

CRUD application Using Servlet JSP JDBC, BootStrap

Step 1: create table and populate some records

```
-----  
create table books(id int not null auto_increment,  
isbn varchar(20) not null,  
title varchar(40) not null,  
author varchar(80) not null,  
pubDate date not null, price double not null,  
primary key (id), unique key (isbn));  
  
populate some records  
insert into books(isbn, title, author, pubDate, price )  
values ('12PZ', 'C basics', 'ekta', '1011-12-22', 345);
```

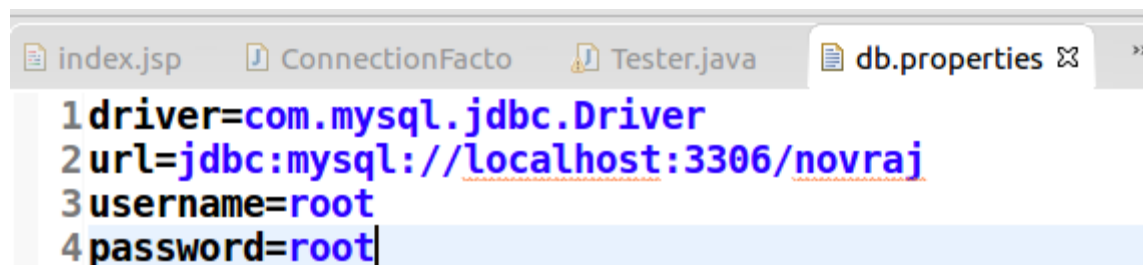
Step 2: Create DAO, DTO

```
public class Book {
    private int id;
    private String isbn;
    private String title;
    private String author;
    private Date pubDate;
    private double price;

    import java.util.*;
    public interface BookDao {
        public List<Book> getAll();
        public Book add(Book book);
        public Book delete(int bookId);
        public Book update(Book book);
        public Book getById(int bookId);
    }
```

Step 3: Create connection factory

```
public static Connection getConnection(){  
    InputStream is=ConnectionFactory.class.getClassLoader().getResourceAsStream("db.properties");  
    Properties prop=new Properties();  
    try {  
        prop.load(is);  
    } catch (IOException e1) {  
        e1.printStackTrace();  
    }  
    try {  
        Class.forName(prop.getProperty("driver"));  
        //System.out.println("driver is loaded....");  
    } catch (ClassNotFoundException e) {  
        e.printStackTrace();}  
  
    try {  
        connection=DriverManager.getConnection(prop.getProperty("url"),  
            prop.getProperty("username"),prop.getProperty("password"));  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    return connection;  
}
```



The screenshot shows an IDE window with four tabs: index.jsp, ConnectionFacto, Tester.java, and db.properties. The db.properties tab is active, displaying the following properties:

```
1driver=com.mysql.jdbc.Driver  
2url=jdbc:mysql://localhost:3306/novraj  
3username=root  
4password=root
```

Step 4: Create getAll books

```
@Override
public List<Book> getAll() {
    //isbn, title, author, pubDate, price
    List<Book> books=new ArrayList<>();
    try {
        Statement stmt=connection.createStatement();
        ResultSet rs=stmt.executeQuery("select * from books");
        while(rs.next()){
            books.add(new Book(rs.getInt("id"), rs.getString("isbn"),
                                rs.getString("title"), rs.getString("author"),
                                rs.getDate("pubDate"), rs.getDouble("price")));
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }

    return books;
}
```

Step 5: Create add book

@Override

```
public Book add(Book book) {  
    //isbn, title, author, pubDate, price  
    String query_insert="insert into books(isbn,title, author, pubDate, price) values (?,?,,?,?,?)";  
    try {  
        PreparedStatement pstmt=connection.prepareStatement(query_insert,  
            Statement.RETURN_GENERATED_KEYS);  
  
        pstmt.setString(1, book.getIsbn());  
        pstmt.setString(2, book.getTitle());  
        pstmt.setString(3, book.getAuthor());  
        pstmt.setDate(4, new Date(book.getPubDate().getTime()));  
        pstmt.setDouble(5, book.getPrice());  
        int numberOfRowsEffected=pstmt.executeUpdate();  
  
        if(numberOfRowEffected> 0){  
            ResultSet rs=pstmt.getGeneratedKeys();  
            rs.next();  
            book.setId(rs.getInt(1));  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    return book;  
}
```

Step 6: get book by id

```
@Override
public Book delete(int bookId) {
    // TODO Auto-generated method stub
    return null;
}

@Override
public Book getById(int bookId) {
    Book book=null;
    //id | isbn | title| author | pubDate| price
    String get_book_query="select * from books where id=?";
    try {
        PreparedStatement pstmt=connection.prepareStatement(get_book_query);
        pstmt.setInt(1, bookId);
        ResultSet rs=pstmt.executeQuery();
        if(rs.next()){
            book=new Book(rs.getInt(1), rs.getString(2),
                rs.getString(3), rs.getString(4), rs.getDate(5), rs.getDouble(6));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return book;
}
```

Step 7: delete a book

```
@Override
public Book delete(int bookId) {
    Book book=getById(bookId);
    if(book!=null){
        String delete_book_query="delete from books where id=?";
        try {
            PreparedStatement pstmt=connection.prepareStatement(delete_book_query);
            pstmt.setInt(1, bookId);
            pstmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return book;
}
```


Step 7: update a book

```
@Override
public Book update(Book book) {
    ////isbn, title, author, pubDate, price
    String update_book_query="update books set price=? where id=?";
    try {
        PreparedStatement pstmt=connection.prepareStatement(update_book_query);
        pstmt.setDouble(1, book.getPrice());
        pstmt.setInt(2, book.getId());
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return book;
}
```

