JAX-RS Java EE

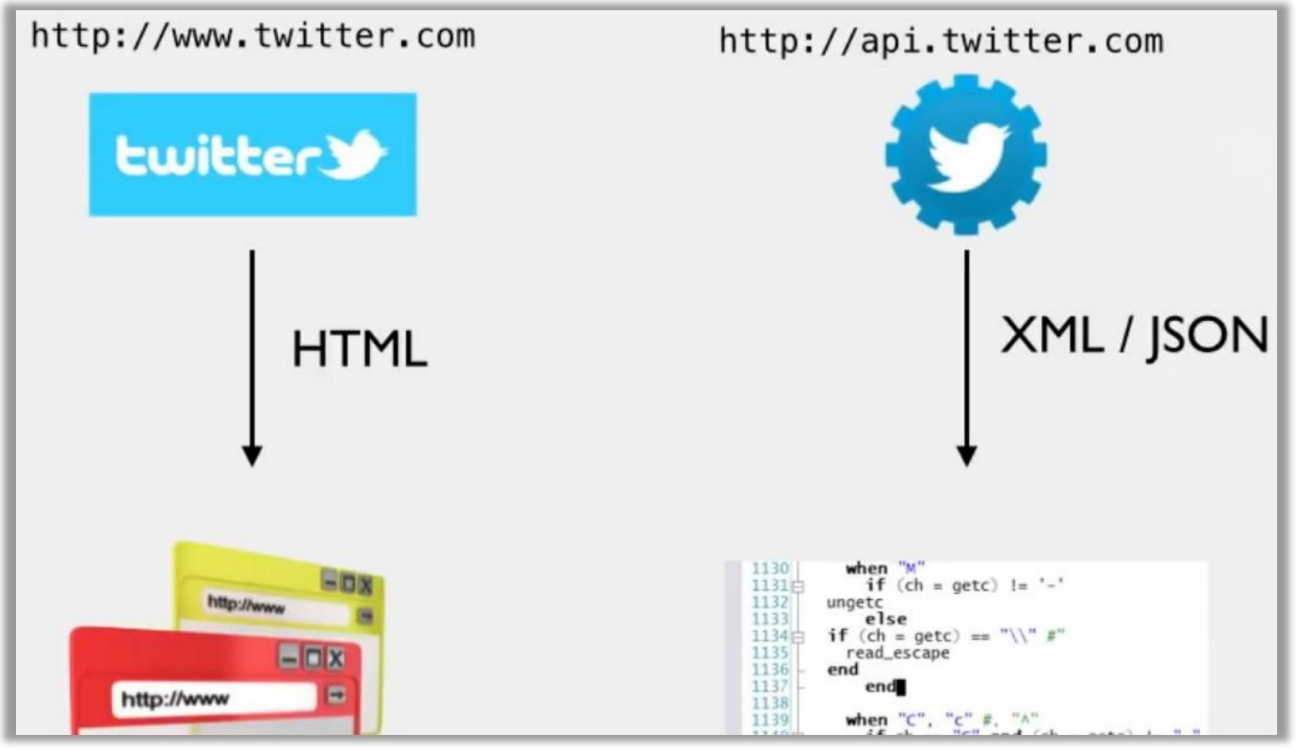
P-I

What is Web Services

- هي مجموعة الخدمات التي تكون متاحة على الانترنت ليتم استخدامها برمجياً في تطوير التطبيقات.
- يمكن اعتبارها Online API يمكنك استدعائها من خلال برنامجك.

What is Web Services

- يتم توفيرها عن طريق URL يتم تبادل المعلومات بصيف XML/JSON.



