PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS

COMUNICAÇÃO WEB

Comunicação Web

- Uma das tarefas fundamentais do desenvolvimento mobile
- Por que é tão importante?
- Exemplos?

Conexão HTTP

- Processo semelhante ao de recuperar imagens:
- Declarar objetos URL e InputStream
- No entanto, também é necessário um objeto de conexão HTTP:
- Classe HttpURLConnection
- Executa requisições GET e POST

Conexão HTTP

```
URL url = new URL("http://endereco");

HttpURLConnection con =
        (HttpURLConnection) url.openConnection();

InputStream in = con.getInputStream();
```

Recebendo a Resposta

- A resposta é fornecida pelo objeto InputStream
- É necessário criar um BufferedReader e um InputStreamReader
- Em seguida, ler linha a linha do BufferedReader e jogar em uma String

Exemplo de Recebimento de Resp.

```
BufferedReader streamReader = new BufferedReader(
        new InputStreamReader(in, "UTF-8"));


StringBuilder responseStrBuilder = new
StringBuilder();


String inputStr;
while ((inputStr = streamReader.readLine()) != null)
    responseStrBuilder.append(inputStr);


String result = responseStrBuilder.toString();
```

Parâmetros

- É possível configurar parâmetros de conexão:
- Timeout de conexão em milissegundos
- Timeout de leitura em milissegundos
- Os parâmetros precisam ser configurados antes de chamar o método “getInputStream()”

Parâmetros

```
HttpURLConnection con =
        (HttpURLConnection) url.openConnection();

con.setConnectTimeout(15000);
con.setReadTimeout(30000);

InputStream in = con.getInputStream();
```

Enviando Dados

- É possível enviar dados de duas formas:
  - Utilizando parâmetros de URL com o método GET
  - Utilizando parâmetros POST

Enviando Dados Via GET

- Para enviar dados via GET, basta acrescentá-los a URL, seguindo o padrão HTTP:
- http://fabiohenriqueaf.esy.es/getObject.php**?num1=12&num2=23**

Enviando Dados Via POST

- Para enviar dados via POST, é necessário:
  - Codificar os parâmetros usando “URLEncoder.encode()”
  - Chamar o método “setDoOutput(true)” do HttpURLConnection
  - Criar um “OutputStreamWriter”
  - Escrever os parâmetros no “OutputStreamWriter”
  - Fechar o “OutputStreamWriter”

Exemplo de Envio de Dados Via POST

```
String parametro = URLEncoder.encode("parâmetro", "UTF-
8");


URL url = new URL("http://endereco.com");
HttpURLConnection con =
        (HttpURLConnection) url.openConnection();

con.setDoOutput(true);


OutputStreamWriter out = new OutputStreamWriter(
        con.getOutputStream());

out.write("param=" + parametro);
out.close();


InputStream in = con.getInputStream();
```

Trabalhando com JSON

- Se a resposta for um objeto JSON, jogar a String em um JSONObject
- Use o método “get” para recuperar uma informação do JSONObject (existem métodos para cada tipo)

Recuperando valor JSON

```
JSONObject json = new JSONObject(result);
String resposta = json.getString("info");
```

Trabalhando com JSON

- Se a resposta for um array em JSON, jogar a String em um JSONArray
- Use o método “get” para recuperar as informações na posição especificada do JSONArray

Recuperando array JSON

```
JSONArray jsonArray = new JSONArray(result);
jsonArray.get(0);
```

Permissão

- Como com o exemplo de imagens, para realizar qualquer comunicação com a web no Android é necessário dar permissão de Internet no AndroidManifest.xml

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

Observações

- Toda a execução da requisição deverá ocorrer em uma thread separada, conforme visto no download de imagens.


"That's all Folks!"