



POLITÉCNICO
DE LEIRIA
ESCOLA SUPERIOR
DE TECNOLOGIA
E GESTÃO

CTeSP de Programação de Sistemas de Informação

Acesso Móvel a Sistemas de Informação

Desenvolvimento Móvel em Android

Características de uma API REST e Requisitos de Acesso com Aplicações Android

Sónia Luz, sonia.luz@ipleiria.pt

David Safadinho, david.safadinho@ipleiria.pt

Departamento de Engenharia Informática

Escola Superior de Tecnologia e Gestão

Instituto Politécnico de Leiria

1º Semestre - 2021/2022

Características de uma API REST

- *API – Application Programming Interface*
(Interface de Programação de Aplicações)
- Corresponde a um conjunto de rotinas e padrões estabelecidos e documentados por uma aplicação A;
 - Para que outras aplicações consigam utilizar as funcionalidades desta aplicação A;
 - Sem precisar conhecer detalhes da implementação do software;

Características de uma API REST

- As APIs permitem uma interoperabilidade entre aplicações
 - Ou seja, a comunicação entre aplicações e entre os utilizadores;
 - O que reforça ainda mais a importância de pensarmos em algo padronizado;
 - De fácil representação e compreensão por humanos e máquinas;
 - Podem fornecer diversas formas de representação:
 - XML; JSON; YAML;

Características de uma API REST

- REST – *Representational State Transfer*
(Transferência de Estado Transacional)
- É uma abstração da arquitetura da Web;
- Consiste em princípios/regras/*constraints* que, quando seguidas, permitem a criação de um projeto com interfaces bem definidas;
 - Desta forma, permitindo, por exemplo, que aplicações comuniquem entre si;

Características de uma API REST

- Diferença entre REST e RESTful
 - Ambos representam os mesmos princípios;
 - Diferença é apenas gramatical;
 - Ou seja, sistemas que utilizam os princípios REST são designados de RESTful.
 - REST: conjunto de princípios de arquitetura;
 - RESTful: capacidade de determinado sistema aplicar os princípios de REST;

Características de uma API REST

- Princípios API REST
 - Um protocolo cliente/servidor sem estado:
 - Cada mensagem HTTP contém toda a informação necessária para compreender o pedido.
 - Como resultado: nem o cliente, nem o servidor necessitam gravar nenhum estado das comunicações entre mensagens.
 - Na prática, muitas aplicações baseadas em HTTP utilizam cookies e outros mecanismos para manter o estado da sessão (algumas destas práticas, como a reescrita de URLs, não são permitidas pela regra do REST).

Características de uma API REST

- Princípios API REST
 - Um conjunto de operações bem definidas que se aplicam a todos os recursos de informação:
 - Protocolo HTTP (POST, GET, PUT ...)

Método HTTP	Operação
POST	Criar
GET	Ler
PUT	Atualizar
DELETE	Apagar
PATCH	Atualizar parcialmente
HEADER	Mostrar header

Características de uma API REST

- *Web Service* REST
 - O principal num *web service* RESTful são:
 - Os **URL's** do sistema;
 - Os ***Resources***;
- Um ***Resource*** é um recurso, uma entidade, ou seja, um objeto com informação que será representada em **XML**;

Características de uma API REST

- *Web Service* REST
 - No geral o **URL** para aceder ao recurso será sempre o mesmo;
 - Mas se mudarmos o método HTTP, o resultado do pedido será diferente;

Método	exemplo.com/recurso	Exemplo.com/recurso/1
GET	Lista de recursos	Detalhe de um recurso
POST	Adiciona um recurso	-
PUT	-	Atualiza um recurso
DELETE	-	Remove um recurso

Características de uma API REST

- Formato de dados: **JSON x XML**
 - São formatos para a notação de dados a serem transmitidos;
 - Principal diferença é o espaço que ocupam
 - O XML ocupa muito mais espaço que o JSON
 - Quando representam o mesmo objeto;
 - Porque o XML usa uma tag para identificar o início e o fim de cada nó;
 - O JSON é muito recomendado quando falamos de dispositivos móveis:
 - Porque consome menos largura de banda da ligação de internet do utilizador;

Características de uma API REST

- JSON

- *JavaScript Object Notation* - é uma estrutura de dados para armazenamento e transmissão segura de informações no formato texto;
 - Apesar de muito simples, tem sido bastante utilizado por aplicações Web e Mobile;
 - Devido à sua capacidade de estruturar informações de uma forma compacta e leve, tornando mais rápida a leitura dessas informações;
 - Para cada valor representado, atribui-se um nome que descreve o seu significado.

