BookController --- HAS --> BooService ----HAS----> BookDAO

```
class BookCOntroller {
BookService bs;
bs = BookService();
bs.m1();
bs.m2();
}
```

```
class BookService {
BookDAO bd;
bd = new BookDAO();
bd.m1();
bd.m2();
}
```

```
class BookDAO {
m1()…
m2()…


}
```

Controller Layer

```
bookcontroller

BookService bs;
bs = new BookServiceImpl();
bs.m1();
b2.m2();
```

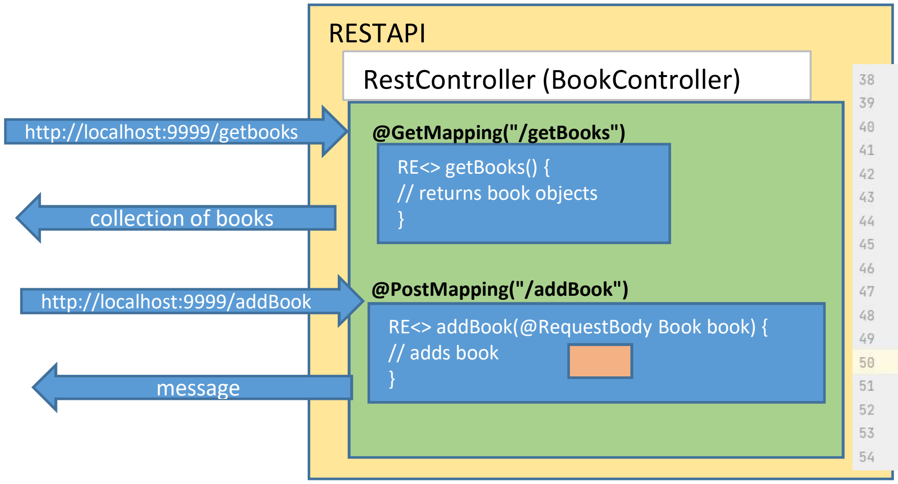
Service Layer

```
interface BookService
        void m1();
        void m2();

class BookServiceImpl impl..
        BookDAO bookDao;
        bookDao = new BookDAOImpl();
        void m1 () { }
        void m2() { }
```

DAO Layer

```
interface BookDAO



class BookDAOImpl impl…
```

Yesterdays demo 22/12/2022



RESTAPI

RestController (BookController)

http://localhost:9999/getbooks

collection of books

http://localhost:9999/addBook

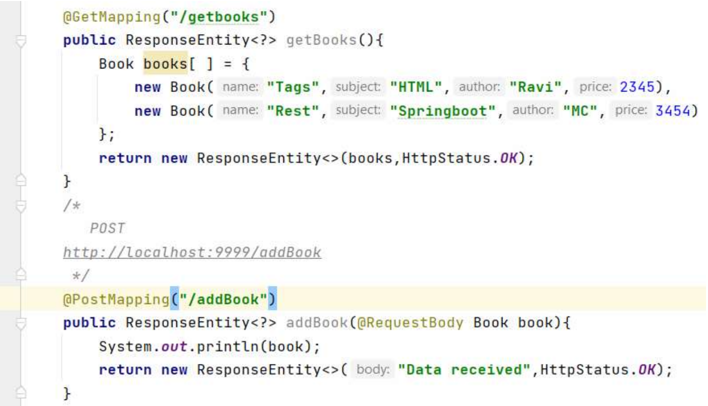
message

@GetMapping("/getBooks")

RE<> getBooks() {
// returns book objects
}

@PostMapping("/addBook")

RE<> addBook(@RequestBody Book book) {
// adds book
}



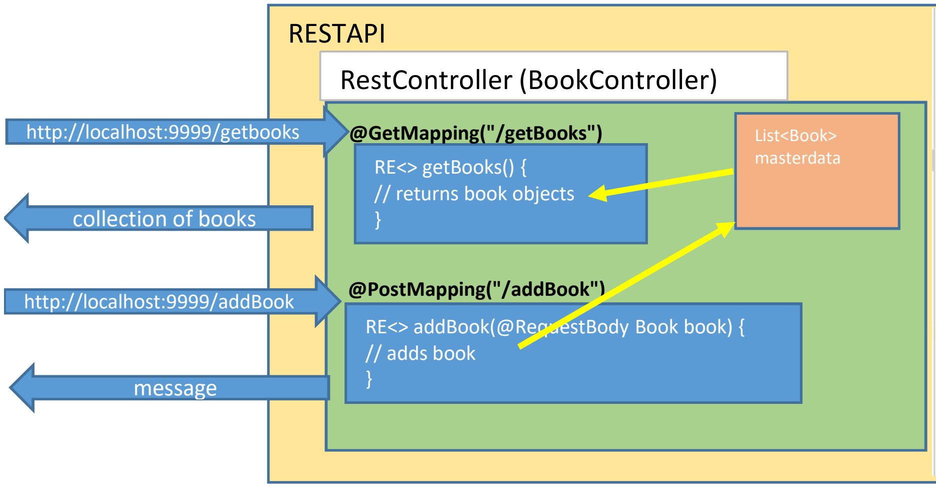
```
@GetMapping("/getbooks")
public ResponseEntity<?> getBooks(){
    Book books[ ] = {
        new Book( name: "Tags", subject: "HTML", author: "Ravi", price: 2345),
        new Book( name: "Rest", subject: "Springboot", author: "MC", price: 3454)
    };
    return new ResponseEntity<>(books,HttpStatus.OK);
}
/*
   POST
http://localhost:9999/addBook
 */
@PostMapping("/addBook")
public ResponseEntity<?> addBook(@RequestBody Book book){
    System.out.println(book);
    return new ResponseEntity<>( body: "Data received",HttpStatus.OK);
}
```