Web Services Types:

- RESTFul Web Services
- SOAP Web Services

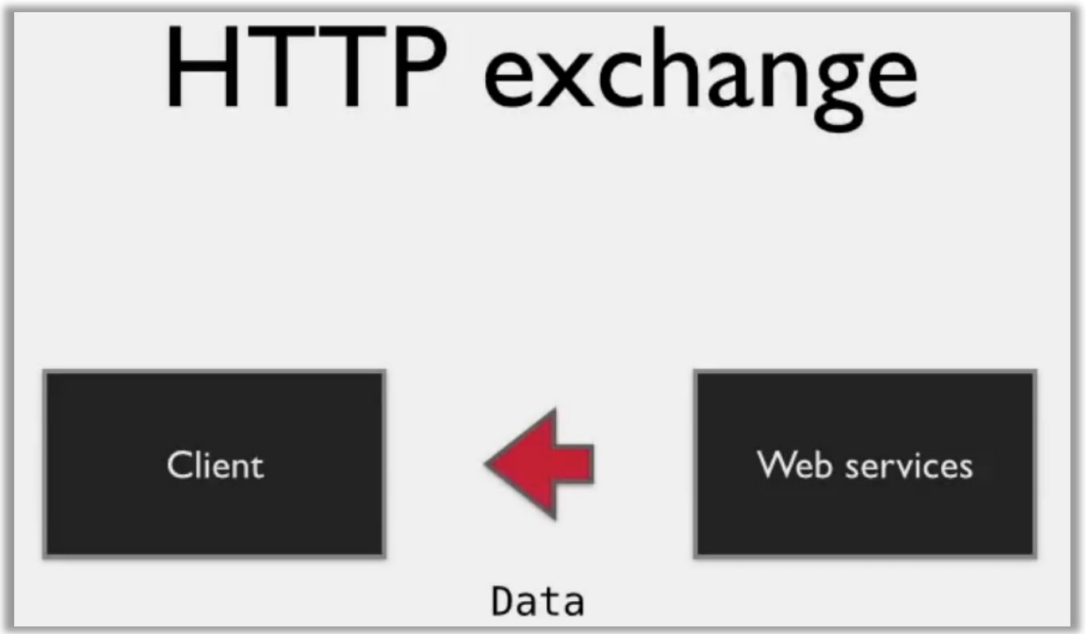
RESTFul Web Services

- هي Web Service تتبع REST Architecture .
- تسمح بتقديم API بصورة موحدة وآمنة.
- مستخدم هذه API يمكن يجري مجموعة من العمليات باستخدام HTTP Methods .

HTTP Exchange

- يتم تبادل المعلومات بين client و service بأستخدام HTTP protocol ويتم من خلاله تبادل البيانات بصيغ مختلفة مثل:

- XML
- JSON
- TEXT



HTTP Methods

- GET
- POST
- PUT
- DELETE

HTTP Status Codes

- تستخدم لتبين نجاح HTTP Request من عدمه.

Code	Description	Code	Description
200	OK	400	Bad Request
201	Created	401	Unauthorized
202	Accepted	403	Forbidden
301	Moved Permanently	404	Not Found
303	See Other	410	Gone

Content Types

- تستخدم لتبين نوع البيانات المرسلة والمستقبل إلى السيرفر.

```
Content-Type: text/html; charset=UTF-8
Content-Type: multipart/form-data; boundary=something
```

