

INSTITUTO FEDERAL
PIAUÍ
Campus Parnaíba

Json e Bson

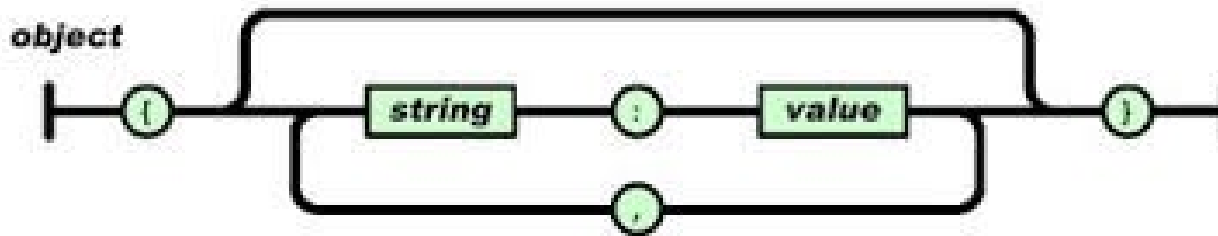
Prof. Msc Denival A. dos Santos

JSON - Introdução

- Java Script Object Notation, ou notação de objetos java script.
- Formato de serialização de dados, tal qual YAML e XML.
- Um formato de serialização é o que permite tranferir um objeto mais ou menos complexo (como um dicionário cheio de valores ou, meramente, um número inteiro) por um transporte simples e não-tipado (um socket, um pipe ou até um simples arquivo).
- Criado para o armazenamento e a troca de dados entre plataformas.
- Baseada em JavaScript, mais armazenada como texto simples (text plain).

JSON - Introdução

- A sintaxe do JSON (conforme consta no site json.org) é bem simples.
- Cada objeto JSON é delimitado por “{ }” (Chaves).
- O delimitador entre Chave-Valor é o “:” (dois pontos).
- Cada entrada “Chave-Valor” é separada por “,” (virgula).
- As chaves são do tipo “String”.



- Exemplos:

`{"alfa": 1}`

`{"alfa": 1, "beta": 2, "gama": "três", "delta": {}}`

Tipos de dados

- O padrão Json aceita como valor os seguintes tipos de dados:
 - Strings
 - Numeros
 - Booleanos (true/false)
 - Null
 - Arrays
 - Objects/documents

Json - representações

- Para representar um número
“Idade”:10
- Para representar um número real
“altura”:1.72
- Para representação de string
“site”: “denivalsantos.webnode.com”
- Para representação de número negativo
“temperatura”:-2
- Para representar um valor booleano
“casado”:true

Json - representações

- Para representar valores nulos

“salario”: null

- Para representar Array utilizamos colchetes com os elementos separados por vírgula.

- Array de strings

[“CE”, “PI”, “MA”]

- Matriz de inteiros

[

[1,5],

[-1,9],

[100,0]

]

Json - Tipos de dados

▪ Exemplo

```
{  
  "nome":"João", //string  
  "idade":21, //number  
  "eleitor":true, //boolean  
  "escolaridade": null, //null  
  "hobbies":["tênis","xadrez"], //array  
  "endereço":{  
    "cidade":"São Paulo",  
    "estado":"SP"  
  } //object/document  
}
```

Json

- Vamos criar um objeto pessoa com os campos:
 - Nome, idade, sexo, materias
- Abra um navegador que não seja o IE e aperte F12, em seguida, vá a aba console.

```
> var pessoa = {  
    "nome": "Denival",  
    "idade": 40,  
    "sexo": "M",  
    "materias": ["IBD", "PBD", "TEIT", "TBD"]  
}
```

```
< undefined
```

```
> |
```


Json

```
> pessoa
< ▼ {nome: "Denival", idade: 40, sexo: "M", materias: Array(4)} ⓘ
  idade: 40
  ▶ materias: (4) ["IBD", "PBD", "TEIT", "TBD"]
    nome: "Denival"
    sexo: "M"
    ▶ __proto__: Object

> pessoa.nome
< "Denival"

> pessoa.sexo
< "M"


> pessoa.materias
< ▶ (4) ["IBD", "PBD", "TEIT", "TBD"]

> pessoa.materias[0]
< "IBD"


> pessoa.sexo == "M"
< true
```

