

HW 6

FulltextSearch 功能实现

Doc 与 Index 创建

为每本书创建了一个 `json` 格式的文件，存放其 id、作者、简介信息。

Java

```
public static void create_docs_files(Integer id, String description, String author, Boolean update) {
    JSONObject jsonObject = new JSONObject();
    jsonObject.put("book_id", id);
    jsonObject.put("author", author);
    jsonObject.put("description", description);
    jsonObject.put("")
    String filePath = FilesPositionConfig.bookid2filePath(id);
    try {
        writeFile(filePath, jsonObject);
        String[] args = {"-index", FilesPositionConfig.indexPath, "-docs", FilesPositionConfig.bookid2filePath(id),
            update ? "-update" : "-not-update"};
        IndexFiles.index_interface(args);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

对每本书调用上述方法

Java

```
public void createIndex() {
    List<Book> books = bookDao.getAllBooks();
    for (Book book : books){
        ReadWriteFiles.create_docs_files(book.getId(), book.getBookDescription(), book.getAuthor(),true);
    }
}
```

同时在 `INDEX` 目录下创建索引

Java

```
public static void index_interface(String[] args) {
    ...
    ...
    // 省略path识别和文件检验
    try {
        System.out.println("Indexing to directory '" + indexPath + "'...");
        Directory dir = FSDirectory.open(Paths.get(indexPath));
        Analyzer analyzer = new StandardAnalyzer();
        IndexWriterConfig iwc = new IndexWriterConfig(analyzer);
        if (create) {
            iwc.setOpenMode(OpenMode.CREATE);
        } else {
            iwc.setOpenMode(OpenMode.CREATE_OR_APPEND);
        }