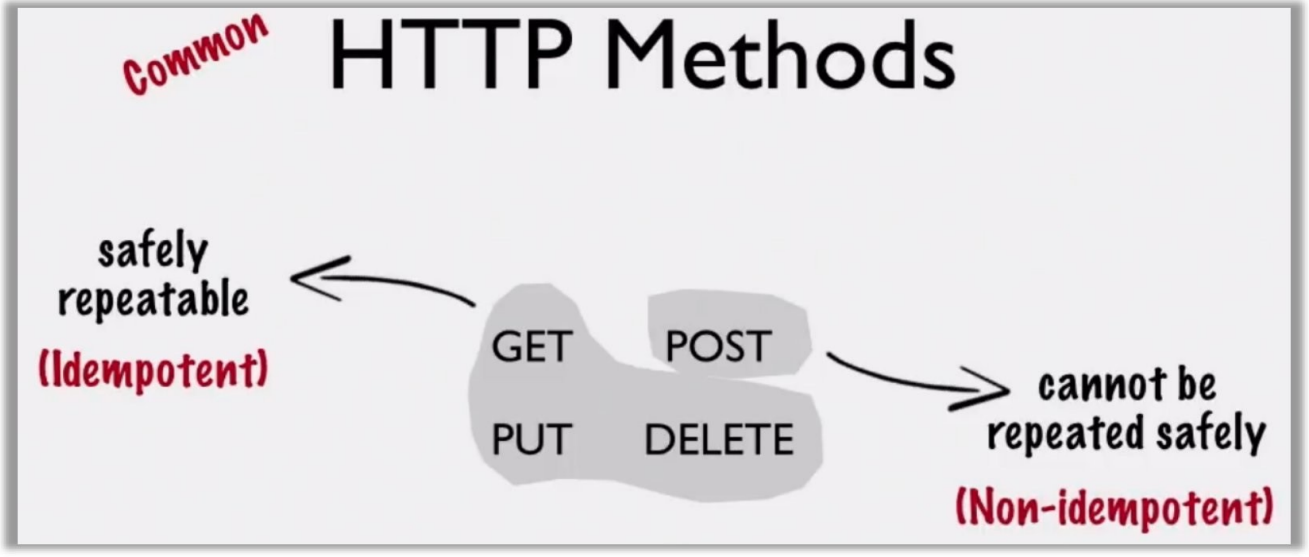
Resource Based URL

- تركز على resource الذي ستم العمل عليه.
- تتكون عادة من nouns.
- تعتمد على HTTP Methods لتحديد العمل الذي ستقدمه.

```
  Resource-based URL
------------------------------
  GET    /accounts/application
  POST   /accounts
  GET    /queries/form
  POST   /queries
  GET    /order/123
  POST   /order/123/items
  DELETE /order/123
```

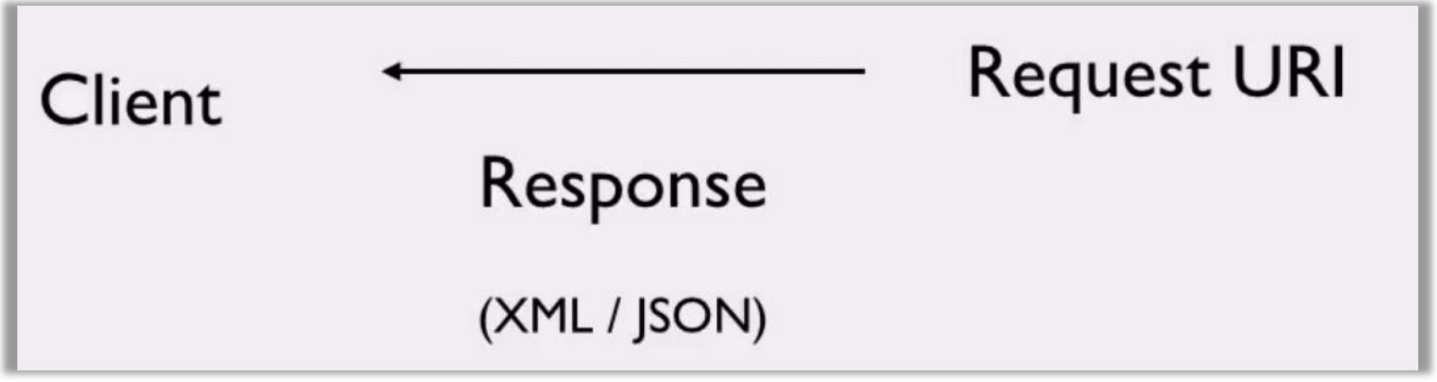
Method Idempotence

- هي إمكانية العملية عدة مرات دون ان تتغير النتيجة على ماكانت عليه اول مرة.



