

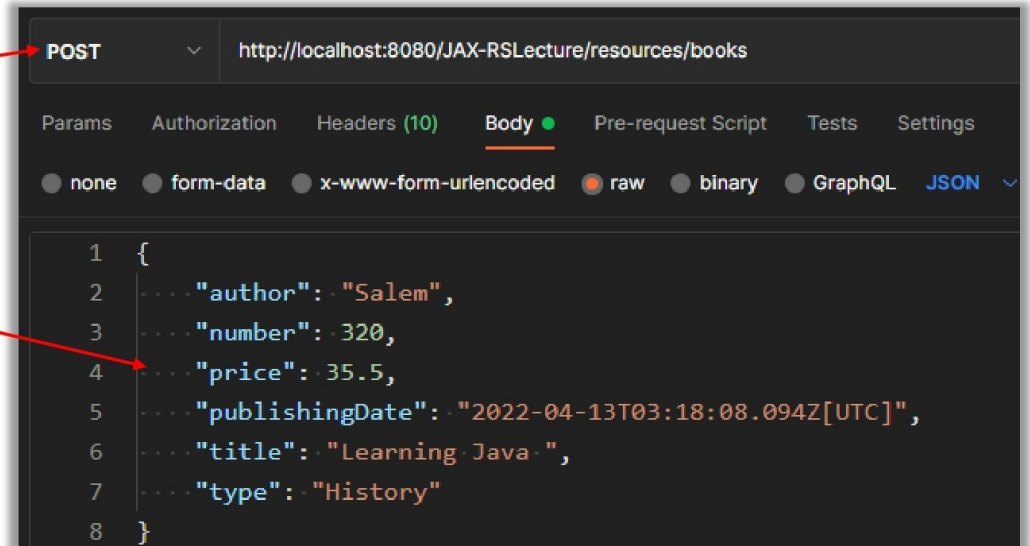


JAX-RS  Java EE

P-II

- تستخدم لتحديد أن هذه الخدمة سيتم طلبها عن طريق HTTP POST method .

```
@POST
@Consumes (MediaType.APPLICATION_JSON)
@Produces (MediaType.APPLICATION_JSON)
public List<Book> addBook (Book book) {
    books.add(book);
    return books;
}
```



POST http://localhost:8080/JAX-RSLecture/resources/books

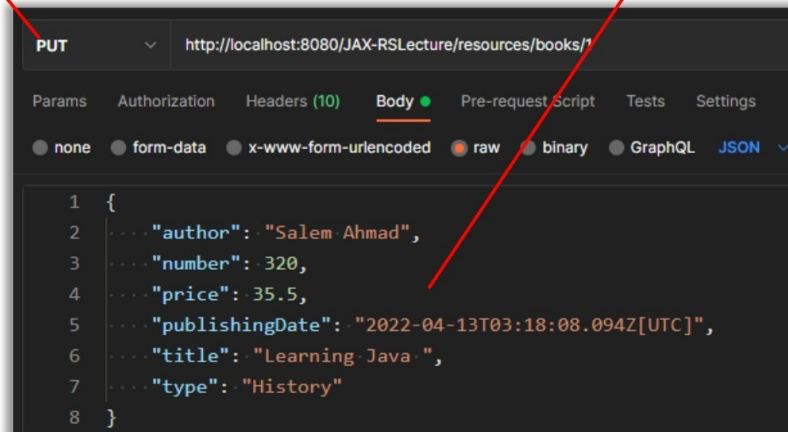
Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

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

```
1 {
2   ... "author": "Salem",
3   ... "number": 320,
4   ... "price": 35.5,
5   ... "publishingDate": "2022-04-13T03:18:08.094Z[UTC]",
6   ... "title": "Learning Java",
7   ... "type": "History"
8 }
```

- تستخدم لتحديد أن هذه الخدمة سيتم طلبها عن طريق HTTP PUT method .

```
@PUT
@Path("/{bookId}")
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public Collection<Book> updateBook(@PathParam("bookId") int bookId, Book book) {
    bookMap = new HashMap<>();
    bookMap.put(1, new Book("Learning Java ", "Salem", 35.5f, new Date(), 320, "History"));
    bookMap.put(2, new Book("Learning C ", "Ali", 35.5f, new Date(), 450, "Comics"));
    bookMap.put(3, new Book("Learning Python ", "Khalid", 35.5f, new Date(), 230, "Scifi"));
    bookMap.put(bookId, book);
    return bookMap.values();
}
```



PUT http://localhost:8080/JAX-RSLecture/resources/books/1

Params Authorization Headers (10) Body Pre-request Script Tests Settings

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

```
1 {
2   ... "author": "Salem Ahmad",
3   ... "number": 320,
4   ... "price": 35.5,
5   ... "publishingDate": "2022-04-13T03:18:08.094Z[UTC]",
6   ... "title": "Learning Java ",
7   ... "type": "History"
8 }
```

- تستخدم لتحديد أن هذه الخدمة سيتم طلبها عن طريق HTTP DELETE method .

