



A Comunicação Remota por WebService

Comandos HTTP

Localizável

Cada objeto ou recurso tem um endereço único

Interface restrita e uniforme

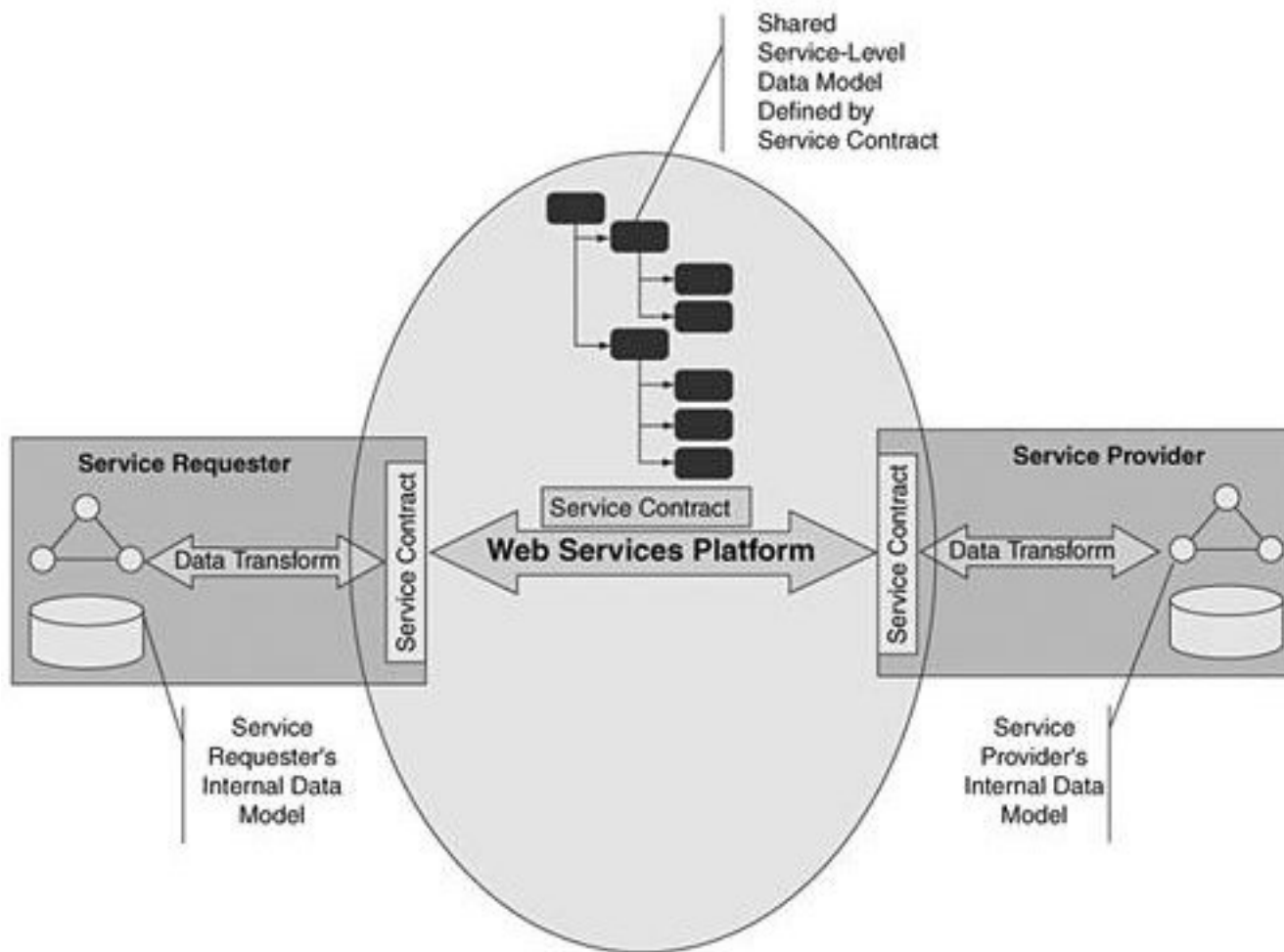
GET : leitura

PUT : atualização

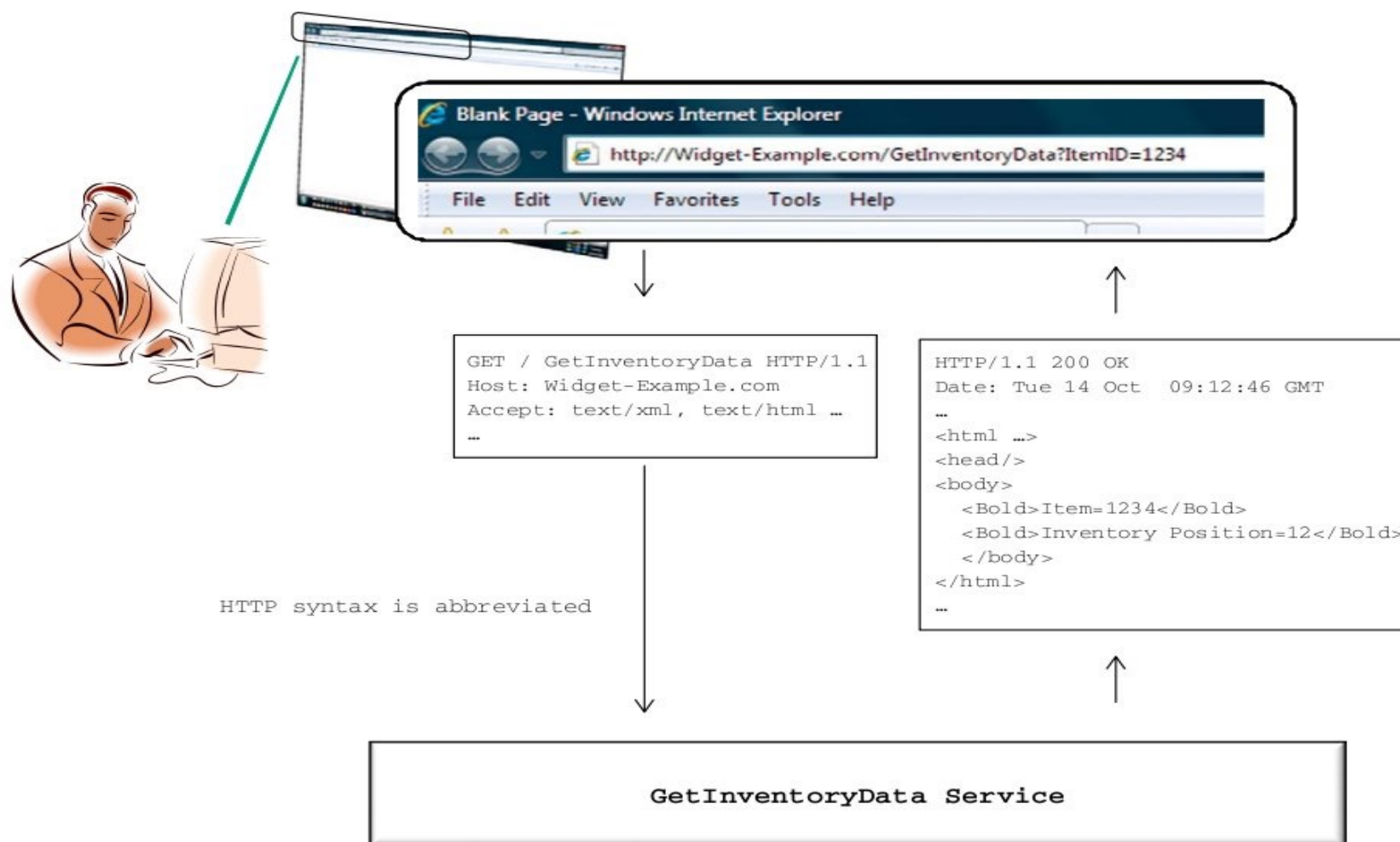
DELETE : remoção

POST : inclusão

Arquitetura Distribuída



Recurso Localizável



A Requisição originada no Android

```
// incluir
new JSONParser.Incluir(url, json, new JSONParser.LocationAndDataCallBack() {
    @Override
    public void setResponse(int code, String location, String json) {
        if(code != 201)
            Toast.makeText(Main.context, text: "Falha em salvar o Album", Toast.LENGTH_LONG).show();
    }
}).execute();
```

A Resposta enviada pelo Back end

```
// Inclui Album
@POST
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public Response addAlbum(Album obj) {
    try {
        Album Album = manager.merge(obj);
        URI uri = new URL(url + "album/" + Album.getId()).toURI();
        return Response.ok(Album).status(201).location(uri).build();
    } catch (Exception ex) {
        logger.log(Level.SEVERE, "Capa length: " + obj.getCapa().length);
        logger.log(Level.SEVERE, ex.getMessage());
        return Response.status(500).build();
    }
}
```

A Alteração

A Requisição originada no Android

```
// atualizar
new JSONParser.Alterar( url: url + "album/" + obj.getId(), json, new JSONParser.LocationCallBack() {
    @Override
    public void setResponse(int code, String location) {
        if(code != 200)
            Toast.makeText(Main.context, text: "Falha em alterar o Album", Toast.LENGTH_LONG).show();
    }
}).execute();
```