REST Response

- عند أستخدام REST الرد عادة يكون بيانات في أحد الصيغتين :
- JSON
- XML



JSON (JavaScript Object Notation)

- هي تنسيق بسيط لنقل البيانات يسهل قراءته وكتابته من قبل المطورين كما يسهل تحليله وتوليده من قبل البرامج المستخدمة له.

```
{
  "id":"10",
  "message":"Hello world",
  "created":"2014-06-01T18:06:36.902",
  "author":"koushik"
}
```

XML (Extensible Markup Language)

- هي عبارة عن markup language تستخدم لتمثيل البيانات لتساعد في نقلها وتخزينها.

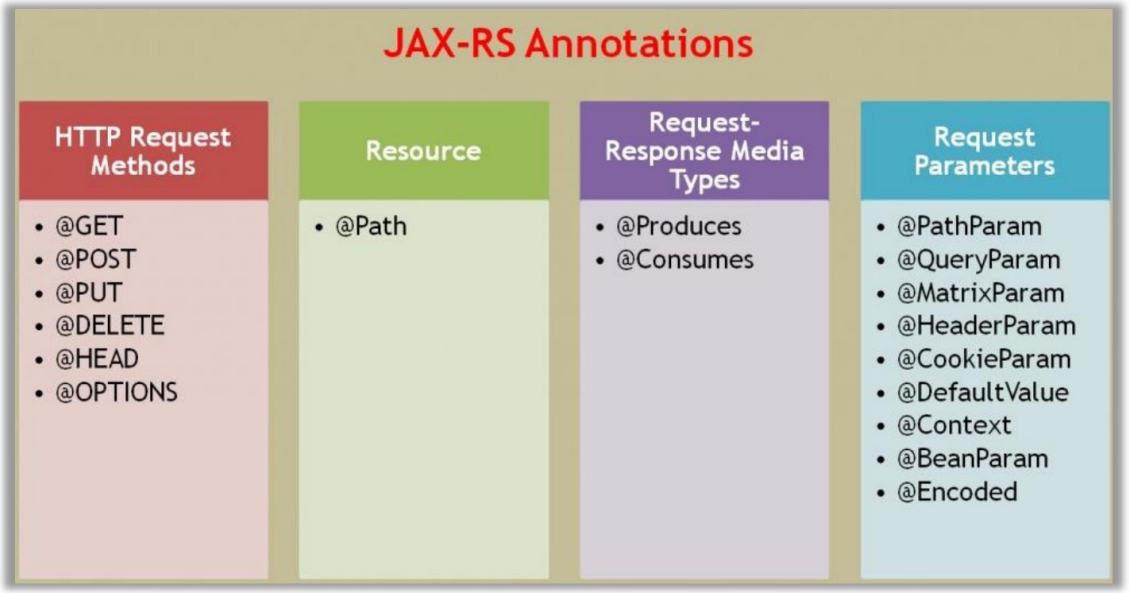
```
<messageEntity>
  <id>10</id>
  <message>Hello world</message>
  <created>2014-06-01T18:06:36.902</created>
  <author>koushik</author>
</messageEntity>
```

JAX-RS

- هي عبارة JAVA EE Specification تقدم مجموعة من interfaces و annotation لتطوير Web Services من النوع REST .

