ＳＴＥＰ０３

Ｓｐｒｉｎｇフレームワーク入門

コードで理解するＳｐｒｉｎｇの世界

DIによる（えせ）書籍一覧画面

目次

[1 ドメインクラス準備 1](#_Toc403735079)

[1.1 書籍クラス作成 1](#_Toc403735081)

[2 ＤＩ ・・ （えせ）書籍一覧サービスをインジェクション 3](#_Toc403735082)

[2.1 （えせ）書籍一覧サービスクラス新規作成 3](#_Toc403735084)

[2.2 書籍管理コントロールクラスにDIする 5](#_Toc403735085)

[2.3 書籍一覧画面Viewでモデルオブジェクトを表示させる 6](#_Toc403735086)

[2.4 画面表示テスト 7](#_Toc403735087)

[3 ＤＩ ・・・ （えせ）書籍一覧Ｄａｏをインジェクション 8](#_Toc403735088)

[3.1 （えせ）書籍一覧Ｄａｏクラス新規作成 8](#_Toc403735090)

[3.2 書籍管理サービス実装クラスにインジェクション 10](#_Toc403735091)

[3.3 画面表示テスト 11](#_Toc403735092)

# ドメインクラス準備

書籍情報を保持するドメインクラスを作成する。



## ドメインクラス （book：書籍クラス） 作成

１）ドメイン書籍クラス（Book.java）新規作成

|  |
| --- |
| /src/main/java で右クリック → New → Class    ● Java Class 画面  ---------------  Package: 「jp.sample.bookmgr.biz.domain」  Name: 「Book」 と入力して Finish  ---------------  クラスのひな形が自動作成される |

２）書籍クラス（Book.java）新規作成

|  |
| --- |
| /src/main/java/jp.sample.bookmgr.biz.domain.Book.java （新規作成） |
| package jp.sample.bookmgr.biz.domain;  /\*\*  \* 書籍クラス  \* @author 長住@NTT-AT  \* @version 1.0  \*/  public class Book {  /\*\*  \* 書籍ID  \*/  private int id = 0;    /\*\*  \* 書籍ISBN  \*/  private String isbn = "";    /\*\*  \* 書籍名  \*/  private String name = "";    /\*\*  \* 書籍価格  \*/  private int price = 0;    /\*\*  \* 書籍IDゲッタ・セッタ  \*/  public int getId() { return id; }  public void setId(int id) { this.id = id; }    /\*\*  \* ISBNゲッタ・セッタ  \*/  public String getIsbn() { return isbn; }  public void setIsbn(String isbn) { this.isbn = isbn; }  /\*\*  \* 書籍名ゲッタ・セッタ  \*/  public String getName() { return name; }  public void setName(String name) { this.name = name; }    /\*\*  \* 価格ゲッタ・セッタ  \*/  public int getPrice() { return price; }  public void setPrice(int price) { this.price = price; }  } |

# ＤＩ ・・ （えせ）書籍一覧サービスをインジェクション

書籍情報一覧サービスオブジェクトを書籍画面クラスにインジェクションしてみる。



## （えせ）書籍一覧サービスクラス新規作成

１）書籍一覧サービスインターフェース（ListBookService.java）新規作成

|  |
| --- |
| /src/main/java で右クリック → New → Interface    ● Java Class 画面  ---------------  Package: 「jp.sample.bookmgr.biz.service」  Name: 「ListBookService」 と入力して Finish  ---------------  インターフェースのひな形が自動作成される |

２）書籍一覧サービスインターフェース（ListBookService.java）コーディング

|  |
| --- |
| /src/main/java/jp.sample.bookmgr.biz.service.ListBookService.java （新規作成） |
| package jp.sample.bookmgr.biz.service;  import java.util.List;  import jp.sample.bookmgr.biz.domain.Book;  /\*\*  \* 書籍一覧サービスクラスインターフェース  \* @author 長住@NTT-AT  \* @version 1.0  \*/  public interface ListBookService {  /\*\*  \* 書籍一覧取得サービス  \* @return 書籍一覧情報  \*/  public List<Book> getBookList() throws Exception;  } |

３）（えせ）書籍一覧サービス実装クラス（ListBookServiceImpl.java）新規作成

|  |
| --- |
| bookmgr/src/main/java で右クリック → New → Class  ● Java Class 画面  ---------------  Package: 「jp.sample.bookmgr.biz.service」  Name: に「ListBookServiceImpl」 と入力  Interfaces: に「jp.sample.bookmgr.biz.service.ListBookService」を選択してFinish  ---------------  クラスのひな形が自動作成される |