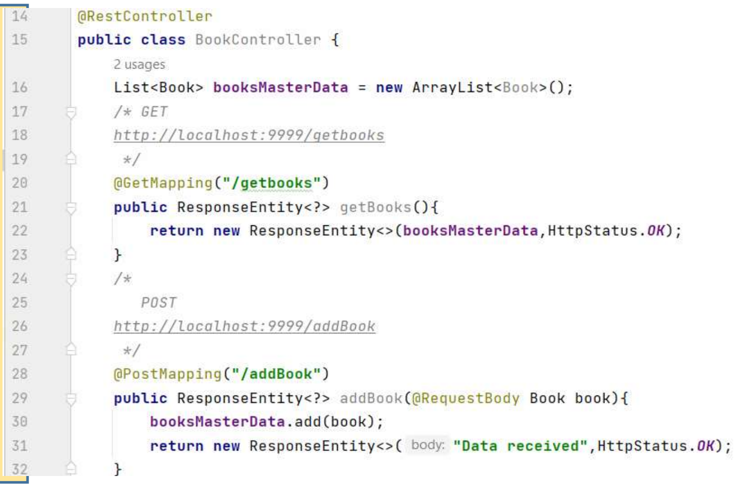
Today 23/12/2022

collection created at controller

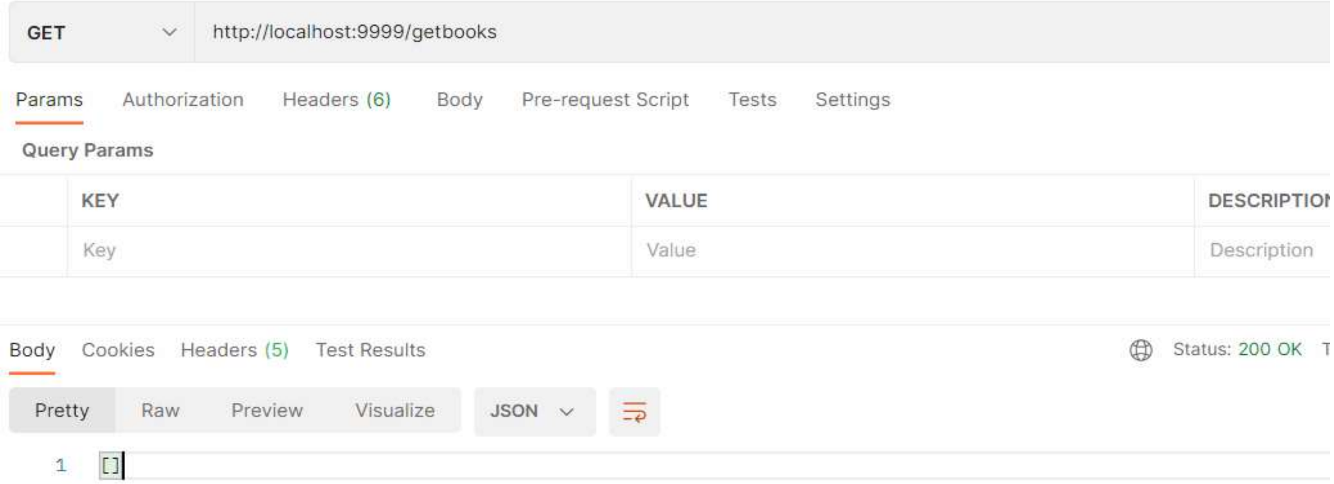
getbooks request getting records from collection

addbook request adding book object into same collection





```
@RestController
public class BookController {
    2 usages
    List<Book> booksMasterData = new ArrayList<Book>();
    /* GET
    http://localhost:9999/getbooks
     */
    @GetMapping("/getbooks")
    public ResponseEntity<?> getBooks(){
        return new ResponseEntity<>(booksMasterData,HttpStatus.OK);
    }
    /*
       POST
    http://localhost:9999/addBook
     */
    @PostMapping("/addBook")
    public ResponseEntity<?> addBook(@RequestBody Book book){
        booksMasterData.add(book);
        return new ResponseEntity<>( body: "Data received",HttpStatus.OK);
    }
```