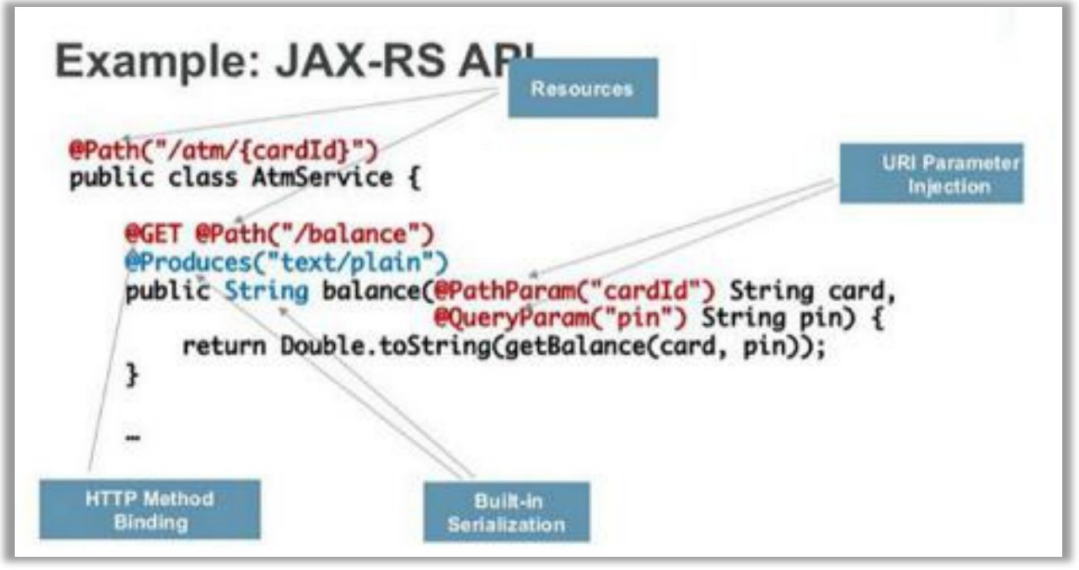
JAX-RS Annotations

- تعتمد JAX-RS على annotation في تطوير Web Services والجدول التالي يبين أهمها:



Example

- الشكل التالي يبين أستخدام JAX-RS annotations لتطوير Web Service .



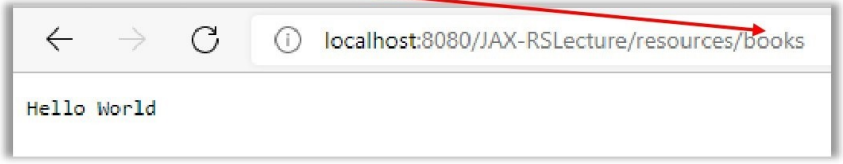
REST Resource

- هو عبارة عن Object له نوع وبيانات مرتبطة وعلاقات مع resources الأخرى ومجموعة من method التي تعمل عليه. ويمكن تجميعه في مجموعات متجانسة تعرف ب Collection .

@Path

- عند انشاء resource تستخدم @Path annotation لعمل mapping له مع HTTP Request URI.

```
@Path("books")
public class BooksResource {

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String getBooks() {

        return "Hello World";
    }

}
```



@GET

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

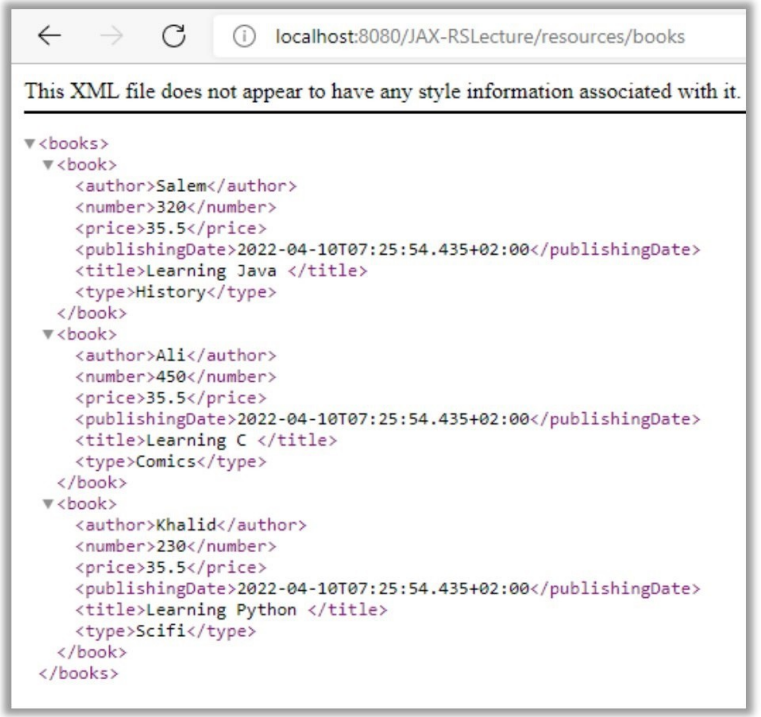
```
@Path("books")
public class BooksResource {

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String getBooks() {

        return "Hello World";
    }

}
```

