

2013/6/3

From: 朱越尧

关于 RestAPI 的实现、调用流程

1.在提供 RestAPI 的模块中（以 VirtualNetwork 为例），向 IRestApiServer 增加 RestletRoutable

@init 方法:

```
restApi = context.getServiceImpl(IRestApiService.class);
```

@startup 方法:

```
restApi.addRestletRoutable(new VirtualNetworkWebRoutable());
```

2.其中的 VirtualNetworkWebRoutable 实现了 RestletRoutable 接口，该接口在 basePath 中给出 basePath，在 getRestlet 中给出了不同的 URL 的服务类

@Override

```
public Restlet getRestlet(Context context) {  
    Router router = new Router(context);  
    router.attach("/tenants/{tenant}/networks", NetworkResource.class); // GET  
    .....  
    router.attachDefault(NoOp.class);  
    return router;  
}
```

@Override

```
public String basePath() {  
    return "/networkService/v1.1";  
}
```

3.上述对应 URL 的服务类 NetworkResource 继承了 ServerResource 类，在每个方法前通过 @Get、@Post、@Path 等参数对应 Http 请求

@Get("json")

```
public Collection <VirtualNetwork> retrieve() {  
    IVirtualNetworkService vns =  
        (IVirtualNetworkService)getContext().getAttributes().  
            get(IVirtualNetworkService.class.getCanonicalName());  
  
    return vns.listNetworks();  
}
```

在程序执行中，如果收到 REST 请求（curl……），RestServer 调用 RestletRoutable 接口（VirtualNetworkWebRoutable），得到对应 URL 的服务类（此处为 NetworkResource.class），然后在该类中，通过 @GET 等参数调用不同的函数，返回 String 可以显示给 Http 请求者，也可以返回一个 Collection 对象，由 RestAPI 自动转换成 JSON 串。