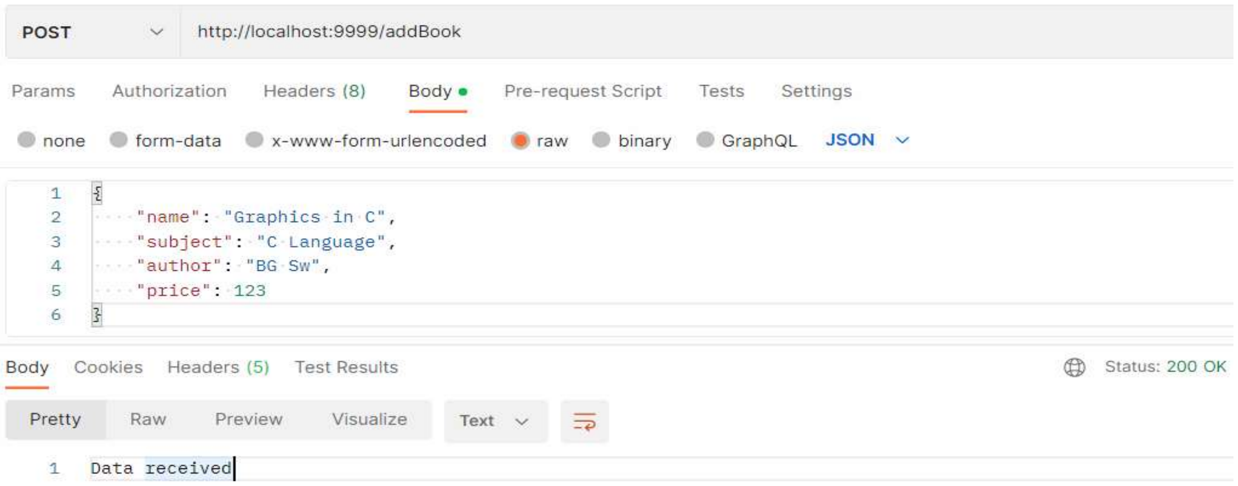
GET http://localhost:9999/getbooks

Query Params

KEY	VALUE	DESCRIPTIOI
Key	Value	Description

Status: 200 OK

```
1 []
```

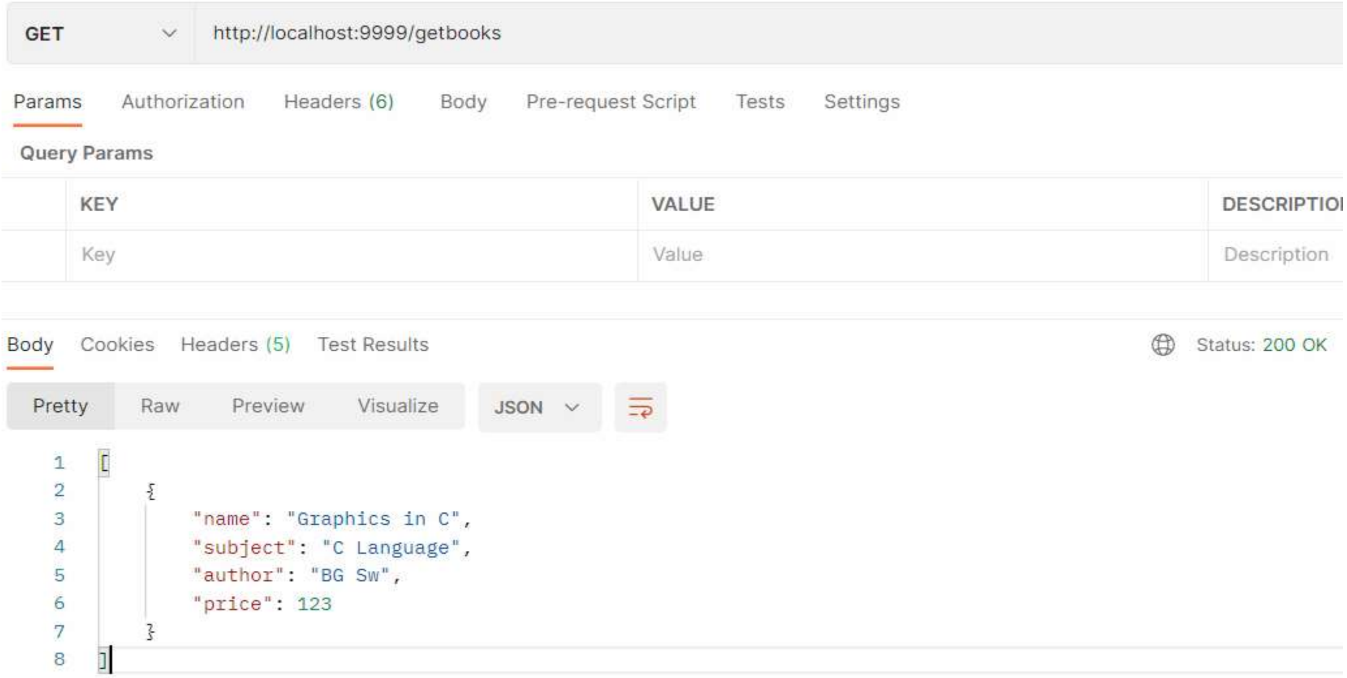


http://localhost:9999/addBook

```
{
    "name": "Graphics in C",
    "subject": "C Language",
    "author": "BG Sw",
    "price": 123
}
```

Status: 200 OK

```
Data received
```



http://localhost:9999/getbooks

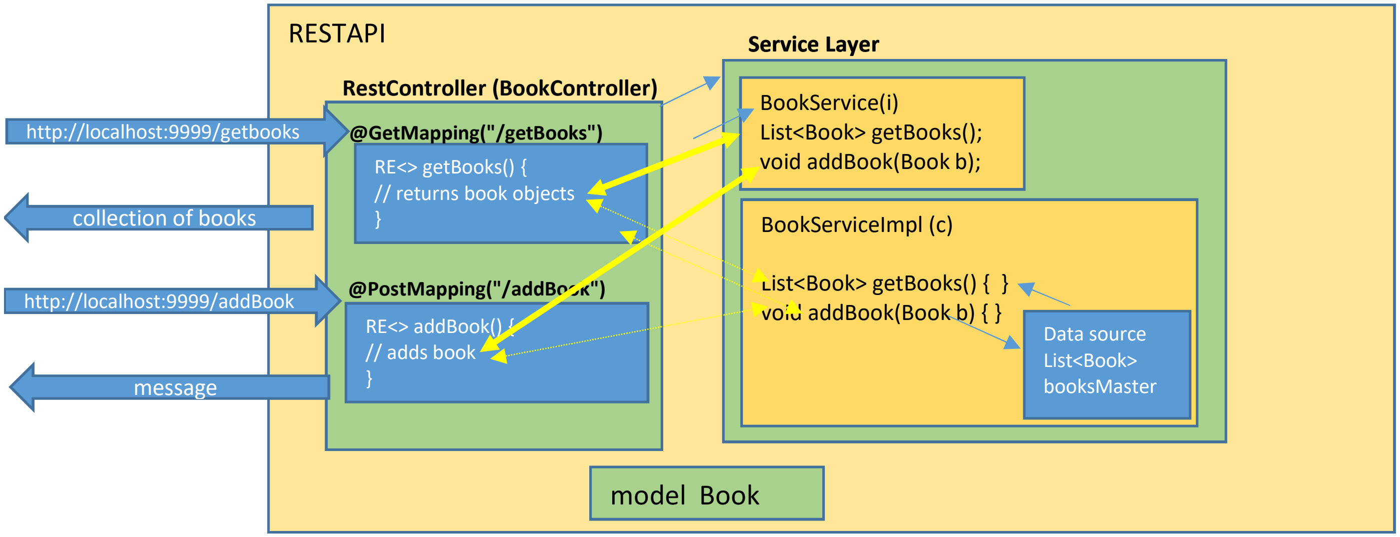
Query Params

KEY	VALUE	DESCRIPTIOI
Key	Value	Description

Status: 200 OK

```
[
    {
        "name": "Graphics in C",
        "subject": "C Language",
        "author": "BG Sw",
        "price": 123
    }
]
```

Controller has service



RESTAPI

http://localhost:9999/getbooks

collection of books

http://localhost:9999/addBook

message

RestController (BookController)

@GetMapping("/getBooks")

RE<> getBooks() {
// returns book objects
}

@PostMapping("/addBook")

RE<> addBook() {
// adds book
}

Service Layer

BookService(i)
List<Book> getBooks();
void addBook(Book b);

BookServiceImpl (c)

