WEBSERVICES: SERVIDOR

Necessário definir código do servidor

- Métodos com diferentes parâmetros devem ter nomes diferentes
- Classe/métodos anotados: @WebService, @WebMethod

```
@WebService
public class FileServerImplWS {
        private File basePath;
        public FileServerImplWS() { basePath = new File(“.”); }
        public FileServerImplWS( String p) { basePath = new File(p); }

        @WebMethod
        public String[] dir(String path) throws InfoNotFoundException {
            File f = new File( basePath, path);
            if( f.exists())
                return f.list();
            else
                throw new InfoNotFoundException( "Directory not found”);
        }
        ...
}
```

WEBSERVICES: SERVIDOR (CONT.)

Código de criação do servidor

- Descriptor do servidor: Tomcat, etc.
  - Complexo manualmente
- Java 6.0 inclui pequeno servidor que pode ser usado de forma simples

```
import javax.jws.*;
import javax.xml.ws.*;

@WebService
public class FileServerImplWS {
  ...
  @WebMethod(exclude=true)
  public static void main(String[] args) {
      Endpoint.publish(
         "http://localhost:8080/FileServer",
         new FileServerImplWS());
  }

}
```

WebServices: servidor (cont.)

Criar servidor

1. Compilar classes do servidor
2. Gerar classes *web services* (opcional)

   wsgen -cp bin -s src -d bin aula3.srv.FileServerImplWS

   Opção –wsdl gera ficheiro wsdl para visualização
   [Ficheiro WSDL tem descrição do serviço, e pode ser usado para gerar o código do cliente que invoca os métodos do servidor – mais informação nas aulas teóricas]

Após colocar servidor a executar, WSDL disponível em

http://localhost:8080/FileServer?wsdl

ASPECTOS IMPORTANTES

Não é possível passar como parâmetros/resultados alguns objectos definidos na biblioteca do Java (e.g. InetAddress, etc.)

- Mas é possível passar estruturas de dados (e.g. List)
- Arrays são transformados em List

Não é possível definir mais do que uma função com o mesmo nome

Servidor deve ter construtor público

CLIENTE

1. **Gerar classes *web services*** a partir de descrição do serviço (WSDL)

   wsimport -d bin -s src -p aula3.clt.ws http://localhost:8080/FileServer?wsdl

2. Criar código do cliente

```
import aula3.clt.ws.*;

public class GetDirList {
   public static void main( String[] args) {
      try {
         FileServerImplWSService service = new FileServerImplWSService();
         FileServerImplWS server = service.getFileServerImplWSPort();

         List<String> files = server.dir( args[0]);
         for( int i = 0; i < files.size(); i++)
            System.out.println( files.get(i));
      } catch (InfoNotFoundException_Exception ex) {
         System.out.println( "Info sobre " + args[0] + " nao disponivel");
      } catch (Exception ex) { ex.printStackTrace(); }
   }
}
```

Outras formas de obter referências para servidores

Referência obtida a partir dum URL

```
...
FileServerImplWSService service = new FileServerImplWSService(
      new URL( "http://" + serverHost + "/FileServer?wsdl"),
      new QName("http://srv.aula3/", "FileServerImplWSService"));
FileServerImplWS server = service.getFileServerImplWSPort();

...
```

Referência obtida a partir de uma anotação

Apenas válido quando se executa num application server (e.g. GlassFish)

```
@WebServiceRef(wsdlLocation="http://localhost:8080/FileServer?wsdl")
private static FileServerImplWSService service;
```