@Produces

- تستخدم لتحديد نوع response التي ترجعه Web Service .
- للحصول على رد من النوع XML نستخدم @Produces(MediaType.APPLICATION_XML)
- للحصول على رد من النوع JSON نستخدم@Produces(MediaType.APPLICATION_JSON)
- للحصول على رد من النوع TEXT نستخدم @Produces(MediaType.TEXT_PLAIN)

XML Response

```java
@Path("books")
public class BooksResource {

    private List<Book> books;

    @GET
    @Produces(MediaType.APPLICATION_XML)
    public List<Book> 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 books;
    }

}
```



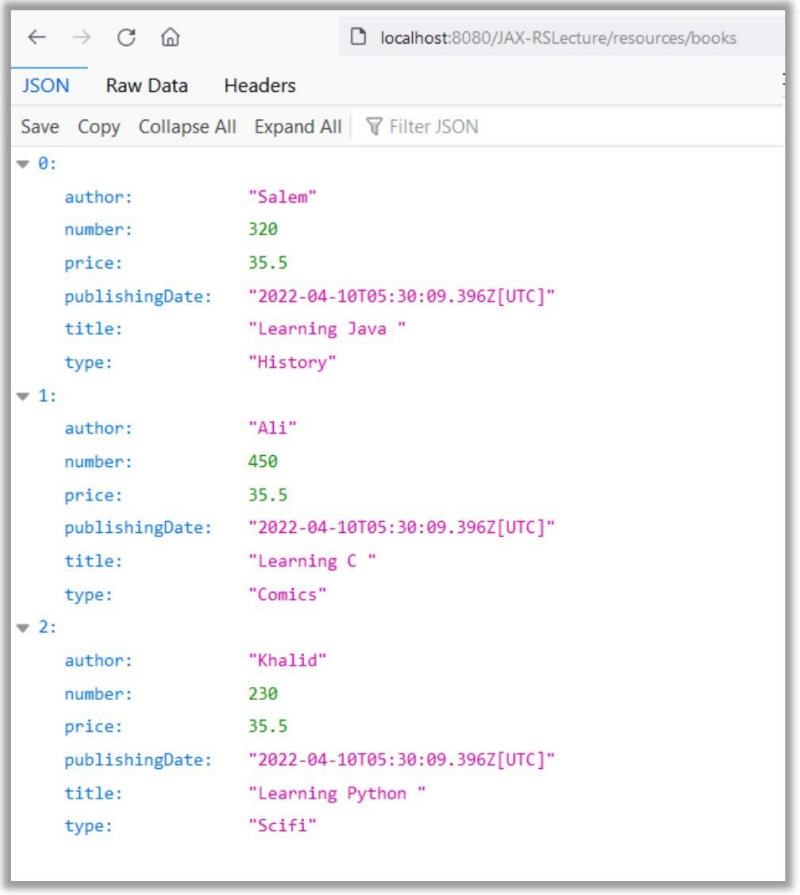
@XmlRootElement

- يتم من خلالها عمل mapping بين Class و XML Root مما يمكنا من الرد بالصيغ XML .

```java
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author elwae
 */
@XmlRootElement
public class Book {


    private String title;
    private String author;
    private Float price;
    private Date publishingDate;
    private int number;
    private String type;

    public Book() {
    }
```