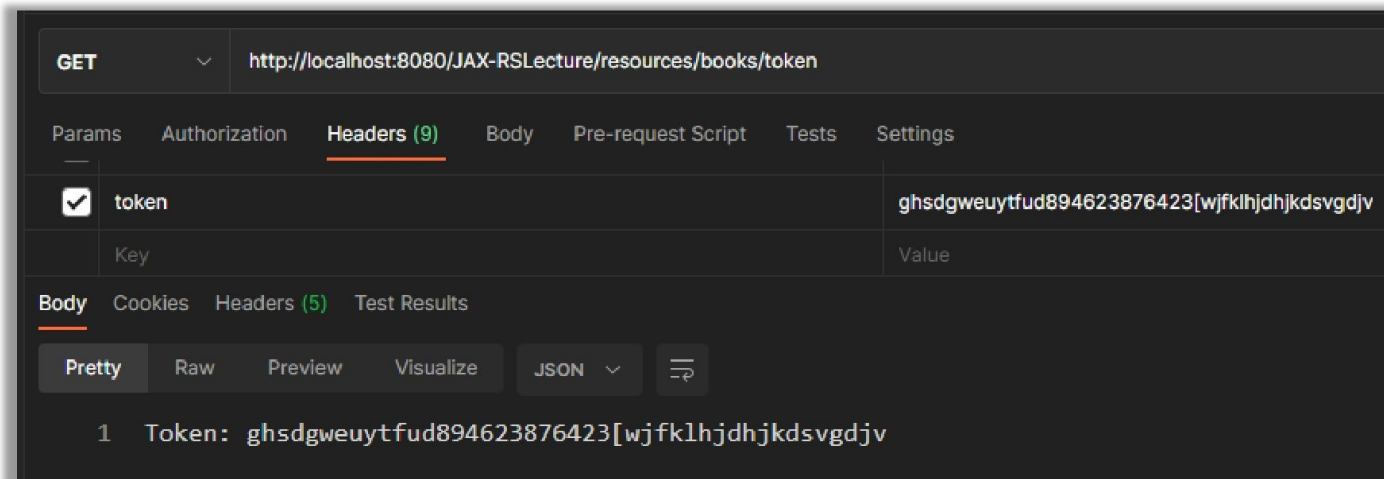
```
@DELETE
@Path("/{id}")
@Produces(MediaType.APPLICATION_JSON)
public Collection<Book> deleteBook(@PathParam("id") int bookId) {
    bookMap = new HashMap<>();
    bookMap.put(1, new Book("Learning Java ", "Salem", 35.5f, new Date(), 320, "History"));
    bookMap.put(2, new Book("Learning C ", "Ali", 35.5f, new Date(), 450, "Comics"));
    bookMap.put(3, new Book("Learning Python ", "Khalid", 35.5f, new Date(), 230, "Scifi"));
    bookMap.remove(bookId);
    return bookMap.values();
}
```

DELETE

http://localhost:8080/JAX-RSLecture/resources/books/1

- تستخدم للحصول على القيم المرسله في HTTP Header .

```
@GET
@Path("/annotation")
public String getHeaderValue (@HeaderParam("token") String token) {
    return "Token: " + token;
}
```



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8080/JAX-RSLecture/resources/books/token
- Headers (9):** token (checked) with value ghsdgweuytfud894623876423[wjfk1hjdhjkdsvgdjv]
- Body:** Pretty view showing 1 Token: ghsdgweuytfud894623876423[wjfk1hjdhjkdsvgdjv]

- يمكن من خلالها معرفة مجموعة من المعلومات مثل:

- AbsolutePath

- Query Parameters

- Path Parameters

@Context with UriInfo



```
@GET
@Path("/context")
public String getContext(@Context UriInfo uriInfo) {
    return "Path: " + uriInfo.getAbsolutePath();
}
```

