



PROGRAMAÇÃO  
PARA DISPOSITIVOS  
MÓVEIS

Ana Karina

# Programação para Dispositivos Móveis

## Aula 9 - Requisições HTTP

Ana Karina D. Salina de Oliveira

Faculdade de Computação  
Universidade Federal de Mato Grosso do Sul

Programação para Dispositivos Móveis  
2018



# Roteiro

PROGRAMAÇÃO  
PARA DISPOSITIVOS  
MÓVEIS

Ana Karina

- 1 Requisições HTTP
- 2 Thread
- 3 Handle
- 4 runOnUiThread
- 5 LogCat
- 6 Exemplo

Adaptação do material de Marcio Silva e Lucas Bedogoni  
<https://www.androidpro.com.br/web/service/>



# Android

## Requisições HTTP

PROGRAMAÇÃO  
PARA DISPO-  
SITIVOS  
MÓVEIS

Ana Karina

- As requisições HTTP servem para incluirmos informações sobre os dados que queremos consultar de um determinado computador na rede/internet.
- Na requisição incluimos um localizador de recurso, ou URL, que é o endereço dos dados que queremos.
- Uma requisição HTTP pode ser utilizada com uma URL `http://...`
  - A URL é uma String que você digita no navegador quando quer visitar uma página na Internet



# Android

## Requisições HTTP

PROGRAMAÇÃO  
PARA DISPO-  
SITIVOS  
MÓVEIS

Ana Karina

- As requisições são empacotadas para serem enviadas ao destino
- Exemplo: `https://developer.android.com/guide/index.html?hl=pt-br`
  - **http:** A primeira parte é o esquema ou protocolo. Geralmente é http ou https.
  - **developer.android.com:** Esse é o domínio, é a identidade do recurso online que você está conectando.
  - **guide/index.html:** Esse é o caminho, que age como uma estrutura de pastas no seu computador ou qualquer outra hierarquia de dados.
  - **?hl=pt-br:** A parte final chama-se query, que começa no "?". A query é uma forma opcional de incluir filtros de busca.



# Android

## Requisições HTTP

PROGRAMAÇÃO  
PARA DISPO-  
SITIVOS  
MÓVEIS

Ana Karina

### Requisições HTTP no Android

- Para fazer requisições HTTP a um WebService em aplicativos Android precisa-se de duas classes da plataforma Android:
  - HttpURLConnection
  - AsyncTask.

## Adicionando permissões

- É preciso adicionar permissões ao AndroidManifest para habilitar no aplicativo o acesso a Internet do dispositivo.
- Exemplo:

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

## Adicionando permissões

- Acrescentar as permissões na Estrutura geral do arquivo de manifesto

```
<?xml version="1.0"encoding="utf-8"?>
<manifest>
<uses-permission />
<permission />
<application>
  <activity>
    <intent-filter>
      <action />
      <category />
    </intent-filter>
    <meta-data />
  </activity>
</application>
</manifest>
```



## Segurança com HTTPS e SSL

- O Secure Sockets Layer (SSL) — agora conhecido como Transport Layer Security (TLS)
  - é uma base comum para estabelecer comunicações criptografadas entre clientes e servidores.
- É possível que um aplicativo use o SSL incorretamente
  - permitindo que entidades mal-intencionadas consigam interceptar seus dados na rede.





## Segurança com HTTPS e SSL

- Os servidores geralmente são configurados com emissores conhecidos, chamados Autoridades de Certificação (CAs).
- A plataforma do host geralmente contém uma lista de CAs conhecidas nas quais ela confia.
- A partir do Android 4.2 (Jelly Bean), o Android contém mais de 100 CAs que são atualizadas a cada versão.
- Semelhante a um servidor, uma CA tem um certificado e uma chave privada.
  - Ao emitir um certificado para um servidor, a CA assina o certificado do servidor usando sua chave privada.
  - Então, o cliente pode verificar se o servidor tem um certificado emitido por uma CA conhecida pela plataforma.

- O uso da `URLConnection` com suporte para recursos de HTTP:
  - 1 Obtenha um novo `HttpURLConnection` chamando `URL.openConnection()`.
  - 2 Prepare o pedido.
  - 3 Opcionalmente, envie um corpo da solicitação. As instâncias devem ser configuradas com `setDoOutput(true)` e a transmissão dos dados é feita dados com `getOutputStream()`.
  - 4 Leia a resposta. O corpo da resposta pode ser lido no fluxo retornado por `getInputStream()`. Se a resposta não tiver corpo, esse método retornará um fluxo vazio.
  - 5 Desconecte chamando `disconnect()`. Desconectar libera os recursos mantidos por uma conexão.