List<Book> getBooks() { }
void addBook(Book b) { }

Data source
List<Book>
booksMaster

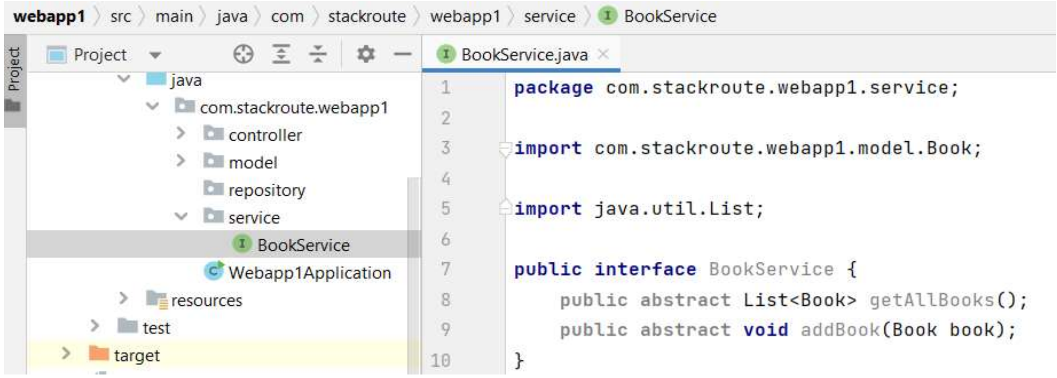
model Book

Controller
dedicated for handling request
should not perform any data operations

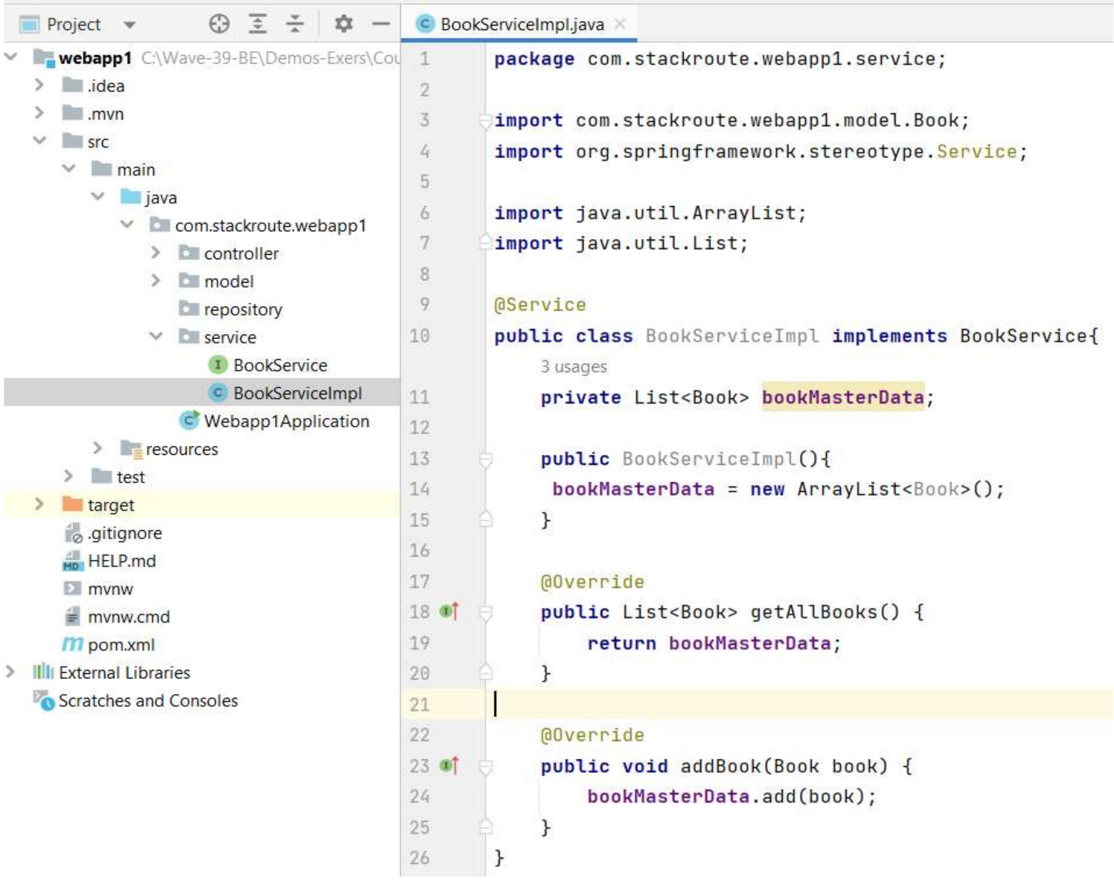
one method to return list of book object
one method to push received book object into list

List<Book> getBooks ( );
void addBook(Book book);

Step 1 create service layer



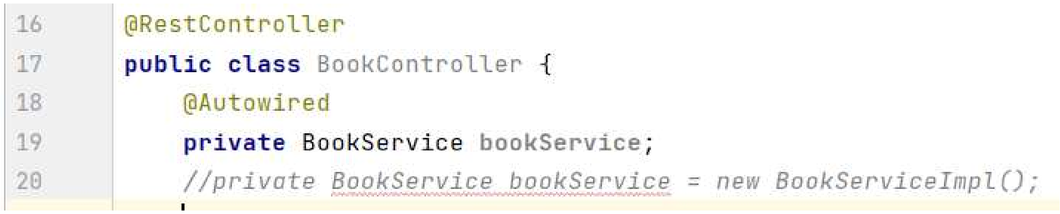
```
package com.stackroute.webapp1.service;

import com.stackroute.webapp1.model.Book;

import java.util.List;

public interface BookService {
    public abstract List<Book> getAllBooks();
    public abstract void addBook(Book book);
}
```



```java
package com.stackroute.webapp1.service;

import com.stackroute.webapp1.model.Book;
import org.springframework.stereotype.Service;

import java.util.ArrayList;
import java.util.List;

@Service
public class BookServiceImpl implements BookService{
    private List<Book> bookMasterData;

    public BookServiceImpl(){
     bookMasterData = new ArrayList<Book>();
    }

    @Override
    public List<Book> getAllBooks() {
        return bookMasterData;
    }

    @Override
    public void addBook(Book book) {
        bookMasterData.add(book);
    }
}
```

