

SOCKETS URL

DISCIPLINA DE SISTEMAS DISTRIBUÍDOS

PROF. MARCELO LISBOA ROCHA - UFT

SOCKET URL

- Representa uma **URL**
- Principais métodos
 - **openStream()** obtém um *InputStream* para os dados
 - **openConnection()**: retorna um objeto *URLConnection* que contém métodos para ler o cabeçalho dos dados
 - **getContent()**: retorna os dados diretamente como *Object* se conteúdo for conhecido (texto, imagens, etc.)
- Para imprimir a página HTML de um site

```
try {  
    URL url = new URL("http://www.site.com");  
    InputStreamReader reader =  
        new InputStreamReader(url.openStream());  
    BufferedReader br = new BufferedReader(reader);  
    String linha = "";  
    while ( (linha = br.readLine()) != null) {  
        System.out.println(linha);  
    }  
} catch (MalformedURLException e) { ... }
```

"http://www.site.com:5678"
"http://www.site.com",5678

EXCEÇÕES DE REDE

- Várias **exceções** podem ocorrer em um ambiente de rede
 - O programa deve tomar medidas para reduzir o impacto das exceções inevitáveis, como rede fora do ar ou conexão recusada
 - O compilador irá informar, durante o desenvolvimento, as exceções que precisam ser declaradas ou tratadas
- As exceções mais comuns do pacote `java.net` são
 - **`SocketException`**
 - **`MalformedURLException`**
 - **`UnknownHostException`**
 - **`ProtocolException`**
- Operações de timeout, liberação de threads, sincronização, transações, etc. devem ser implementados pelo programador em aplicações de rede
 - Não há exceções tratando esses problemas automaticamente

```
try{
    .....
}catch(Exception e){
    System.out.println("\nErro="+e.getMessage());
}
```