Json - representações

- Os objetos são especificados entre chaves e podem ser compostos por múltiplos pares nome:valor, por arrays e também por outros objetos. Desta forma, um objeto JSON pode representar, virtualmente, qualquer tipo de informação.
- Exemplo:



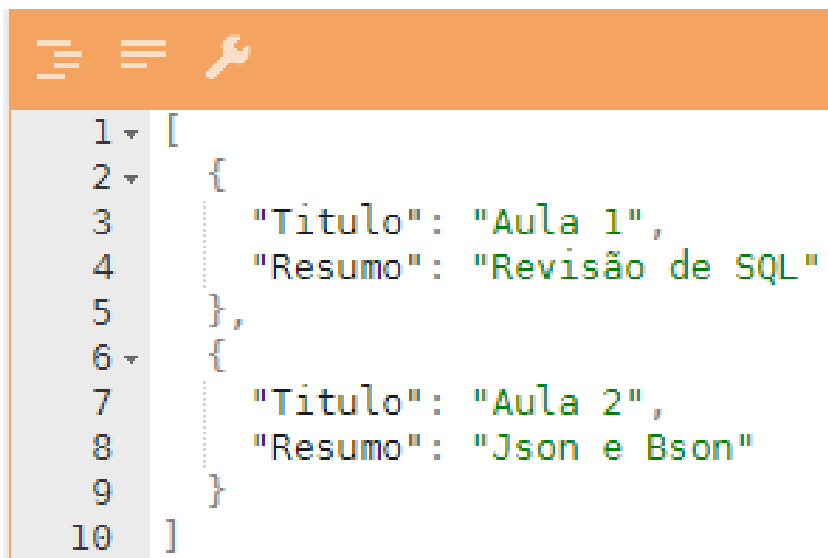
```
1 {  
2   "Titulo": "Json e Bson",  
3   "Resumo": "A volta dos que não foram",  
4   "Ano": 2018,  
5   "Genero": ["Aventura", "Ação", "Ficção"]  
6 }
```



```
1 {  
2   "Titulo": "Json e Bson",  
3   "Resumo": "A volta dos que não foram",  
4   "Ano": 2018,  
5   "Genero": ["Aventura", "Ação", "Ficção"]  
6 }
```

Json - representações

- É possível representar mais de um objeto ou registro de uma só vez com a utilização de Arrays.
- Exemplo:



```
1 [
2   {
3     "Titulo": "Aula 1",
4     "Resumo": "Revisão de SQL"
5   },
6   {
7     "Titulo": "Aula 2",
8     "Resumo": "Json e Bson"
9   }
10 ]
```

- BSON [BSON 2015] É uma representação binária do JSON, utilizada pelo MongoDB para armazenar seus documentos. É utilizado devido a:
 - Rápida escaneabilidade (fast scanability), ou seja, torna possível varrer um documento procurando um valor rapidamente.
 - Novos tipos de dados (Date, ObjectId, Binary Data).
- Todo documento tem um campo chamado `_id` obrigatoriamente. Por padrão, é um campo do tipo ObjectId [MongoDB 2015c], mas pode ser denido como qualquer outro tipo de dado.
- Nasceu com 3 vertentes principais:
 - Leveza
 - Eficiente
 - Atravessável (mover-se facilmente)

Bson

■ Formato BSON

```
{
  dataCriacao: ISODate("2015-07-24T03:00:00Z"), // date (UNIX timestamp)
  RegEx: /[a-z]\w+/i,                          // expressão regular
  numeroGrande: NumberLong(99999999999),        // números de 8 e 4 bytes
  _id: ObjectId("507f191e810c19729de860ea"),
    // identificador único de 12 bytes para documentos
  binario: 01000110100101,                     // dados binários
  codigo: function () {                         // código Javascript
    return;
  }
}
```

▪ BSON - Exemplo de documento

```
{
  _id: ObjectId("55e57a9f97d1ef4073b4b6d1"),
  autor: "Jordan Kobellarz",
  criado: ISODate("2015-01-09T09:01:00Z"),
  titulo: "como usar o Mongo Shell",
  conteudo: "...",
  tags: ["mongo", "nosql", "utfpr"],
  comentarios: [
    {
      data: ISODate("2015-01-09T09:11:00Z"),
      autor: "Fulano da Silva",
      mensagem: "muito bom!"
    },
    {
      data: ISODate("2015-01-09T09:20:00Z"),
      autor: "Sicrano das Dores",
```