Step 2 Modify controller Note : remove created collection in previous demo

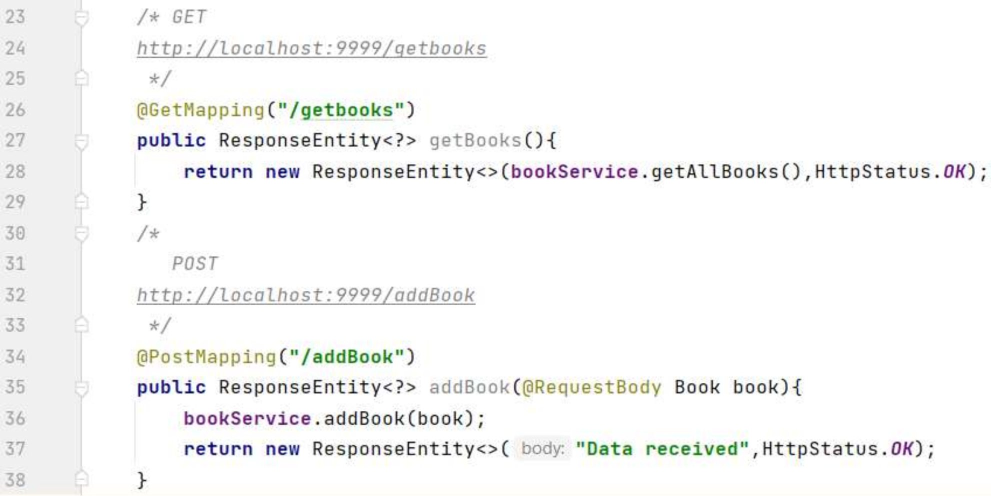
inject service into controller



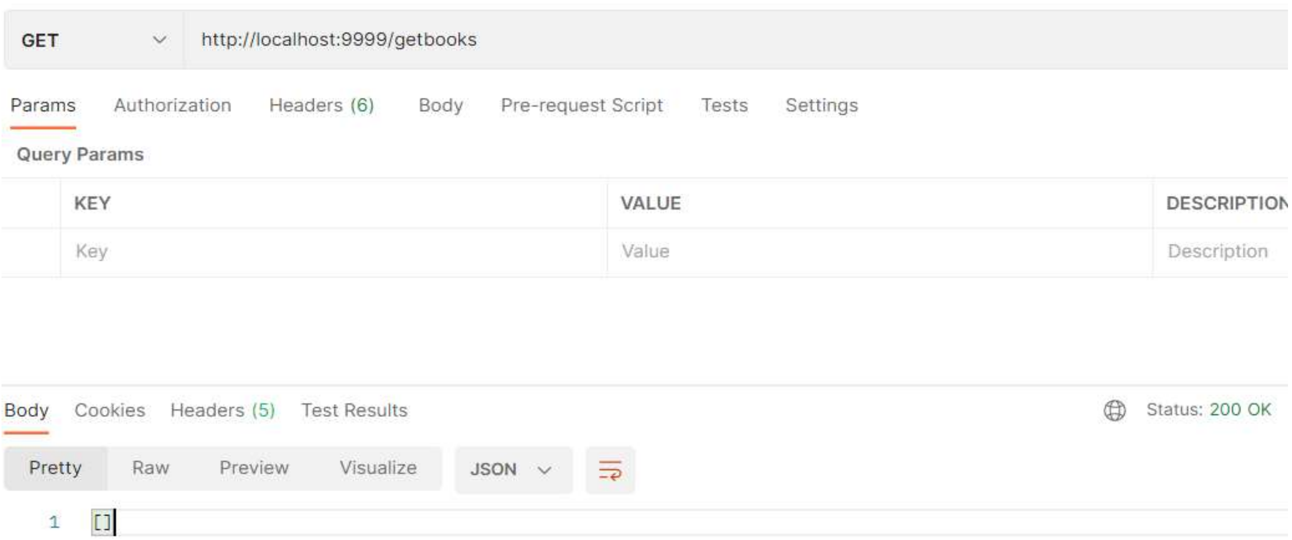
```java
@RestController
public class BookController {
    @Autowired
    private BookService bookService;
    //private BookService bookService = new BookServiceImpl();
```

Step 3 edit request handler methods

call service methods



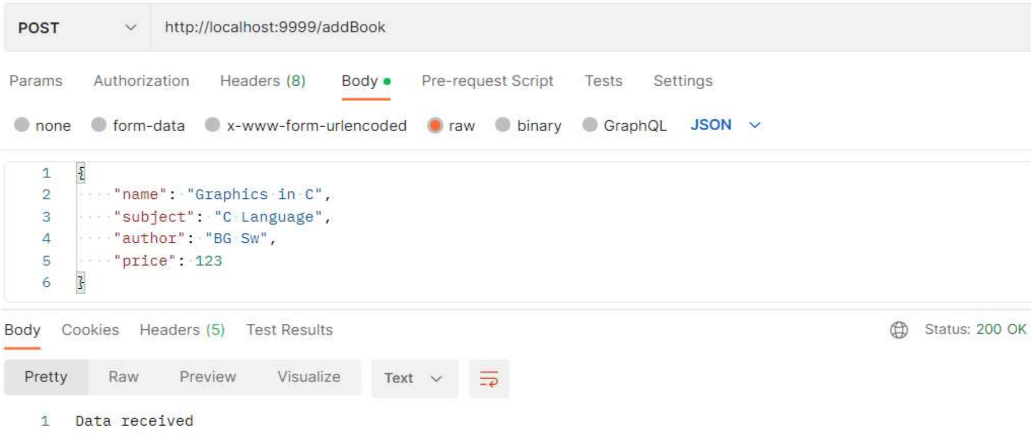
```
/* GET
http://localhost:9999/getbooks
 */
@GetMapping("/getbooks")
public ResponseEntity<?> getBooks(){
    return new ResponseEntity<>(bookService.getAllBooks(),HttpStatus.OK);
}
/*
   POST
http://localhost:9999/addBook
 */
@PostMapping("/addBook")
public ResponseEntity<?> addBook(@RequestBody Book book){
    bookService.addBook(book);
    return new ResponseEntity<>( body: "Data received",HttpStatus.OK);
}
```



