

# **Serviços e Interoperabilidade de Sistemas**

**Representação agnóstica de dados  
JSON**

# Representação agnóstica de dados

## A necessidade: exemplo motivacional [1]

Sistema bancário on-line:

Permite depósitos na sua conta a partir de outros bancos

Aplicação web/mobile que liga ao *backend* da autoridade de transferências de fundos (e.g. SIBS)

Simplificação: vamos assumir que para transferências é apenas necessário o nome e o montante

# Representação agnóstica de dados

A necessidade: exemplo motivacional

**Primeiro: a ligação (protocolo)?**

TCP/IP, porque não? É um protocolo ubíquo...

Suporte em todos os sistemas operativos e linguagens de programação...

# Representação agnóstica de dados

## A necessidade: exemplo motivacional

```
String hostName = "www.sibs.pt";  
int port = 80;  
IPHostEntry hostInfo = Dns.GetHostByName(hostName);  
IPAddress address = hostInfo.AddressList[0];  
IPEndPoint endpoint = new IPEndPoint(address, port);  
Socket socket = new Socket(address.AddressFamily,  
SocketType.Stream, ProtocolType.Tcp);  
socket.Connect(endpoint);  
int val=1000;  
byte[] amount = BitConverter.GetBytes(val);  
byte[] name = Encoding.ASCII.GetBytes("Joe");  
int bytesSent = socket.Send(amount);  
bytesSent += socket.Send(name);  
socket.Close();
```

[1]

7 bytes enviados. "Simples!"  
**Como podem dizer que a integração  
é um processo complicado?**

# Representação agnóstica de dados

A necessidade: exemplo motivacional

Problemas:

**1. Dependência da plataforma (são usados dados binários!)**

Depende da arquitetura da máquina (tamanho dos dados)

32bits vs 64 bits!

Se envio de máquina de 32 bit e destino de 64 bit, o nome do cliente é interpretado como parte do montante!

# Representação agnóstica de dados

A necessidade: exemplo motivacional

Problemas:

1. Dependência da plataforma (são usados dados binários!)

**Ordem dos bytes em memória: little-endian vs big-endian**

Numbers: highest byte first vs lowest byte first

232 3 0 0 **vs** 0 0 3 232

# Representação agnóstica de dados

A necessidade: exemplo motivacional

Problemas:

**2. Vamos supor que o sistema é mudado para outro domínio**

Código está demasiado ligado a  
**"www.sibs.pt"**

# Representação agnóstica de dados

A necessidade: exemplo motivacional

Problemas:

## **3. Dependências temporais**

TCP/IP é orientado à ligação

Várias mensagens trocadas só para estabelecer a ligação

A aplicação cliente e a servidora têm que existir ao mesmo tempo!

Se erro de rede, os dados não são enviados  
“Sistemas demasiados dependentes”



# Representação agnóstica de dados

A necessidade: exemplo motivacional

Problemas:

## 4. Formato de dados

4 bytes (montante) + sequência de chars (nome)

Vamos supor que pretendemos **adicionar** um novo dado à mensagem: **designação da moeda**

Cliente e servidor têm de ser alterados!

# Representação agnóstica de dados

## Representação agnóstica de dados

A partir de agora vamos apenas focar no problema do **formato de dados**

# JSON: JavaScript Object Notation

## O que é o JSON?

**Objetivo: representação de dados standard/agnóstica**

Lembrar: dados binários => muito ligados ao fabricante

Abrindo num editor de texto... “lixo”

Formato tem que ser conhecido (descoberto?)

Vantagem: menos espaço e menos largura de banda!

# JSON: JavaScript Object Notation

O que é o JSON?

**E em relação a ficheiros de texto?**

Legíveis por humanos e máquinas

Atenção ao nº de bits por símbolo(s)

E em relação aos *encodings*?

ANSI, UTF-8, etc...

Vamos assumir que a **primeira linha** o define

Desv: mais espaço (disco e largura de banda)



# JSON: JavaScript Object Notation

O que é o JSON?

E em relação a ficheiros de texto?

**E se adicionássemos meta-dados?**

Dados que apenas caracterizam os dados originais

# JSON: JavaScript Object Notation

O que é o JSON?

**Meta-dados é a solução!**

Circundar o texto com informação extra

Contudo...

Como diferenciar os meta-dados dos dados?

Que meta-dados são permitidos

HTML é um bom exemplo (header, body, tags, etc...)

# Introdução ao JSON

O que é o JSON?

O JSON é baseado em pares **CHAVE : VALOR**

Labrador;12;33.56      **Exemplo de dados sem meta-dados**

Versão JSON

```
{ "raça": "labrador", "idade": 12, "peso": 33.56 }
```

O tamanho dos dados aumentou!

# Introdução ao JSON

O que é o JSON?