Requisitos de Acesso com Aplicações Android

- Bibliotecas para acesso a API REST
 - Spring Android
 - ION
 - Retrofit
 - Android Volley

Requisitos de Acesso com Aplicações Android

- Biblioteca Volley

- Divulgado no Google I/O 2013;
- Permite programação automática de pedidos de rede;
- Várias ligações de rede simultâneas;
- Armazenamento em cache de resposta de memória e memória transparente
 - Com coerência de cache HTTP padrão;
- Suporte para priorização de solicitação;

Requisitos de Acesso com Aplicações Android

- Biblioteca Volley
 - API de solicitação de cancelamento;
 - Sendo possível cancelar um único pedido;
 - Configurar blocos de solicitações para cancelar;
 - Facilidade de personalização;
 - Ordenação que facilita o preenchimento correto da UI com dados obtidos de forma assíncrona da rede.
 - Ferramentas de depuração e *tracing*;

Requisitos de Acesso com Aplicações Android

- Biblioteca Volley
 - Existem diferentes tipos de pedidos de objetos;
 - Os mais importantes são:
 - StringRequest;
 - JsonObjectRequest;
 - JsonRequest;
 - ImageRequest.

Acesso API REST com Aplicações Android

- Definir o acesso a uma API REST através do Volley:
 - Adicionar a biblioteca Volley ao projeto
 - Colocando como dependência no ficheiro `build.gradle`

```
dependencies {  
    ...  
    implementation 'com.android.volley:volley:1.2.1'  
}
```

- Adicionar permissões de acesso a internet no `AndroidManifest.xml`

```
<uses-permission android:name="android.permission.INTERNET" />
```


Acesso API REST com Aplicações Android

- Definir o acesso a uma API REST através do Volley:
 - Definir uma *RequestQueue* através do método `Volley.newRequestQueue()`
 - Que usa valores por omissão e inicia a fila;
- Ou definir uma classe *Singleton*
 - Que faça o encapsulamento da *RequestQueue*
 - E garanta apenas uma única instância da mesma;

Acesso API REST com Aplicações Android

- Verificar se existe ligação de internet?
 - Definir permissões no **AndroidManifest.xml**

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

- Utilizar a classe **ConnectivityManager**
 - Para consultar a rede ativa;
 - Determinar se ela tem conectividade com a internet;
- Utilizar a classe **NetworkInfo**
 - Para guardar a ligação ativa;
 - Determinar o tipo de ligação;

Acesso API REST com Aplicações Android

- Exemplo:
 - Aplicação para definir acesso a uma API REST efetuando um pedido simples:
 - Através de JsonRequest;
 - Através de JsonRequest;

Desafio:

- Criar uma aplicação com ligação de rede para acesso a uma API REST
 - Usando o url que contém dados falsos,
 - Efetuar pedidos simples através de:
 - *StringRequest*;
 - *JsonObjectRequest*;
 - *JsonArrayRequest*;

Fontes e Mais Informação

- Opções de Armazenamento
 - <https://developer.android.com/training/data-storage>
- Armazenamento recorrendo a Ligação de Rede
 - <https://developer.android.com/guide/topics/connectivity>
- Introdução ao JSON
 - <http://www.json.org/json-pt.html>
- Classes JSON disponíveis no Android Studio
 - <https://developer.android.com/reference/org/json/package-summary.html>
- *Transmitting Network Data Using Volley*
 - <https://developer.android.com/training/volley/index.html>
- *Making a Standard Request*
 - <https://developer.android.com/training/volley/request.html>
- *Determinar Ligação a Internet*
 - <https://developer.android.com/training/monitoring-device-state/connectivity-status-type>

Próximo Tema:

Serviços e Tarefas Assíncronas

- Processos e *Threads*
 - <https://developer.android.com/guide/components/processes-and-threads.html>
- Serviços
 - <https://developer.android.com/guide/components/services.html>
- AsyncTask
 - <https://developer.android.com/reference/android/os/AsyncTask.html>
- Android Volley
 - <https://developer.android.com/training/volley/index.html>
- Android Volley vs AsyncTask
 - <http://www.truiton.com/2015/02/android-volley-vs-asynctask-better-approach/>