GET http://localhost:9999/getbooks

Status: 200 OK

```
[]
```

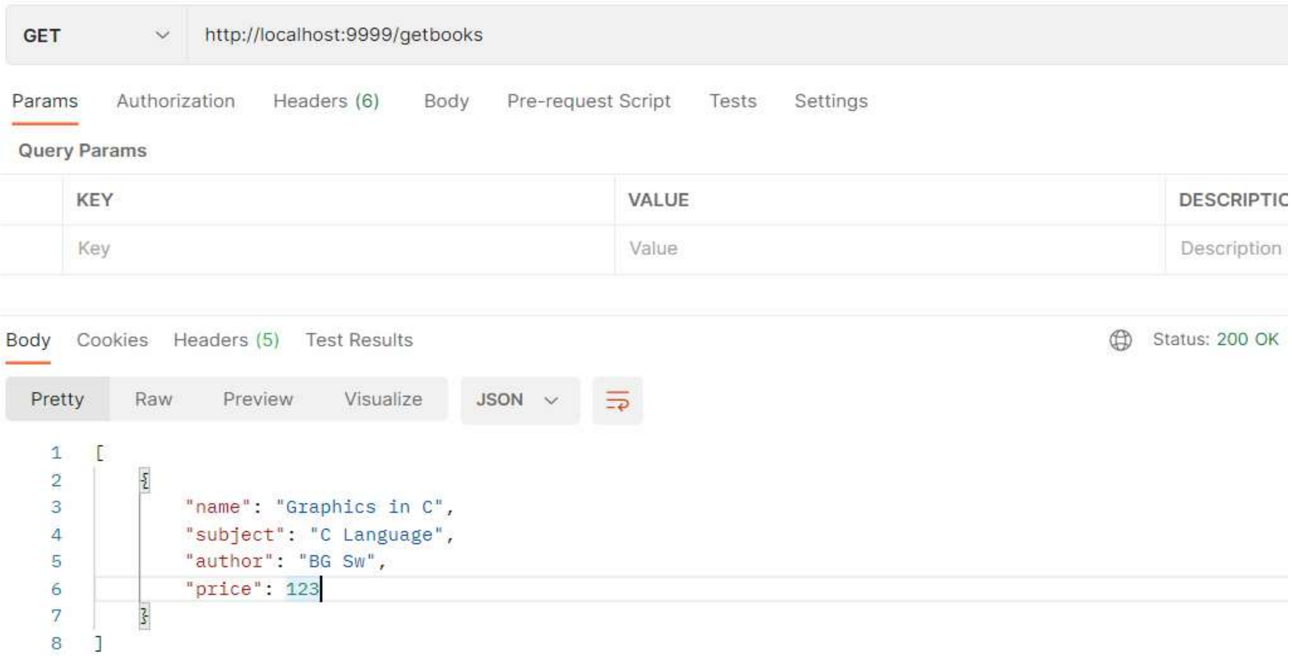


POST http://localhost:9999/addBook

```
{
    "name": "Graphics in C",
    "subject": "C Language",
    "author": "BG Sw",
    "price": 123
}
```

Status: 200 OK

```
Data received
```



GET http://localhost:9999/getbooks

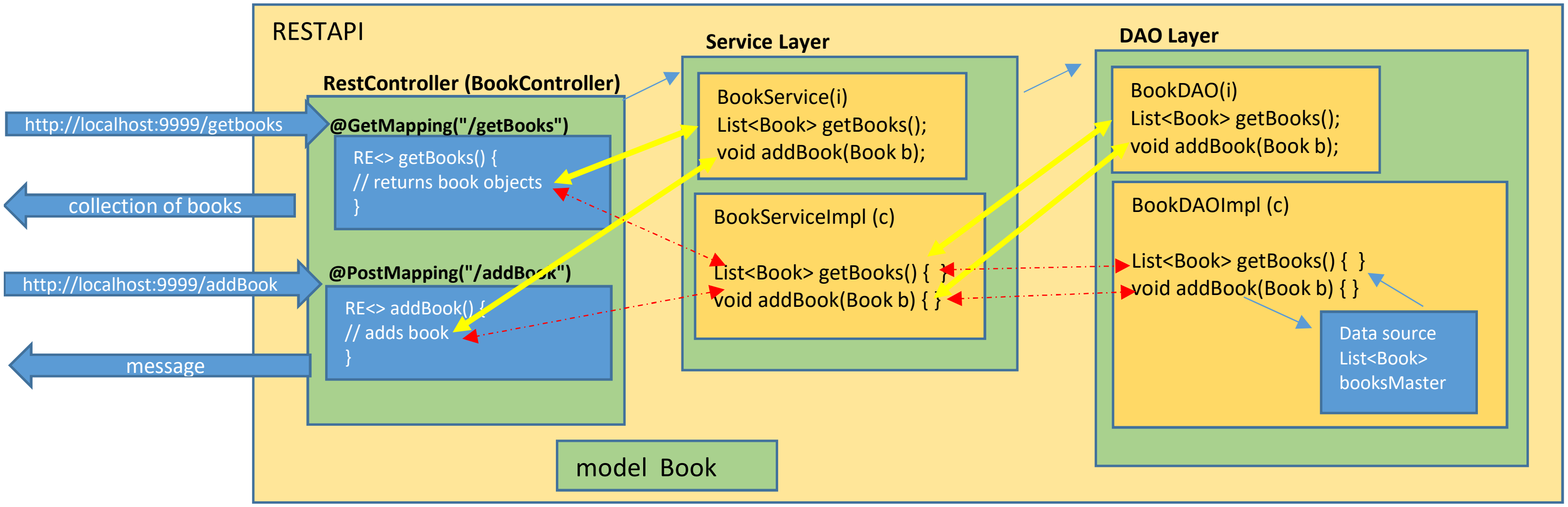
KEY	VALUE	DESCRIPTIC
Key	Value	Description

Status: 200 OK

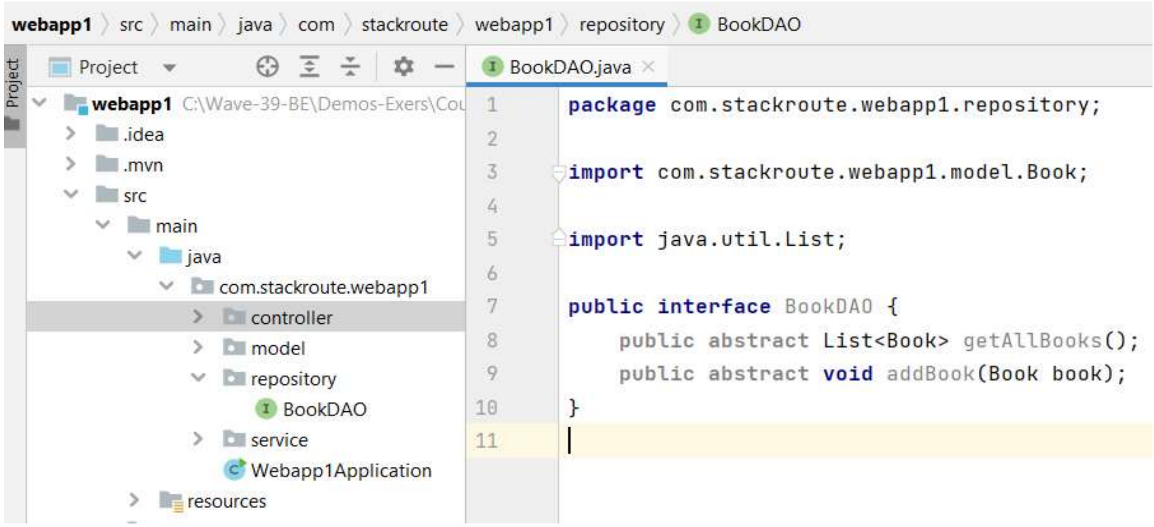
```
[
    {
        "name": "Graphics in C",
        "subject": "C Language",
        "author": "BG Sw",
        "price": 123
    }
]
```