The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: `http://localhost:8080/JAX-RSLecture/resources/books/context`
- Body tab selected, showing the response: `Path: http://localhost:8080/JAX-RSLecture/resources/books/context`

- يمكن من خلالها معرفة مجموعة من المعلومات مثل:

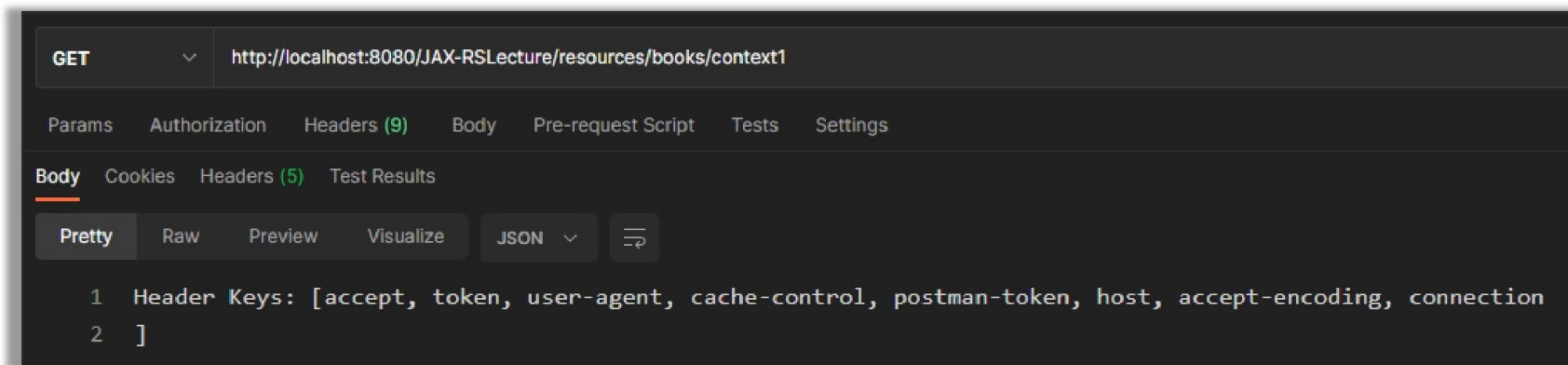
- Media Type

- Cookies

- Request Headers

@Context with HttpHeaders

```
@GET
@Path("/context1")
public String getContextHttpHeader(@Context HttpHeaders headers) {
    return "Header Keys: " + headers.getRequestHeaders().keySet();
}
```



GET

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 Header Keys: [accept, token, user-agent, cache-control, postman-token, host, accept-encoding, connection
2 ]
```

@BeanParam

- يستخدم للوصول لمجموعة من query and path parameters من خلال class واحدة بدلاً من التعامل معهم بشكل منفرد.

```
@GET
@Path("/{bookId}")
@Produces(MediaType.APPLICATION_JSON)
public String getBookParametersBean(
    @PathParam("bookId") int bookId,
    @QueryParam("id") int id,
    @QueryParam("start") int start,
    @QueryParam("end") int end) {
```

```
@GET
@Path("/{bookId}")
@Produces(MediaType.APPLICATION_JSON)
public String getBookParametersBean(@BeanParam BookParametersBean bookParametersBean) {
    return bookParametersBean.toString();
}
```

```
public class BookParametersBean {
    @QueryParam("id") String id;
    @QueryParam("start") String start;
    @QueryParam("end") String end;
    @PathParam("bookId") String bookId;

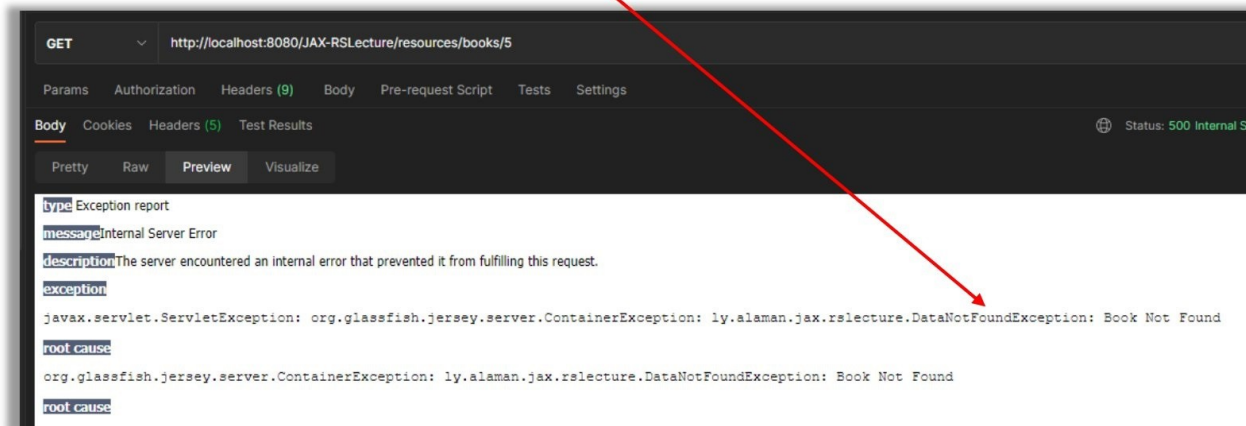
    //setters and getters
}
```

- تستخدم لاعداد response من خلال تحديد status code و نوع الرد وغيرها.