# Android

## Requisições HTTP

PROGRAMAÇÃO  
PARA DISPOSITIVOS  
MÓVEIS

Ana Karina

### Criando a requisição HTTP

- Baixando uma figura de um endereço HTTP

```
URL url = new URL("http://example.com/image.png");
URLConnection connection;
connection = (URLConnection) url.openConnection();
connection.setDoInput(true);
connection.connect();
InputStream input = connection.getInputStream();
final Bitmap imagem = BitmapFactory.decodeStream(input);
connection.disconnect();
```

## Processos e Threads

- Quando um aplicativo inicia, o sistema Android inicia um novo processo no Linux com um único thread(encadeamento de execução).
- Por padrão, todos os componentes do mesmo aplicativo são executados no mesmo processo e na mesma thread.
- Se um componente de aplicativo iniciar e já existir um processo, ele será iniciado dentro deste processo e usará o mesmo thread.
- No entanto, é possível criar Threads adicionais para qualquer processo.

## Encadeamentos de trabalho

- Por causa do modelo de thread única, é essencial para a capacidade de resposta da IU, que você não bloqueie a thread da IU.
- Caso tenha operações a realizar que não sejam instantâneas, você deverá se certificar de fazê-las em threads separadas.

```
public void onClick(View v) {  
    new Thread(new Runnable() {  
        public void run() {  
            Bitmap b = loadImageFromNetwork  
                ("http://example.com/image.png");  
            mImageView.setImageBitmap(b);  
        }  
    }).start();  
}
```



# Android

## Threads

PROGRAMAÇÃO  
PARA DISPO-  
SITIVOS  
MÓVEIS

Ana Karina

- O exemplo anterior viola a regra do modelo de thread única:
  - não acessar o kit de ferramentas de IU do Android de fora da thread da IU
- esse exemplo modifica o ImageView a partir da nova thread que foi criada.
- Isso pode resultar em comportamentos inesperados e indefinidos, que podem ser difíceis de rastrear e consumir bastante tempo.
- Para fazer da forma correta, pode-se usar um **Handler** ou o método **runOnUiThread** para processar as mensagens entregues para thread da IU.



## Handler

- Um Handler permite enviar e processar objetos associados a um thread.
- Cada instância do Handler está associada a um único encadeamento.
- Quando você cria um novo Handler, ele é vinculado à thread que o criou
- Existem dois usos principais para um Handler:
  - 1 para agendar mensagens a serem executadas;
  - 2 enfileirar uma ação a ser executada em um segmento diferente do seu.



# Android

## Threads

PROGRAMAÇÃO  
PARA DISPOSITIVOS  
MÓVEIS

Ana Karina

## Handler

```
public class MainActivity extends Activity {  
    private Handler handler = new Handler();  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
    public void onClick(View v) {  
        new Thread(new Runnable() {  
            ....  
            handler.post(new Runnable() {  
                @Override  
                public void run() {  
                    mImageView.setImageBitmap(imagem);  
                }  
            });  
            ....  
        })  
    }  
}
```





# Android

## Threads

PROGRAMAÇÃO  
PARA DISPOSITIVOS  
MÓVEIS

Ana Karina

### runOnUiThread

- Também é possível usar `runOnUiThread()` para atualizar a interface do usuário de um segmento não-UI, como a partir de um thread.
- `runOnUiThread` executa a ação especificada no thread da interface do usuário.

```
runOnUiThread(new Runnable() {  
    @Override  
    public void run() {  
        mImageView.setImageBitmap(imagem);  
    }  
});
```



## LogCat

- O Logcat é uma ferramenta de linha de comando que gera um registro:
  - de mensagens do sistema,
  - rastros de pilha quando o dispositivo apresenta um erro
  - mensagens escritas pelo seu aplicativo com a classe Log.

## LogCat

- Cada mensagem do registro do Android é associada a uma tag e uma prioridade.
- A tag de uma mensagem de registro do sistema é uma string curta que indica o componente do sistema que originou a mensagem (por exemplo, `ActivityManager`).
- Uma tag definida pelo usuário pode ser qualquer string, como o nome da classe em questão (a tag recomendada).
- Pode-se definir chamando o método `Log`, por exemplo:
  - `Log.d(tag, message);`



## LogCat

- A classe LogCat contém métodos para imprimir informações como:
  - informações (i);
  - debug (d);
  - warning (w);
  - erro (e).



# Android

PROGRAMAÇÃO  
PARA DISPOSITIVOS  
MÓVEIS

Ana Karina

Exemplo com HTTP, Handle, runOnUiThread, LogCat