Controller HAS Service

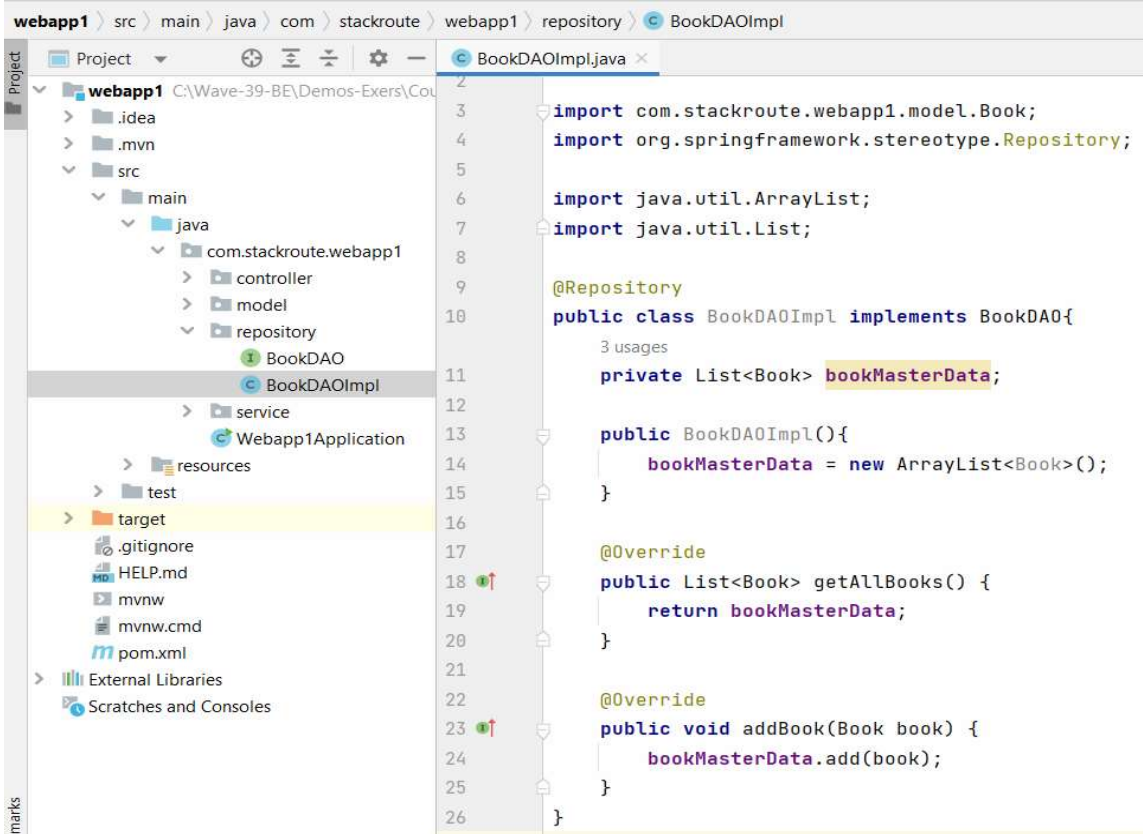
Service HAS DAO



Step 1 Define DAO layer



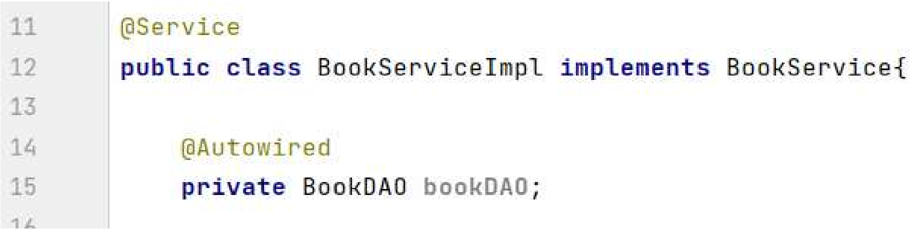
```java
package com.stackroute.webapp1.repository;

import com.stackroute.webapp1.model.Book;

import java.util.List;

public interface BookDAO {
    public abstract List<Book> getAllBooks();
    public abstract void addBook(Book book);
}
```



```java
import com.stackroute.webapp1.model.Book;
import org.springframework.stereotype.Repository;

import java.util.ArrayList;
import java.util.List;

@Repository
public class BookDAOImpl implements BookDAO{
    private List<Book> bookMasterData;

    public BookDAOImpl(){
        bookMasterData = new ArrayList<Book>();
    }

    @Override
    public List<Book> getAllBooks() {
        return bookMasterData;
    }

    @Override
    public void addBook(Book book) {
        bookMasterData.add(book);
    }
}
```

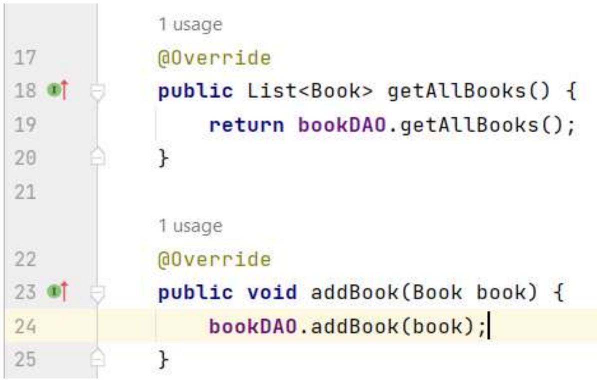
Step 2 Edit serivce impl class
inject dao dependency

Note: remove collection created in prev demo

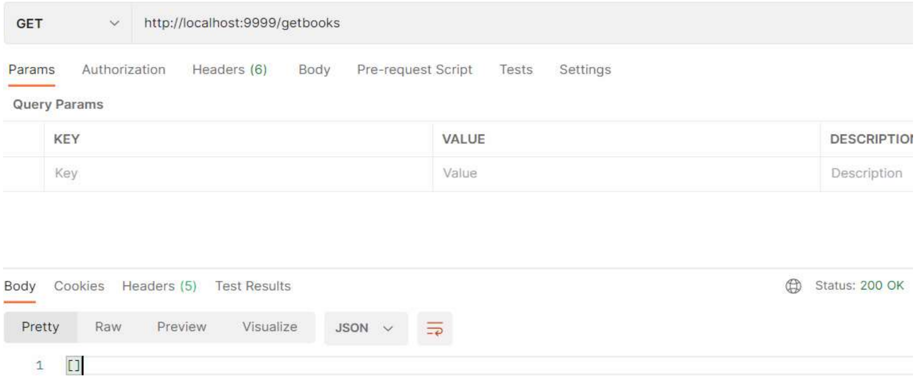


```java
@Service
public class BookServiceImpl implements BookService{

    @Autowired
    private BookDAO bookDAO;
```