        IndexWriter writer = new IndexWriter(dir, iwc);
        indexDocs(writer, docDir);
        writer.close();
        Date end = new Date();
    } catch (IOException e) {
        System.out.println(" caught a " + e.getClass() +
```

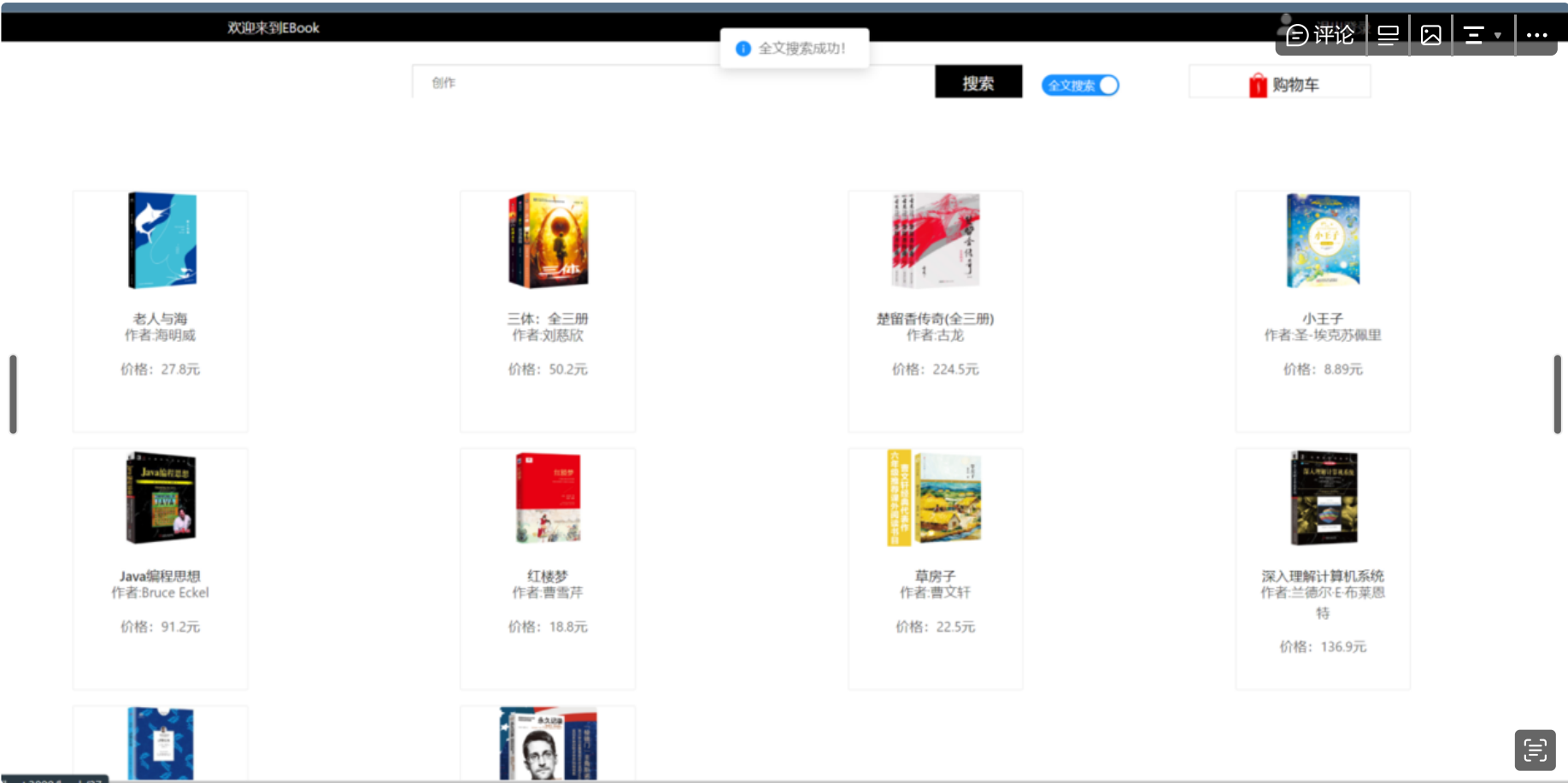
```
        "\n with message: " + e.getMessage());
    }
}
```

效果演示

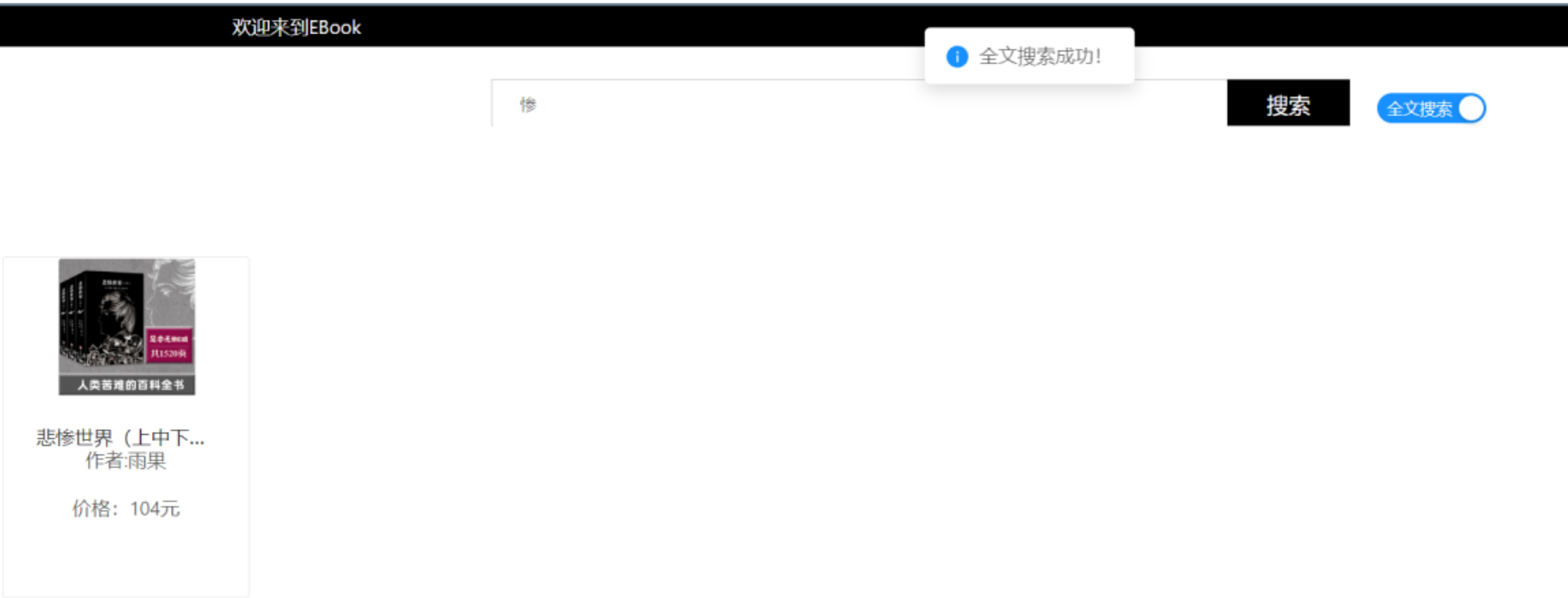
- 按作者查询



- 按书籍简介查询



- 按书名查询



Field 的设计

总共添加了两个 Field：path 用于记录文件的路径，content 承载书籍的具体内容，。

- path : **StringField**

由于在索引之后需要返回满足条件的文件的路径因此添加了这个 Field，类型为 **StringField**，理由是文件的路径是一个整体。索引时不需要进行分词，需要存储。

- content: **TextField**

在 content 里记录有书籍的 id、作者、简介信息，利用这个 field 来进行关键字的模糊查询。

选择 **TextField** 的类型，对内容进行分词，这样文件的文本就会被标记和索引，另外由于只需要查到对应的文件，因此无需对查找的的内容进行存储。