```
@GET
@Produces(MediaType.APPLICATION_JSON)
public Response getBooks() {
    books = new ArrayList<>();
    books.add(new Book("Learning Java ", "Salem", 35.5f, new Date(), 320, "History"));
    books.add(new Book("Learning C ", "Ali", 35.5f, new Date(), 450, "Comics"));
    books.add(new Book("Learning Python ", "Khalid", 35.5f, new Date(), 230, "Scifi"));
    return Response.status(200).entity(books).build();
}
```

Exception Handling

```
@GET
@Path("/{bookId}")
@Produces(MediaType.APPLICATION_JSON)
public Book getBookk(@PathParam("bookId") int bookId) throws Exception {
    bookMap = new HashMap<>();
    bookMap.put(1, new Book("Learning Java ", "Salem", 35.5f, new Date(), 320, "History"));
    bookMap.put(2, new Book("Learning C ", "Ali", 35.5f, new Date(), 450, "Comics"));
    bookMap.put(3, new Book("Learning Python ", "Khalid", 35.5f, new Date(), 230, "Scifi"));
    Book b = bookMap.get(bookId);
    if (b == null) {
        throw new DataNotFoundException("Book Not Found");
    }
    return bookMap.get(bookId);
}
```



GET http://localhost:8080/JAX-RSLecture/resources/books/5

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body Cookies Headers (5) Test Results Status: 500 Internal S

Pretty Raw Preview Visualize

type Exception report

message Internal Server Error

description The server encountered an internal error that prevented it from fulfilling this request.

exception

```
javax.servlet.ServletException: org.glassfish.jersey.server.ContainerException: ly.alaman.jax.rslecture.DataNotFoundException: Book Not Found
```

root cause

```
org.glassfish.jersey.server.ContainerException: ly.alaman.jax.rslecture.DataNotFoundException: Book Not Found
```

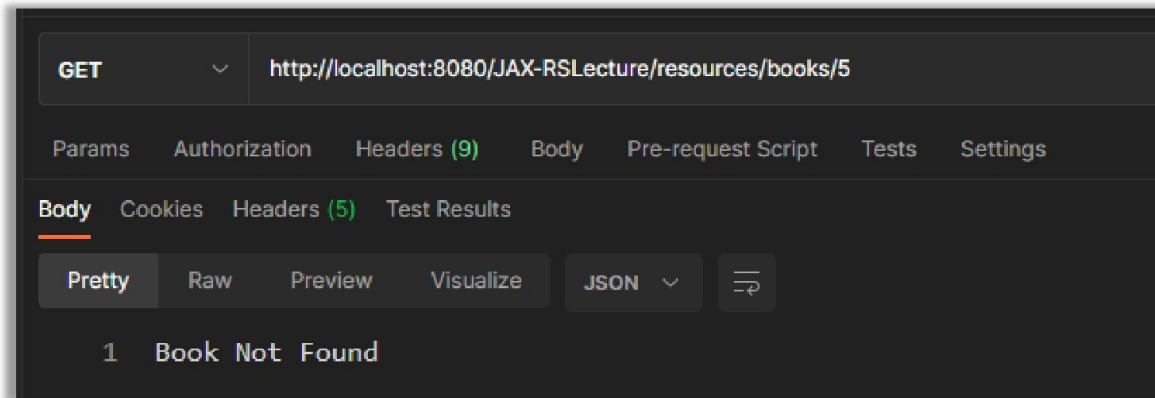
root cause

- للتعامل مع Exceptions في JAX-RS واطهارها بصيغة يمكن التعامل معها يتم استخدام `ExceptionHandler`

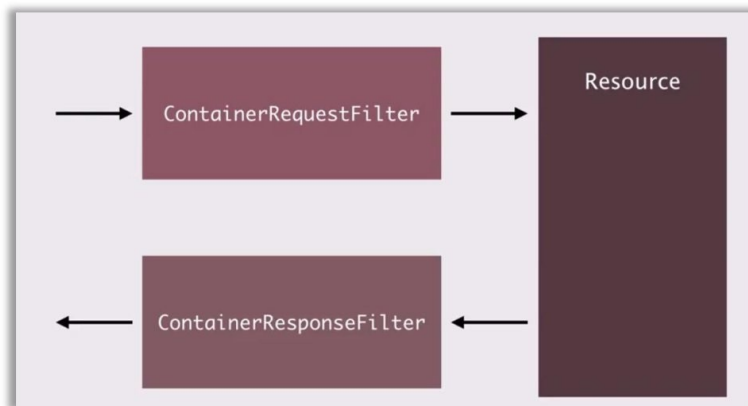
```
public class DataNotFoundException extends Exception{  
  