Step 3 edit methods in service impl
call dao methods



```java
    1 usage
    @Override
    public List<Book> getAllBooks() {
        return bookDAO.getAllBooks();
    }

    1 usage
    @Override
    public void addBook(Book book) {
        bookDAO.addBook(book);
    }
```

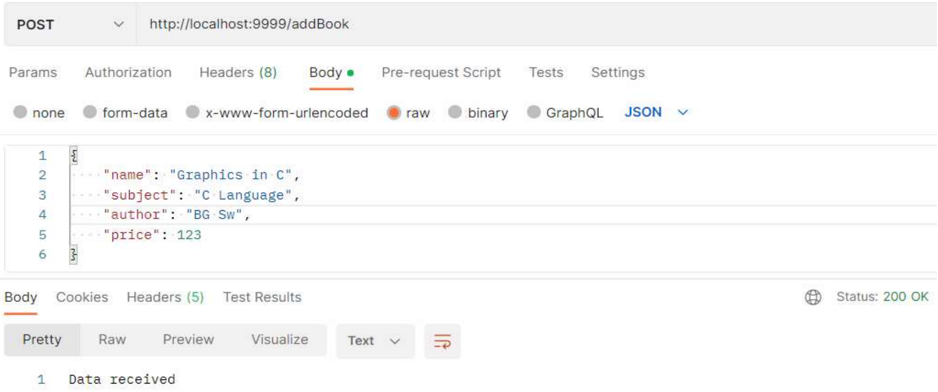


GET http://localhost:9999/getbooks

KEY	VALUE	DESCRIPTIOI
Key	Value	Description

Status: 200 OK

```
[]
```



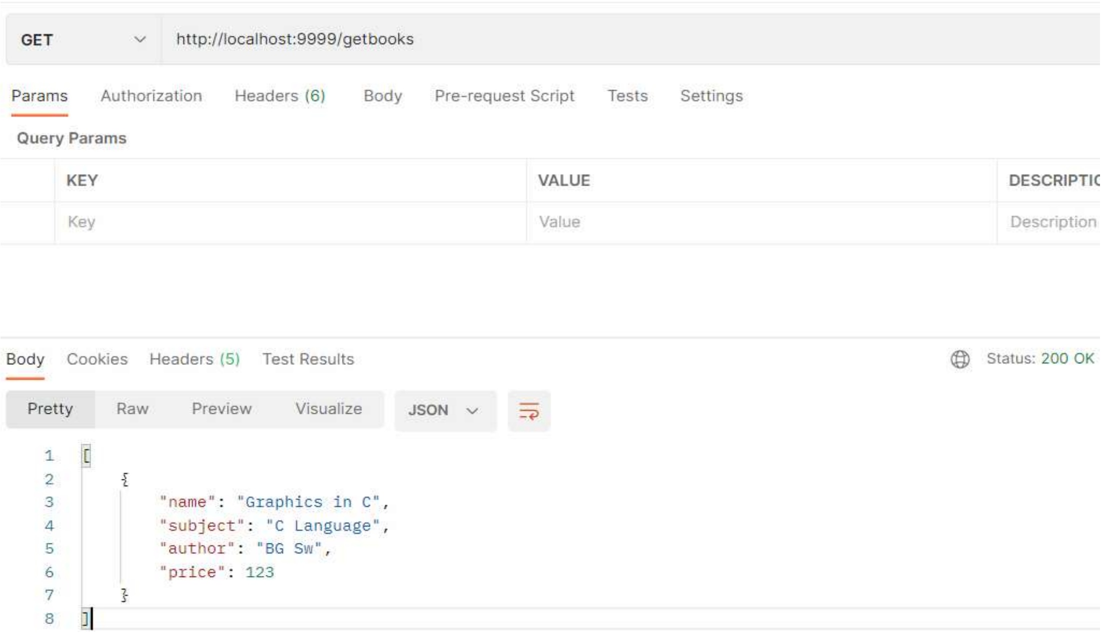
POST http://localhost:9999/addBook

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```json
{
    "name": "Graphics in C",
    "subject": "C Language",
    "author": "BG Sw",
    "price": 123
}
```

Status: 200 OK

```
Data received
```



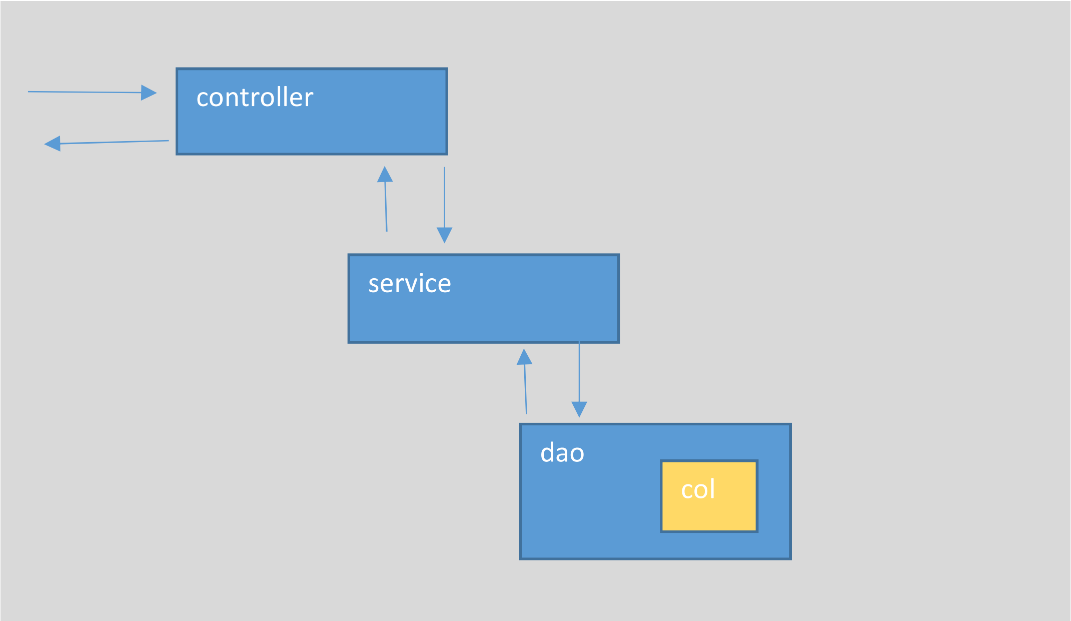
GET http://localhost:9999/getbooks

Query Params

KEY	VALUE	DESCRIPTIC
Key	Value	Description

Status: 200 OK

```
[
    {
        "name": "Graphics in C",
        "subject": "C Language",
        "author": "BG Sw",
        "price": 123
    }
]
```


controller
service
dao
col