Step 8: display all books controller

```
@WebServlet("/bookController")
public class BookController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    private BookDao dao=new BookDaoImpl();

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {

        List<Book> getAllBooks=dao.getAll();
        request.setAttribute("books", getAllBooks);

        RequestDispatcher rd=request.getRequestDispatcher("all_books.jsp");
        rd.forward(request, response);

    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {

    }
```

Step 8: display all books -Jsp

```
<table border="1">
  <thead>
    <tr>
      <th>id</th>
      <th>isbn</th>
      <th>title</th>
      <th>author</th>
      <th>pubDate</th>
      <th>price</th>
    </tr>
  </thead>
  <tbody>
    <c:forEach var="book" items="${books}">
      <tr>
        <td><c:out value="${book.id}"/></td>
        <td><c:out value="${book.isbn}"/></td>
        <td><c:out value="${book.title}"/></td>
        <td><c:out value="${book.author}"/></td>
        <td><c:out value="${book.pubDate}"/></td>
        <td><c:out value="${book.price}"/></td>
      </tr>
    </c:forEach>
  </tbody>
</table>
```

← → ↻ ⓘ localhost:8080/prj

id	isbn	title	author	pubDate	price
1	12AZ	12AZ	12AZ	1011-12-11	345.0
3	12PZ	12PZ	12PZ	1011-12-22	560.0

Step 9: add book controller

```
return rd(request, response);  
} else if ("add".equals(action)) {  
    RequestDispatcher rd = request.getRequestDispatcher("add_book.jsp");  
    rd.forward(request, response);  
}
```

```
<form action="bookController" method="post">  
    <input type="hidden" name="id" /><br/>  
    Enter isbn : <input type="text" name="isbn" /><br/>  
    Enter title : <input type="text" name="title" /><br/>  
    Enter author: <input type="text" name="author" /><br/>  
    Enter pubDate: <input type="text" name="pubDate" /><br/>  
    Enter price: <input type="text" name="price" /><br/>  
    <input type="submit"/>
```

Step 9: add book controller

```
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {

    String isbn = request.getParameter("isbn");
    String title = request.getParameter("title");
    String author = request.getParameter("author");
    Date pubDate = null;
    try {
        pubDate = new SimpleDateFormat("dd/MM/yyyy").parse(request
            .getParameter("pubDate"));
    } catch (ParseException e) {
        e.printStackTrace();
    }
    Double price = Double.parseDouble(request.getParameter("price"));

    Book book = new Book(isbn, title, author, pubDate, price);

    dao.add(book);

    response.sendRedirect("bookController?action=allbooks");
}
```

Step 10: delete book

```
} else if ("delete".equals(action)) {  
    int id = Integer.parseInt(request.getParameter("id"));  
    dao.delete(id);  
    response.sendRedirect("bookController?action=allbooks");  
}
```

Step 11: update book

```
} else if ("update".equals(action)) {  
    int id = Integer.parseInt(request.getParameter("id"));  
    Book book = dao.getById(id);  
    request.setAttribute("book", book);  
    request.getRequestDispatcher("add_book.jsp").forward(request,  
        response);  
}
```

```
protected void doPost(HttpServletRequest request,  
    HttpServletResponse response) throws ServletException, IOException {  
    Integer id = Integer.parseInt(request.getParameter("id"));  
    String isbn = request.getParameter("isbn");  
    String title = request.getParameter("title");  
    String author = request.getParameter("author");  
    Date pubDate = null;  
    try {  
        pubDate = new SimpleDateFormat("dd/MM/yyyy").parse(request  
            .getParameter("pubDate"));  
    } catch (ParseException e) {  
        e.printStackTrace();  
    }  
    Double price = Double.parseDouble(request.getParameter("price"));  
  
    Book book = new Book(id, isbn, title, author, pubDate, price);  
    if (id == 0)  
        dao.add(book);  
    else  
        dao.update(book);  
    response.sendRedirect("bookController?action=allbooks");  
}
```

Step 11: update book

```
<form action="bookController" method="post">
<input type="hidden" name="id" value="<c:out value='${book.id}'/>"><br/>
Enter isbn : <input type="text" name="isbn" value="<c:out value='${book.isbn}'/>"><br/>
Enter title : <input type="text" name="title" value="<c:out value='${book.title}'/>"><br/>
Enter author: <input type="text" name="author" value="<c:out value='${book.author}'/>"><br/>
Enter pubDate: <input type="text" name="pubDate" value="<c:out value='${book.pubDate}'/>"><br/>
Enter price: <input type="text" name="price" value="<c:out value='${book.price}'/>"><br/>
<input type="submit"/>
```

← → ↻ ⓘ localhost:8080/prs/bookController?action=allbook

id	isbn	title	author	pubDate	price		
3	12PZ	12PZ	12PZ	1011-12-22	560.0	Update	Delete
4	r1234	r1234	r1234	2011-11-11	890.0	Update	Delete

[addbook](#)

Step 12: show all books with links

```
9 <tbody>
0 <c:forEach var="book" items="${books}" >
1   <tr>
2     <td><c:out value="${book.id}"/></td>
3     <td><c:out value="${book.isbn}"/></td>
4     <td><c:out value="${book.title}"/></td>
5     <td><c:out value="${book.author}"/></td>
6     <td><c:out value="${book.pubDate}"/></td>
7     <td><c:out value="${book.price}"/></td>
8
9     <td><a href="bookController?action=update&id=<c:out value="${book.id}"/>">Update</a></td>
0     <td><a href="bookController?action=delete&id=<c:out value="${book.id}"/>">Delete</a></td>
1   </tr>
2 </c:forEach>
3 </tbody>
```