    public DataNotFoundException(String message) {  
        super(message);  
    }  
  
}
```

```
@Provider  
public class DataNotFoundExceptionMapper implements ExceptionMapper<DataNotFoundException> {  
  
    @Override  
    public Response toResponse(DataNotFoundException ex) {  
  
        return Response.status(Status.NOT_FOUND).entity(ex.getMessage()).build();  
    }  
  
}
```

Exception Handling



- تستخدم JAX-RS نوعان من Filters :
- Request Filter ويتم تنفيذه قبل استدعاء JAX-RS method .
- Response Filter ويتم تنفيذه بعد استدعاء JAX-RS method .



Example:

- المثال التالي يبين كيفية إضافة بعض البيانات على كل response .

```
@Provider
public class PoweredByFilter implements ContainerResponseFilter {

    @Override
    public void filter(ContainerRequestContext requestContext, ContainerResponseContext responseContext)
        responseContext.getHeaders().add("Powered By: ", "Java EE Course");
    }
}
```

GET http://localhost:8080/JAX-RSLecture/resources/books

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body Cookies Headers (6) Test Results

KEY	VALUE
Server	Payara Server 5.2021.4 #badassfish
X-Powered-By	Servlet/4.0 JSP/2.3 (Payara Server 5.2021.4 #badassfish Ja
Powered By	: Java EE Course

GET http://localhost:8080/JAX-RSLecture/resources/books/1

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body Cookies Headers (6) Test Results

KEY	VALUE
Server	Payara Server 5.2021.4 #badassfish
X-Powered-By	Servlet/4.0 JSP/2.3 (Payara Server 5.2021.4 #badassfish Ja
Powered By	: Java EE Course

Example:

- المثال التالي يبين كيفية استخدام filter في عمل logging للبيانات الموجودة في كل من Request header و Response header .

```
@Provider
public class LoginFilter implements ContainerRequestFilter, ContainerResponseFilter {

    @Override
    public void filter(ContainerRequestContext requestContext) throws IOException {
        System.out.println("-- response info --");
        System.out.println("Path: " + requestContext.getUriInfo().getPath());
        requestContext.getHeaders().entrySet().forEach(h -> System.out.println(h.getKey() + ": " + h.getValue()));
    }

    @Override
    public void filter(ContainerRequestContext requestContext, ContainerResponseContext responseContext) throws IOException {
        System.out.println("-- request info --");
        System.out.println("Path: " + requestContext.getUriInfo().getPath());
        responseContext.getHeaders().entrySet().forEach(h -> System.out.println(h.getKey() + ": " + h.getValue()));
    }
}
```

```
Payara Server x Run (JAX-RSLecture-1.0-SNAPSHOT) x
-- response info --|#]
Path: books/1|#]
accept: [application/json]|#]
token: [ghsdgweuytfud894623876423[wj#klhjdjhkdsvgd|#]
user-agent: [PostmanRuntime/7.29.0]|#]
cache-control: [no-cache]|#]
postman-token: [1b2fe6b5-1992-44a9-8eca-bec61e177ef9]|#]
host: [localhost:8080]|#]
accept-encoding: [gzip, deflate, br]|#]
connection: [keep-alive]|#]
-- request info --|#]
Path: books/1|#]
Content-Type: [application/json]|#]
Powered By: : [Java EE Course]|#]
```