Representação de vetores/arrays

```
{  
  "cães": [  
    {"raça": "labrador", "idade": 12, "peso": 33.56},  
    {"raça": "beagle", "idade": 10, "peso": 23.50},  
    {"raça": "boxer", "idade": 2, "peso": 6.30}  
  ]  
}
```



# Introdução ao JSON

O que é o JSON?

Tipos de dados suportados pelo JSON

Tipo	Exemplo
Número	{ "peso" : 12.1 }
String	{ "nome" : "ola" }
Booleano	{ "valido": true/false }
Vetor	[ "aaa", "bbb", "ccc", ].
Objeto	{ "país" : "PT", "Naturalidade": "Leiria", ... }
Null	Vazio

# Introdução ao JSON

O que é o JSON?

Outras vantagens:

**Extensibilidade**

Facilmente se adicionam campos

```
{ "raça": "labrador", "cor": "bege", "idade": 12, "peso": 33.56 }
```



E as aplicações já existentes?

# Introdução ao JSON

O que é o JSON?

## **Cenários para uso do JSON**

Configuração de aplicações em ficheiros

Web Services (serialização)

Importância para o AJAX (sem necessidade de processamento quando comparado com XML)

O JSON já é Javascript!

Etc.

# Introdução ao JSON

## JSON na linguagem de programação Java

Jackson (biblioteca externa)

Download: <http://www.java2s.com/Code/JarDownload/jackson-all/jackson-all-1.9.0.jar.zip>

jackson-all-x.x.x.jar => guardar na pasta do projeto

### **Adicionar ao projeto:**

Clique direito em **Libraries** (pasta do projeto, no NetBeans)

**Add Jar** e seleccionar o .jar descarregado

# Introdução ao JSON

JSON na linguagem de programação Java

**Jackson: import e ObjectMapper**

```
import org.codehaus.jackson.*;
```

```
...
```

```
//ObjectMapper pertence ao package jackson!
```

```
//Permite serializar e des-serializar dados/JSON
```

```
ObjectMapper mapper = new ObjectMapper();
```

# Introdução ao JSON

## JSON na linguagem de programação Java

### Exemplo

#### **class Student**

```
{  
    private String name;  
    private int age;  
  
    public Student()  
    {  
    }  
    public String getName()  
    {  
        return name;  
    }  
}
```

```
    public void setName(String name)  
    {  
        this.name = name;  
    }  
    public int getAge()  
    {  
        return age;  
    }  
    public void setAge(int age)  
    {  
        this.age = age;  
    }  
    public String toString()  
    {  
        return "Student [ name: "+name+", age: "+ age+ " ]";  
    }  
}
```

# Introdução ao JSON

## JSON na linguagem de programação Java

### Exemplo

```
ObjectMapper mapper = new ObjectMapper();  
String jsonString = "{\"name\":\"Joao\", \"age\":21}\";  
  
//map json to student  
Student student = mapper.readValue(jsonString, Student.class);  
System.out.println(student);  
  
//mapper.enable(SerializationConfig.Feature.INDENT_OUTPUT);  
jsonString = mapper.writeValueAsString(student);  
System.out.println(jsonString);
```

# Introdução ao JSON

## JSON na linguagem de programação Java

### Exemplo para tipos simples String ( Map<String,String> )

```
String str=new String("{\"marca1\":\"Volvo\",\"marca2\":\"Honda\"}");
```

```
Map<String,String> map = new HashMap<String,String>();  
map = mapper.readValue(str, HashMap.class);
```

```
String jstr = new String();  
jstr=mapper.writeValueAsString(map);
```

```
System.out.println(jstr);
```

```
{  
  "marca1" : "Volvo",  
  "marca2" : "Honda"  
}
```



# Introdução ao JSON

JSON na linguagem de programação PHP

Resume-se ao uso de duas funções:

`json_encode`  
`json_decode`



# Introdução ao JSON

## JSON na linguagem de programação PHP

Exemplo de conversão de JSON para (objeto) PHP:

```
<?php

$json = '{"titulo": "JSON for Beginners",
        "autor": "ESTG",
        "edição": 2
}';

$livro = json_decode($json); //Para objeto PHP

echo $livro->titulo;
?>
```

JSON for Beginners



# Introdução ao JSON

## JSON na linguagem de programação PHP

Exemplo de conversão de PHP para JSON:

```
<?php
class pessoa {
    public $nome = "";
    public $hobbies = "";
    public $nascimento = "";
}

$p = new pessoa();
$p->nome = "joao";
$p->hobbies = "desporto";
$p->nascimento = "8/5/1974 12:20:03";
echo json_encode($p);
?>
```

```
{"nome":"joao","hobbies":"desporto","nascimento":"8/5/1974 12:20:03"}
```