



Http Access, JSONParser, Input/Output  
Stream e AsyncTask

Roy Fielding

Apresentou em 2000 como tese de doutorado um conjunto de regras a ser aplicada à arquitetura de sistemas distribuídos.

Esta técnica foi chamada de **Representational state transfer**, também conhecida como **RESTFull Web Services**.



## Localizável

Cada objeto ou recurso tem um endereço único

## Interface restrita e uniforme

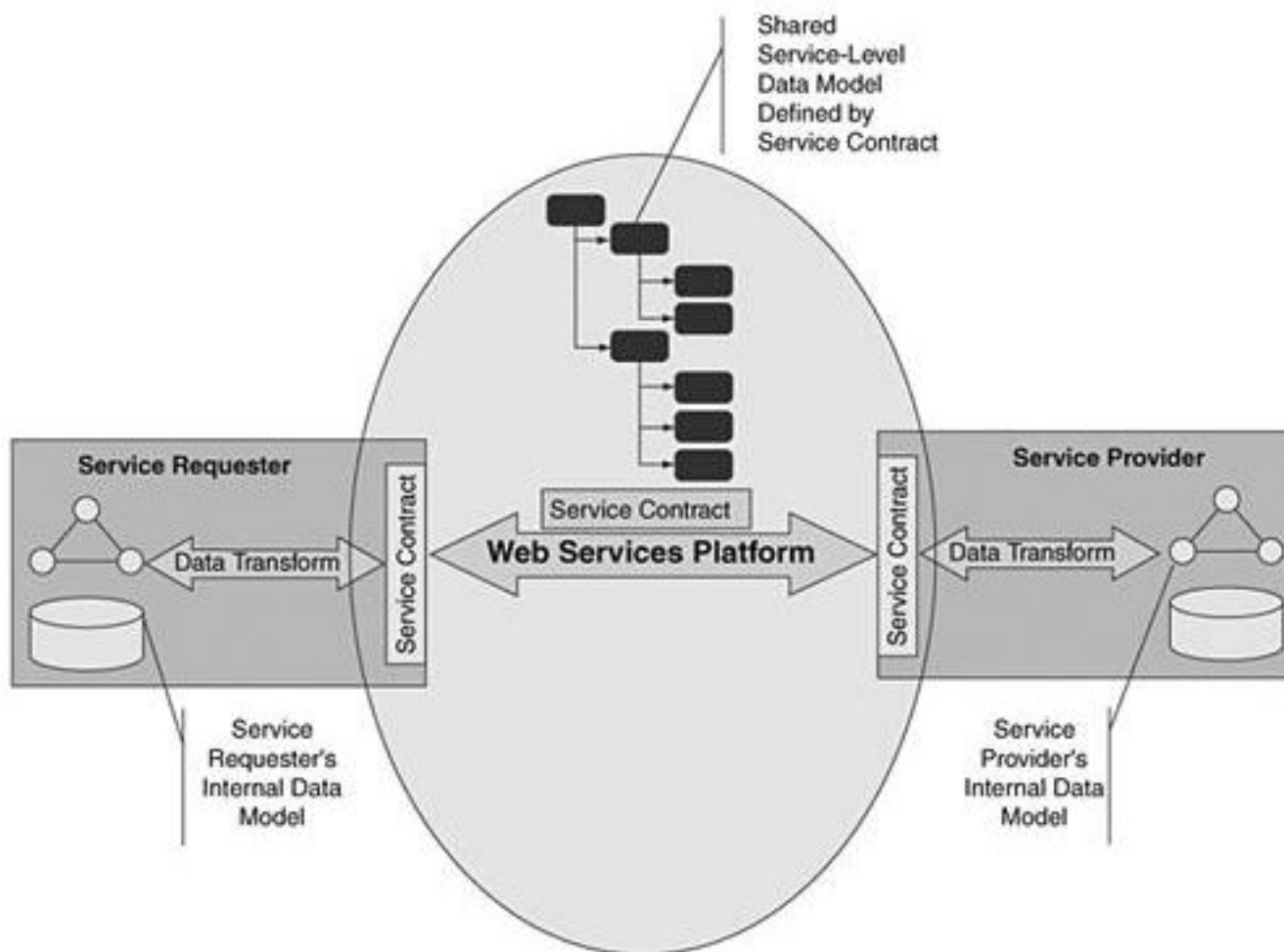
**GET** : leitura

**PUT** : atualização

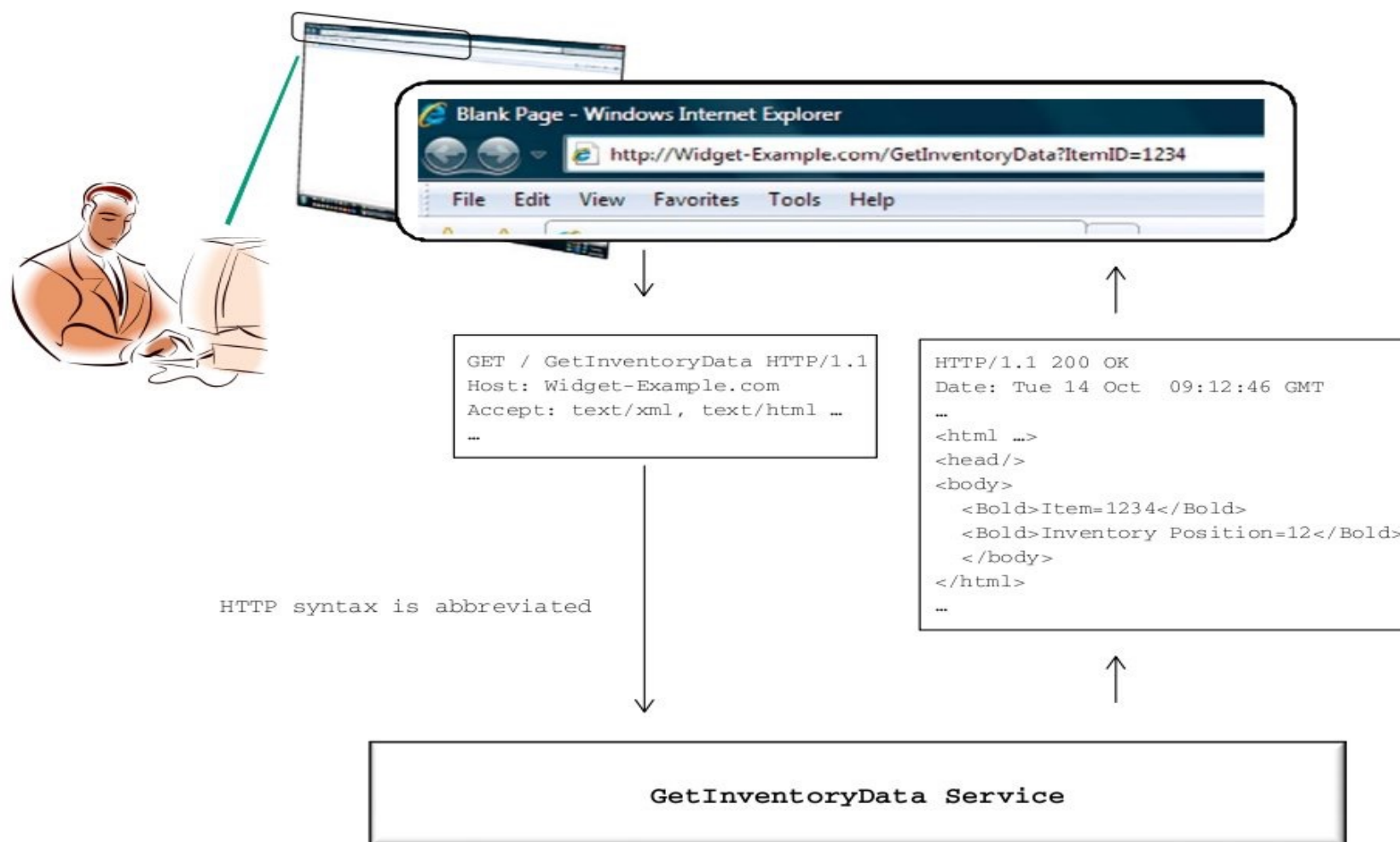
**DELETE** : remoção

**POST** : inclusão

# Arquitetura Distribuída



# Recurso Localizável



## A Requisição originada no Android

```
private final String url = "http://192.168.2.250:8080/ToDoServer/";

String json = new JSONParser.Consultar(url + "lista/" + (feito ? 1 : 0), new JSONParser.DataCallBack() {
    @Override
    public void setResponse(int code, String json) {
        if (code != 200)
            Toast.makeText(Main.context, "Falha em ler a lista de Todo", Toast.LENGTH_LONG).show();
    }
}).execute().get();
```

## A Resposta enviada pelo Back end

```
@GET
@Path("lista/{feito}")
@Produces(MediaType.APPLICATION_JSON + ";charset=UTF-8")
public Response getAll(@PathParam("feito") int feito) {
    return Response.ok(manager.createQuery("select t from Todo t where t.feito = :feito", Todo.class)
        .setParameter("feito", feito == 1)
        .getResultList()).status(200).build();
}
```

O Layout de resposta produzida pelo Back end

```
[ {  
  "id" : 4,  
  "titulo" : "Preparação da Aula de Java",  
  "feito" : false,  
  "itens" : [ {  
    "id" : 3,  
    "descricao" : "Configuração do DataSource",  
    "feito" : true  
  }, {  
    "id" : 5,  
    "descricao" : "Utilização do Entity Manager",  
    "feito" : false  
  }, {  
    "id" : 6,  
    "descricao" : "Elaborar o mapeamento XML",  
    "feito" : true  
  } ]  
} ]
```