４）（えせ）書籍一覧サービス実装クラス（ListBookServiceImpl.java）コーディング

|  |
| --- |
| /src/main/java/jp.sample.bookmgr.biz.service.ListBookServiceImpl.java （新規作成） |
| package jp.sample.bookmgr.biz.service;  import java.util.ArrayList;  import java.util.List;  import org.springframework.stereotype.Service;  import jp.sample.bookmgr.biz.domain.Book;  /\*\*  \* 書籍一覧サービス実装クラス  \* @author 長住@NTT-AT  \* @version 1.0  \*/  @Service // サービスクラスとしてDI可能というアノテーションを宣言  public class ListBookServiceImpl implements ListBookService {  /\*\*  \* 書籍一覧取得サービス  \* @return 書籍一覧情報  \*/  @Override  public List<Book> getBookList() throws Exception {  // 本当はデータベースからデータを取得する必要があるが、 レベル１では簡易にえせリストを作成する  List<Book> bookList = new ArrayList<Book>();  Book book1 = new Book();  book1.setId(1);  book1.setIsbn("ISBN-0001");  book1.setName("テスト書籍１");  book1.setPrice(1000);  bookList.add(book1);    Book book2 = new Book();  book2.setId(2);  book2.setIsbn("ISBN-0002");  book2.setName("テスト書籍２");  book2.setPrice(2000);  bookList.add(book2);    Book book3 = new Book();  book3.setId(3);  book3.setIsbn("ISBN-0003");  book3.setName("テスト書籍３");  book3.setPrice(3000);  bookList.add(book3);    return bookList;  }  } |

＜Import挿入方法＞

Import文を記述せずに、とりあえずクラス定義を先に作成してみましょう。

当然エラーとなりますが、Eclipse の機能を使えば Import 文を後から自動的に

挿入させることができます！ぜひこのワザを覚えましょう！

C:\Users\nagazumi\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\W71ETJYL\MC900438000[1].wmf

## 書籍管理コントロールクラスにDIする

１）書籍管理コントローラクラスの修正、書籍一覧サービスをインジェクションして使う

|  |
| --- |
| /src/main/java/jp.sample.bookmgr.web.controller.BookController.java |
| package jp.sample.bookmgr.web.controller;  import java.util.List;  import org.slf4j.Logger;  import org.slf4j.LoggerFactory;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Controller;  import org.springframework.ui.Model;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RequestMethod;  import jp.sample.bookmgr.biz.domain.Book;  import jp.sample.bookmgr.biz.service.ListBookService;  /\*\*  \* 書籍管理処理コントローラクラス  \*  \* @author 長住@NTT-AT  \* @version 1.0  \*/  @Controller // ”コントローラとしてDI可能" というアノテーションを宣言  public class BookController {  /\*\*  \* ロガー  \*/  private static final Logger logger = LoggerFactory.getLogger(BookController.class);    /\*\*  \* 書籍一覧処理を実装したビジネスロジックサービス  \*/  @Autowired // ListBookServiceオブジェクトをインジェクション  ListBookService listBookService;    /\*\*  \* 書籍一覧画面コントローラ  \*  \* @param model Spring MVC Modelオブジェクト  \* @return 画面JSP名  \*/  @RequestMapping(value = "/listbook", method = RequestMethod.GET)  public String listBook(Model model) throws Exception {    logger.debug("listBook() start");    // 書籍一覧取得ロジック処理  List<Book> books = listBookService.getBookList();  // 書籍一覧情報をモデルに登録  model.addAttribute("books", books);    // 画面表示に listbook.jsp を呼び出す  return "listbook";  }  ・・・  ・・・ |

## 書籍一覧画面Viewでモデルオブジェクトを表示させる

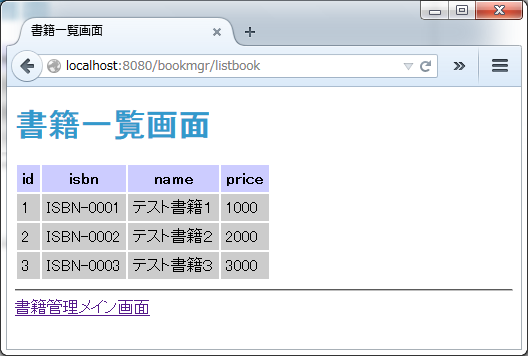
１）書籍一覧画面View（listbook.jsp）修正

|  |
| --- |
| /src/main/webapp/WEB-INF/views/listbook.jsp |
| <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/bookmgr.css" />  <title>書籍一覧画面</title>  </head>  <body>  <h1>書籍一覧画面</h1>  <table>  <tr><th>id</th><th>isbn</th><th>name</th><th>price</th></tr>  <c:forEach items="${books}" var="book" varStatus="status">  <tr>  <td><c:out value="${book.id}"/></td>  <td><c:out value="${book.isbn}"/></td>  <td><c:out value="${book.name}"/></td>  <td><c:out value="${book.price}"/></td>  </tr>  </c:forEach>  </table>  <hr />  <a href="main">書籍管理メイン画面</a>  </body>  </html> |

