目设计文档

1. 体系结构设计

1.1 逻辑视角



1.2 组合视角



2. 数据持久化对象

```
1 public class news
2 {
3     @Id
4     private Long id;
5     private String title;//标题
6     private Date date;//日期
7     private String content;//内容
8     private String source;//来源
9     private List<String> categories;//类别
10 }
```

3. 接口

GET数据库记录总数

url:localhost:9010/spy/news/count

参数:

无需传递参数。

返回值:

```
1 {
2     "code":str,
3     "message":str,
4     "result":long
5 }
```

GET单个新闻

News getNews(Long id)

url:localhost:9010/spy/news/{id}

参数:

无需传递参数。请注意：news的id请在url的路径中传递。

返回值:

```
1 {
```

```

2     "code":str,
3     "message":str,
4     "result":{
5         "id":long,
6         "title":str,
7         "date":str,
8         "content":str,
9         "source":str,
10        "categories":[{"category":str}]
11    }
12 }

```

GET搜索及根据时间、来源进行筛选

url:localhost:9010/spy/news/criteria

参数：（注意时间的格式@DateTimeFormat(pattern = "yyyy-MM-dd")）

```

1 {
2     "currentPage":long,//必要参数
3     "pageSize":long,//必要参数
4     "keyword":str,//必要参数，无筛选项时传递空字符串
5     "startDate":str,//不是必要参数，无筛选项时可传递空字符串，传递参数的格式为"yyyy-MM-dd"
6     "endDate":str,//不是必要参数，无筛选项时可传递空字符串，传递参数的格式为"yyyy-MM-dd"
7     "source":str//必要参数，无筛选项时传递空字符串
8 }

```

返回值：

```

1 {
2     "code":str,
3     "message":str,
4     "result":{
5         "list":[{"
6             "id":long,
7             "title":str,
8             "date":str,
9             "content":str,
10            "source":str,
11            "categories":[{"category":str}]
12        }],
13        "totalCount":long

```

```
14     }  
15 }
```

GET获取所有的来源

url:localhost:9010/spy/sources

返回值：

```
1 {  
2     "code":str,  
3     "message":str,  
4     "result":[{"  
5         "source":str  
6     }]  
7 }
```

有关于时间的格式

这个类对应的 JSON 形式可能如下所示：

```
json Copy code  
  
{  
  "id": "some_id_value",  
  "title": "some_title_value",  
  "date": "2024-02-29T12:00:00Z",  
  "content": "some_content_value",  
  "source": "some_source_value",  
  "categories": ["category1", "category2", "category3"]  
}
```

这是一个典型的 JSON 格式，其中包含了类中定义的字段。需要注意的是，日期字段在 JSON 中表示为字符串，并且遵循 ISO 8601 格式。

POST增加新闻

Void addNews(News news);

url:localhost:9010/spy/news

参数：

```
1 {
2   "news":{
3     "title":str,
4     "date":str,
5     "content":str,
6     "source":str,
7     "categories":[{"category":str}]
8   }
9 }
```

返回值：

```
1 {
2   "code":str,
3   "message":str,
4   "result":null
5 }
```

PUT修改新闻

Void updateNews (News news);

curl:localhost:9010/spy/news/{id}

参数：

请注意：news的id请在url的路径中传递。

```
1 {
2   "news":{
3     "title":str,
4     "date":str,
5     "content":str,
6     "source":str,
7     "categories":[{"category":str}]
8   }
9 }
```

返回值：

```
1  {  
2    "code":str,  
3    "message":str,  
4    "result":null  
5  }
```

DELETE删除新闻

Void deleteNews(Long id);

url:localhost:9010/spy/news/{id}

参数：

无需传递参数。请注意：news的id请在url的路径中传递。

返回值：

```
1  {  
2    "code":str,  
3    "message":str,  
4    "result":null  
5  }
```