A Resposta enviada pelo Back end

```
// Altera Album
@PUT
@Path("album/{id}")
@Consumes(MediaType.APPLICATION_JSON)
public Response updateAlbum(Album obj) {
    Album album = obj;
    try {
        album = manager.merge(album);

        URI uri = new URL(url + "album/" + album.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.Remove( url: url + "album/" + id, new JSONParser.ResponseCodeCallBack() {  
    @Override  
    public void setResponse(int code) {  
        if(code != 204)  
            Toast.makeText(Main.context, text: "Falha ao excluir o Album", Toast.LENGTH_LONG).show();  
    }  
}).execute();
```

A Resposta enviada pelo Back end

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

Obter uma lista de IDs ordenada

A Requisição originada no Android

```
String json = new JSONParser.Consultar( url: url + "lista/" + ordem.toLowerCase(), new JSONParser.DataCallBack() {  
    @Override  
    public void setResponse(int code, String json) {  
        if(code != 200)  
            Toast.makeText(Main.context, text: "Falha ao consultar os Dados", Toast.LENGTH_LONG).show();  
    }  
}).execute().get();
```

A Resposta enviada
pelo Back end

```
// Lista Albums em ordem  
@GET  
@Path("lista/{ordem}")  
@Produces(MediaType.APPLICATION_JSON + ";charset=UTF-8")  
public Response listaIds(@PathParam("ordem") String ordem) {  
    String queryString = "select t.id from Album t order by t.";  
    if(ordem.equals("banda")) {  
        queryString += "banda";  
    } else if(ordem.equals("album")) {  
        queryString += "nome";  
    } else {  
        queryString += "lançamento";  
    }  
  
    return Response.ok(manager.createQuery(queryString, Long.class)  
        .getResultList()).status(200).build();  
}
```


Localizar por ID

A Requisição originada no Android

```
String json = new JSONParser.Consultar( url: url + "album/" + id, new JSONParser.DataCallBack() {  
    @Override  
    public void setResponse(int code, String json) {  
        if(code != 200)  
            Toast.makeText(Main.context, text: "Falha em localizar o Album", Toast.LENGTH_LONG).show();  
    }  
}).execute().get();
```

A Resposta enviada
pelo Back end

```
// Localizar Por ID  
@GET  
@Path("album/{id}")  
@Produces(MediaType.APPLICATION_JSON + ";charset=UTF-8")  
public Response getById(@PathParam("id") long id) {  
    Album album = manager.find(Album.class, id);  
    return Response.ok(album).status(200).build();  
}
```

Exclusão de itens marcados

A Requisição originada no Android

```
new JSONParser.Remove( url: url + "marcados", new JSONParser.ResponseCodeCallBack() {  
    @Override  
    public void setResponse(int code) {  
        if(code != 204)  
            Toast.makeText(Main.context, text: "Falha ao excluir alguns Albuns", Toast.LENGTH_LONG).show();  
    }  
}).execute();
```

A Resposta enviada
pelo Back end

```
// Remove Albuns marcados para apagar  
@DELETE  
@Path("marcados")  
public Response removeMarcados() {  
    try {  
        manager.createQuery("delete from Album t where t.del = true").executeUpdate();  
        return Response.noContent().build();  
    } catch(PersistenceException ex) {  
        return Response.notModified().build();  
    }  
}
```

Quantidade de itens marcados

A Requisição originada no Android

```
String json = new JSONParser.Consultar( url: url + "marcados", new JSONParser.DataCallBack() {  
    @Override  
    public void setResponse(int code, String json) {  
        if(code != 200)  
            Toast.makeText(Main.context, text: "Falha em consultar Albuns para apagar", Toast.LENGTH_LONG).show();  
    }  
}).execute().get();
```

A Resposta enviada
pelo Back end

```
// Quantidade marcado para apagar  
@GET  
@Path("marcados")  
@Produces(MediaType.APPLICATION_JSON + ";charset=UTF-8")  
public Response feito() {  
    return Response.ok("[{\"count\": "  
        + ((Long) manager.createQuery("select count(t) from Album t where t.del = true")  
            .getSingleResult()).intValue()  
        + "}]").status(200).build();  
}
```

Desmarcar de itens marcados

A Requisição originada no Android

```
new JSONParser.Removever( url: url + "limpar", new JSONParser.ResponseCodeCallBack() {  
    @Override  
    public void setResponse(int code) {  
        if (code != 204)  
            Toast.makeText(Main.context, text: "Falha em atualizar alguns Albuns", Toast.LENGTH_LONG).show();  
    }  
}).execute();
```

A Resposta enviada
pelo Back end

```
// Limpar Albuns marcados  
@DELETE  
@Path("limpar")  
public Response limpaMarcados() {  
    try {  
        manager.createQuery("update Album t set t.del = false where t.del = true").executeUpdate();  
        return Response.noContent().build();  
    } catch (PersistenceException ex) {  
        return Response.notModified().build();  
    }  
}
```

FIM