# Interpretando o JSON

## Rotina para construção do objeto Todo

```
private Todo carregaTodo(JsonReader reader) throws IOException {
    Todo obj = new Todo();
    List<Item> itens = new ArrayList<>();

    reader.beginObject();
    while (reader.hasNext()) {
        String name = reader洗洗洗Name();
        if (name.equals("id")) {
            obj.setId(reader.nextLong());
        } else if (name.equals("titulo")) {
            obj.setTitulo(reader.nextString());
        } else if (name.equals("itens")) {
            reader.beginArray();
            while (reader.hasNext()) {
                Item item = carregaItem(reader);
                if (item != null)
                    itens.add(item);
            }
            reader.endArray();
        } else {
            reader.skipValue();
        }
    }
    reader.endObject();

    if (itens.size() > 0)
        obj.setItens(itens);

    return obj;
}
```

## Rotina para construção do objeto Item

```
private Item carregaItem(JsonReader reader) throws IOException {
    Item obj = new Item();
    reader.beginObject();

    while (reader.hasNext()) {
        String name = reader洗洗洗Name();
        if (name.equals("id")) {
            obj.setId(reader.nextLong());
        } else if (name.equals("descricao")) {
            obj.setDescricao(reader.nextString());
        } else if (name.equals("feito")) {
            obj.setFeito(reader.nextBoolean());
        } else {
            reader.skipValue();
        }
    }
    reader.endObject();
    return obj;
}
```



## A Requisição originada no Android

```
new JSONParser.Incluir(url, makeJson(todo), new JSONParser.LocationAndDataCallBack() {  
    @Override  
    public void setResponse(int code, String location, String json) {  
        if (code != 201)  
            Toast.makeText(Main.context, "Falha ao incluir o Todo", Toast.LENGTH_LONG).show();  
    }  
}).execute();
```

```
@POST  
@Consumes(MediaType.APPLICATION_JSON)  
@Produces(MediaType.APPLICATION_JSON)  
public Response addTodo(Todo obj) {  
    try {  
        List<Item> itens = obj.getItems();  
        List<Item> itensGravados = new ArrayList<>();  
  
        for (int i = 0; i < itens.size(); i++) {  
            Item item = itens.get(i);  
            item = manager.merge(item);  
            itensGravados.add(i, item);  
        }  
        obj.setItems(itensGravados);  
  
        Todo todo = manager.merge(obj);  
  
        URI uri = new URL(url + "todo/" + todo.getId()).toURI();  
  
        return Response.ok(todo).status(201).location(uri).build();  
    } catch (Exception ex) {  
        logger.log(Level.SEVERE, ex.getMessage());  
        return Response.status(500).build();  
    }  
}
```

A Resposta enviada pelo Back end

# Construindo o JSON

Rotina para construção do objeto JSON

```
private String makeJson(Todo obj) throws IOException {  
    StringWriter json = new StringWriter(100);  
    JsonWriter writer = new JsonWriter(json);  
    writer.beginObject();  
    writer.name("id").value(obj.getId());  
    writer.name("titulo").value(obj.getTitulo());  
    writer.name("feito").value(obj.isFeito());  
    writer.name("itens");  
    writer.beginArray();  
    for(Item item : obj.getItens()) {  
        writer.beginObject();  
        writer.name("id").value(item.getId());  
        writer.name("descricao").value(item.getDescricao());  
        writer.name("feito").value(item.isFeito());  
        writer.endObject();  
    }  
    writer.endArray();  
    writer.endObject();  
  
    return json.toString();  
}
```

## A Requisição originada no Android

```
new JSONParser.Alterar(url + "todo/" + todo.getId() , makeJson(todo), new JSONParser.LocationCallBack() {  
    @Override  
    public void setResponse(int code, String location) {  
        if(code != 200)  
            Toast.makeText(Main.context, "Falha em salvar o Todo", Toast.LENGTH_LONG).show();  
    }  
}).execute();
```

## A Resposta enviada pelo Back end

```
@PUT  
@Path("todo/{id}")  
@Consumes(MediaType.APPLICATION_JSON)  
public Response updateTodo(Todo obj) {  
    Todo todo = obj; //.getValue();  
    try {  
        todo = manager.merge(todo);  
  
        URI uri = new URL(url + "todo/" + todo.getId()).toURI();  
  
        return Response.ok().status(200).location(uri).build();  
    } catch (Exception ex) {  
        logger.log(Level.SEVERE, ex.getMessage());  
        return Response.status(500).build();  
    }  
}
```

## A Requisição originada no Android

```
new JSONParser.Remover(url + "todo/" + idTodo, new JSONParser.ResponseCodeCallBack() {  
    @Override  
    public void setResponse(int code) {  
        if(code != 204)  
            Toast.makeText(Main.context, "Falha em remover o Todo", Toast.LENGTH_LONG).show();  
    }  
}).execute();
```

## A Resposta enviada pelo Back end

```
@DELETE  
@Path("item/{id}")  
@Consumes(MediaType.APPLICATION_JSON)  
public Response delItem(@PathParam("id") long id) {  
    try {  
        Item item = manager.find(Item.class, id);  
        manager.remove(item);  
        return Response.noContent().build();  
    } catch(PersistenceException ex) {  
        return Response.notModified().build();  
    }  
}
```