

Arquitetura de Software

Arquitetura Broker - Rest

Felipe Fernandes¹

¹Tecnólogo em Análise e Desenvolvimento de Sistemas
FSPOA

Engenharia de Software II, 2015

1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- Princípios do Corba
- Diagrama Corba
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- Princípios do Corba
- Diagrama Corba
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

- Nesta apresentação, iremos abordar dois tipos de arquitetura de software baseada em Service-Oriented Architecture (SOA).

1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- Princípios do Corba
- Diagrama Corba
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

- Corba (Common Object Request Broker Architecture) é uma arquitetura padrão criada pelo OMG para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos.

1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- Princípios do Corba
- Diagrama Corba
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

- Rest (Representational State Transfer), trata-se de um modelo arquitetural, utilizado para o desenvolvimento web.

1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- Princípios do Corba
- Diagrama Corba
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

- A arquitetura CORBA define o ORB (Object Request Broker) como um módulo intermediário entre cliente e objeto, sendo responsável em aceitar a requisição do cliente, enviá-la para o objeto competente e, assim que disponível a resposta, entregá-la para o cliente.

1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- **Princípios do Corba**
- Diagrama Corba
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

- Object Request Broker (ORB) - manipulam as requisições dos objetos. Um intermediário entre o cliente e o servidor (objeto).

- Object Request Broker (ORB) - manipulam as requisições dos objetos. Um intermediário entre o cliente e o servidor (objeto).
- Serviços CORBA - definem serviços que ajudam a gerenciar e a manter objetos.

- Object Request Broker (ORB) - manipulam as requisições dos objetos. Um intermediário entre o cliente e o servidor (objeto).
- Serviços CORBA - definem serviços que ajudam a gerenciar e a manter objetos.
- Facilidades CORBA - definem facilidades e interfaces no nível de aplicação, manipulação de dados e armazenamento.

- Object Request Broker (ORB) - manipulam as requisições dos objetos. Um intermediário entre o cliente e o servidor (objeto).
- Serviços CORBA - definem serviços que ajudam a gerenciar e a manter objetos.
- Facilidades CORBA - definem facilidades e interfaces no nível de aplicação, manipulação de dados e armazenamento.
- Objetos de aplicação - são os objetos que podem ser considerados visíveis ao nível de aplicação.

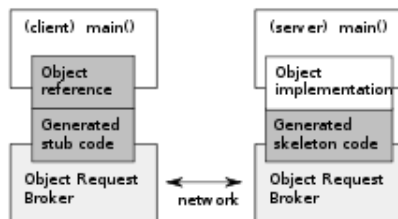
1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- Princípios do Corba
- **Diagrama Corba**
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

Diagrama



Key:



ORB vendor-supplied code



ORB vendor-tool generated code



User-defined application code

1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- Princípios do Corba
- Diagrama Corba
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

- Representational State Transfer (REST), é uma abstração da arquitetura da World Wide Web (Web), é um estilo arquitetural que consiste de um conjunto coordenado de restrições arquiteturais aplicadas a componentes.

1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- Princípios do Corba
- Diagrama Corba
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

- Em REST, tudo é definido em termos de recursos, sendo estes os conjuntos de dados que são trafegados pelo protocolo. Os recursos são representados por URI's.

Princípios do Rest

- Em REST, tudo é definido em termos de recursos, sendo estes os conjuntos de dados que são trafegados pelo protocolo. Os recursos são representados por URI's.
- Os princípios básicos de REST.

Princípios do Rest

- Em REST, tudo é definido em termos de recursos, sendo estes os conjuntos de dados que são trafegados pelo protocolo. Os recursos são representados por URI's.
- Os princípios básicos de REST.
- Uso adequado dos métodos HTTP, de URL's, de cabeçalhos HTTP.

- Em REST, tudo é definido em termos de recursos, sendo estes os conjuntos de dados que são trafegados pelo protocolo. Os recursos são representados por URI's.
- Os princípios básicos de REST.
- Uso adequado dos métodos HTTP, de URL's, de cabeçalhos HTTP.
- Uso de códigos de status padronizados para representação de sucessos ou falhas.

- Em REST, tudo é definido em termos de recursos, sendo estes os conjuntos de dados que são trafegados pelo protocolo. Os recursos são representados por URI's.
- Os princípios básicos de REST.
- Uso adequado dos métodos HTTP, de URL's, de cabeçalhos HTTP.
- Uso de códigos de status padronizados para representação de sucessos ou falhas.
- Interligações entre vários recursos diferentes.

1 Definição

- Introdução
- Corba
- Rest

2 CORBA e REST

- Definição de Corba
- Princípios do Corba
- Diagrama Corba
- Definição de Rest
- Princípios do Rest
- Exemplo Rest

Exemplo 1

```
<!--Para listar os clientes do seu sistema,
é necessário enviar o seguinte XML para o servidor: -->
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <listarClientes xmlns="http://brejaonline.com.br/administracao/1.0/service"
    </soap:Body>
  </soap:Envelope>

<!--Retorno -->
<soap:Envelope xmlns:domain="http://brejaonline.com.br/administracao/1.0/domain"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <listarClientesResponse xmlns="http://brejaonline.com.br/administracao/1.0/
      <domain:clientes>
        <domain:cliente domain:id="1">
          <domain:nome>Alexandre</domain:nome>
          <domain:dataNascimento>2012-12-01</domain:dataNascimento>
        </domain:cliente>
        <domain:cliente domain:id="2">
          <domain:nome>Paulo</domain:nome>
          <domain:dataNascimento>2012-11-01</domain:dataNascimento>
        </domain:cliente>
      </domain:clientes>
    </listarClientesResponse>
  </soap:Body>
</soap:Envelope>
```

Exemplo 2

```
<!--Para listar os clientes do sistema,  
é necessário enviar o seguinte XML para o servidor:  
  
http://localhost:8080/cevejarla/clientes .  
Ao fazer isso, você pode obter o seguinte resultado: -->  
<clientes>  
  <cliente id="1">  
    <nome>Alexandre</nome>  
    <dataNascimento>2012-12-01</dataNascimento>  
  </cliente>  
  <cliente id="2">  
    <nome>Paulo</nome>  
    <dataNascimento>2012-11-01</dataNascimento>  
  </cliente>  
</clientes>
```



D. Garlan.

Software Architecture: Perspectives on an Emerging Discipline.
Prentice Hall, 1996.



Object Management Group, Inc.

Documents Associated With CORBA, 3.3
<http://www.omg.org/spec/CORBA/3.3/>