## 画面表示テスト

プログラムを起動して、書籍一覧画面を表示させる。

うまくコードが実行されるか、うまく画面が表示されるかを確認。



# ＤＩ ・・・ （えせ）書籍一覧Ｄａｏをインジェクション

書籍情報一覧Ｄａｏオブジェクトを書籍一覧サービスクラスにインジェクションしてみる。



## （えせ）書籍一覧Ｄａｏクラス新規作成

１）書籍一覧Ｄａｏクラスのインターフェース（ListBookDao.java）新規作成

|  |
| --- |
| /src/main/java で右クリック → New → Interface    ● Java Class 画面  ---------------  Package: 「jp.sample.bookmgr.biz.dao」  Name: 「ListBookDao」 と入力して Finish  ---------------  インターフェースのひな形が自動作成される |

２）書籍一覧Ｄａｏクラスのインターフェース（ListBookDao.java）コーディング

|  |
| --- |
| /src/main/java/jp.sample.bookmgr.biz.dao.ListBookDao.java （新規作成） |
| package jp.sample.bookmgr.biz.dao;  import java.util.List;  import jp.sample.bookmgr.biz.domain.Book;  /\*\*  \* 書籍一覧データアクセスクラスインターフェース  \* @author 長住@NTT-AT  \* @version 1.0  \*/  public interface ListBookDao {  /\*\*  \* 書籍データベースから書籍一覧データを取得するメソッド  \* @return 書籍一覧情報  \*/  public List<Book> getBookList() throws Exception;  } |

３）（えせ）書籍一覧Ｄａｏ実装クラス（ListBookDaoImpl.java）新規作成

|  |
| --- |
| bookmgr/src/main/java で右クリック → New → Class  ● Java Class 画面  ---------------  Package: 「jp.sample.bookmgr.biz.dao」  Name: に「ListBookDaoImpl」 と入力  Interfaces: に「jp.sample.bookmgr.biz.dao.ListBookDao」を選択してFinish  ---------------  クラスのひな形が自動作成される |

４）（えせ）書籍一覧Ｄａｏ実装クラス（ListBookDaoImple.java）コーディング

|  |
| --- |
| /src/main/java/jp.sample.bookmgr.biz.dao.ListBookDaoImpl.java （新規作成） |
| package jp.sample.bookmgr.biz.dao;  import java.util.ArrayList;  import java.util.List;  import org.springframework.stereotype.Repository;  import jp.sample.bookmgr.biz.domain.Book;  /\*\*  \* 書籍一覧データアクセス実装クラス  \* @author 長住@NTT-AT  \* @version 1.0  \*/  @Repository // リポジトリ（Dao）クラスとしてDI可能というアノテーションを宣言  public class ListBookDaoImpl implements ListBookDao {  /\*\*  \* 書籍データベースから書籍一覧データを取得するメソッド  \* @return 書籍一覧情報  \*/  @Override  public List<Book> getBookList() throws Exception {    // 本当はデータベースからデータを取得する必要があるが、 レベル１では簡易にえせリストを作成する  List<Book> bookList = new ArrayList<Book>();  Book book1 = new Book();  book1.setId(1);  book1.setIsbn("ISBN-0001");  book1.setName("テスト書籍１");  book1.setPrice(1000);  bookList.add(book1);    Book book2 = new Book();  book2.setId(2);  book2.setIsbn("ISBN-0002");  book2.setName("テスト書籍２");  book2.setPrice(2000);  bookList.add(book2);    Book book3 = new Book();  book3.setId(3);  book3.setIsbn("ISBN-0003");  book3.setName("テスト書籍３");  book3.setPrice(3000);  bookList.add(book3);    return bookList;  }  } |

## 書籍管理サービス実装クラスにインジェクション

|  |
| --- |
| /src/main/java/jp.sample.bookmgr.biz.service.ListBookServiceImpl.java |
| package jp.sample.bookmgr.biz.service;  // import java.util.ArrayList; （ここ削除）  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import jp.sample.bookmgr.biz.dao.ListBookDao;  import jp.sample.bookmgr.biz.domain.Book;  /\*\*  \* 書籍一覧サービス実装クラス  \* @author 長住@NTT-AT  \* @version 1.0  \*/  @Service // サービスクラスとしてDI可能というアノテーションを宣言  public class ListBookServiceImpl implements ListBookService {  /\*\*  \* 書籍一覧取得DAO  \*/  @Autowired // ListBookDaoオブジェクトをインジェクション  ListBookDao listBookDao;  /\*\*  \* 書籍一覧取得サービス  \* @return 書籍一覧情報  \*/  getBookList()  の中身は大幅に修正  @Override  public List<Book> getBookList() throws Exception {  // 書籍一覧を取得  return listBookDao.getBookList();  }  } |

## 画面表示テスト

プログラムを起動して書籍一覧画面を表示する。

うまくコードが実行されるか、うまく画面が表示されるかを確認。

