

# RESTful服务

# 什么是RESTful服务

- RESTful服务是一种Web service
- 以资源，而非过程/操作，为核心关注点
  - 按照资源组织和管理服务接口
  - 使用HTTP标准的操作管理资源
    - GET, POST, PUT, DELETE

# 例子

一个关于订单管理的Web service提供  
若干操作，包括

- listOrders
- getOrder
- createOrder
- cancelOrder

一个关于订单管理的RESTful Web  
service提供如下访问链接

- GET, /orders/
- GET, /orders/{id}/
- POST, /orders/
- DELETE, /orders/{id}/

# 好处与局限

- 好处
  - 更加轻量级
  - 资源为中心，使用URL和HTTP访问，便于访问
- 局限
  - 如果一个服务不能抽象为某个资源，就不能（易）实现为RESTful服务
    - 比如：一个加法服务

# 搭建RESTful服务

- 依然可采取接口-实现的模式，也可直接实现服务

```
@WebService(targetNamespace="http://www.ustb.edu.cn/sei/ws")
@Path(value="/books/")
@Produces(value={MediaType.APPLICATION_XML})
public class BookService {
    .....
}
```


还可以是  
APPLICATION\_JSON

# 搭建RESTful服务

- 实现操作，设置访问方式、路径和参数映射

```
@GET
@Path(value="/")
public BookArray list() {
    BookArray arr = new BookArray();
    arr.books = books.toArray(new BookInfo[0]);
    return arr;
}
```

```
@GET
@Path(value="/{id}")
public BookInfo item(@PathParam(value="id") Integer id) {
    if(id!=null && id>=0 && id<books.size())
        return books.get(id);
    else return null;
}
```



# 搭建RESTful服务

- @PathParam, @QueryParam

```
@POST
@Path(value="/")
@Consumes(value={MediaType.APPLICATION_XML})
public Response modify(BookInfo info) {
    books.add(info);
    return Response.ok().build();
}
```

也可以是  
JSON

```
@PUT
@Path(value="/{id}/")
@Consumes(value={MediaType.APPLICATION_XML})
public Response modify(@PathParam(value="id") Integer id, BookInfo info) {
    if(id!=null && id>=0 && id<books.size())
        books.set(id,info);
    return Response.ok().build();
}
```

# 搭建RESTful服务

- @PathParam, @QueryParam

```
@DELETE
@Path(value="/{id}/")
public Response delete(@PathParam(value="id") Integer id) {
    if(id!=null && id>=0 && id<books.size())
        books.remove(id.intValue());
    return Response.ok().build();
}
```

请思考服务URL的设计方式



# 服务发布

```
JAXRSServerFactoryBean restServer = new  
JAXRSServerFactoryBean();  
restServer.setServiceBean(implementor);  
restServer.setAddress("http://127.0.0.1:9999/");  
restServer.create();
```