

RESTful Уеб Услуги

Цветан Гайдев, Иван Ст. Иванов / SAP Labs Bulgaria
Януари, 2013

Public



Начини за отдалечено извикване

Remote Method Invocation (RMI)

CORBA

Уеб услуги базирани на SOAP

Уеб услуги базирани на REST

REST архитектуры в реалния живот



Въведение в REST

REST е архитектура, не спецификация

Световната мрежа (WWW) е RESTful

Дефиниран от Roy T. Fielding

REST принципи

Всеки ресурс се идентифицира

Унифициран интерфейс за достъп

Различна репрезентация на ресурсите

HATEOAS

Дефиниране на RESTful web service

```
@Stateless
@Path("/customer/")
public class CustomerService {

    @PersistenceContext
    private EntityManager entityManager;

    public List<Customer> listCustomers() {
        Query query = entityManager.createQuery("SELECT
c from Customer c");
        return query.getResultList();
    }
}
```

Деклариране на REST метод

```
@Stateless
@Path("/customer/")
public class CustomerService {

    @PersistenceContext
    private EntityManager entityManager;

    @GET

    public List<Customer> listCustomers() {
        Query query = entityManager.createQuery("SELECT
c from Customer c");
        return query.getResultList();
    }
}
```

Деклариране на репрезентация

```
@Stateless
@Path("/customer/")
public class CustomerService {

    @PersistenceContext
    private EntityManager entityManager;

    @GET
    @Produces("application/xml")
    public List<Customer> listCustomers() {
        Query query = entityManager.createQuery("SELECT
c from Customer c");
        return query.getResultList();
    }
}
```


Достъп до резултата

```
@Stateless
@Path("/customer", )
public class CustomerService {

    @PersistenceContext
    private EntityManager entityManager;

    @GET
    @Produces("application/xml")
    public List<Customer> listCustomers() {
        Query query = entityManager.createQuery("SELECT
c from Customer c");
        return query.getResultList();
    }
}
```

GET http://localhost:8080/myapp/customer

Специфика на параметрите

@XmlElement

```
public class Customer {  
    private Long id;  
    private String customerName;  
    private String email;  
    private int discount;  
}
```

Параметризиране

```
@Path("/customer/")
public class CustomerService {

    ....

    @GET
    @Produces("application/json")
    @Path("{name}")
    public Customer getCustomer(@PathParam("name") String
name) {
        Query query = entityManager
            .createQuery("SELECT c FROM Customer c WHERE
c.customerName = :name");
        query.setParameter("name", name);
        return (Customer) query.getSingleResult();
    }
}
```

Достъп до резултата

```
@Path("/customer/")
```

```
public class CustomerService {  
    GET http://localhost:8080/myapp/customer/sap
```

```
.....
```

```
@GET
```

```
@Produces("application/json")
```

```
@Path("/{name}")
```

```
public Customer getCustomer(@PathParam("name") String  
name) {  
    Query query = entityManager  
        .createQuery("SELECT c FROM Customer c WHERE  
c.customerName = :name");  
    query.setParameter("name", name);  
    return (Customer) query.getSingleResult();  
}  
}
```

POST метод

@POST

@Consumes("application/xml")

```
public void addCustomer(@FormParam("name") String name,  
                        @FormParam("email") String email,  
                        @FormParam("discount") String discount) {
```

```
    Customer newCustomer = new Customer();  
    newCustomer.setCustomerName(name);  
    newCustomer.setEmail(email);  
    newCustomer.setDiscount(Integer.valueOf(discount));  
    entityManager.persist(newCustomer);
```

```
}
```


Достъп до резултата

```
@POST
@Consumes("application/json")
public void addCustomer(@FormParam("name") String name,
                        @FormParam("email") String email,
                        @FormParam("discount") String discount) {

    Customer newCustomer = new Customer();
    newCustomer.setCustomerName(name);
    newCustomer.setEmail(email);
    newCustomer.setDiscount(Integer.valueOf(discount));
    entityManager.persist(newCustomer);
}
```

Конфигурация

1. Чрез сървлет в *web.xml*

```
<servlet>
  <servlet-name>JAX-RS Servlet</servlet-name>
  <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>JAX-RS Servlet</servlet-name>
  <url-pattern>/rest-services/*</url-pattern>
</servlet-mapping>
```

2. Чрез *javax.ws.rs.core.Application* клас и `@ApplicationPath()`

```
import javax.ws.rs.ApplicationPath;
import javax.ws.rs.core.Application;

@ApplicationPath("rest-services")
public class ApplicationConfig extends Application {

}
```



Благодаря за вниманието!

За контакти:

ivan.ivanov@sap.com

tvetan.gaydev@sap.com

SAP Labs Bulgaria

София 1618

бул. Цар Борис III, 136А

тел: 02 91 57 690

Приложение

Тук можете да прикачите файлове с код, примери и т.н. Изтрийте текста в зелено след това.

ИСПОЛЗВАНА ЛИТЕРАТУРА

- Java SOA Cookbook, Eben Hewitt