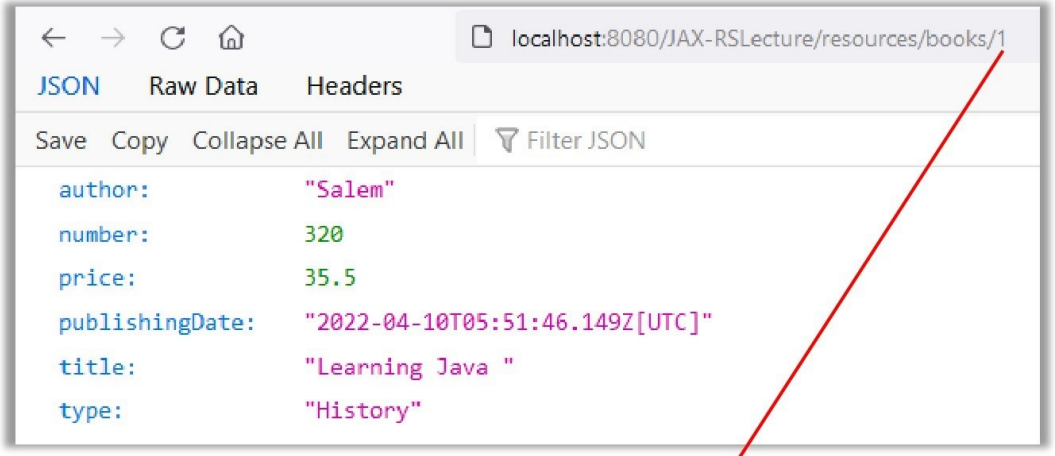
JSON Response

```java
@Path("books")
public class BooksResource {

    private List<Book> books;

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public List<Book> 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 books;
    }

}
```



@PathParam

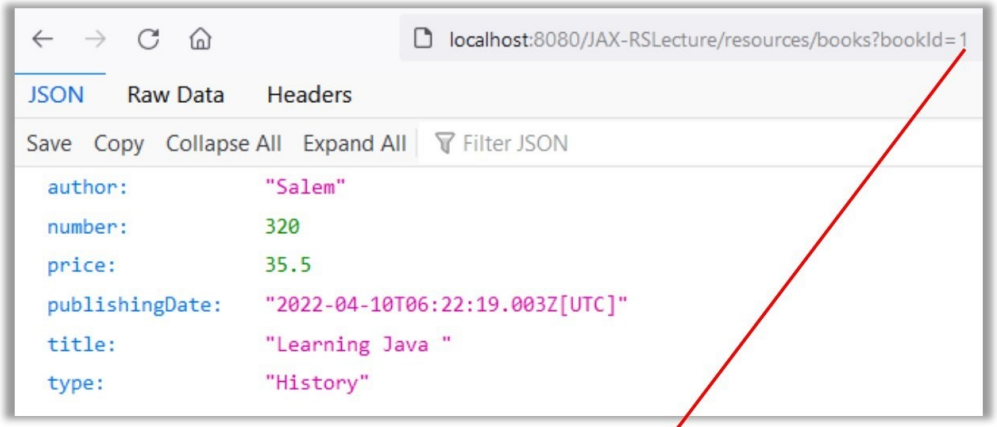
- تسمح بالحصول على القيمة الموجودة في URI template .




```
@GET
@Path("/{bookId}")
@Produces(MediaType.APPLICATION_JSON)
public Book getBook(@PathParam("bookId") 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"));
    return bookMap.get(bookId);
}
```

@QueryParam

- تستخدم للحصول على القيمة الموجودة في URI Query Parameter .




```
@GET
@Produces(MediaType.APPLICATION_JSON)
public Book getBook(@QueryParam("bookId") 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"));
    return bookMap.get(bookId);
}
```