

# Manual Técnico

## Formato JSON



Manual Técnico JSON  
**INTRODUÇÃO E DESCRIÇÃO  
DOS SERVIÇOS**



## 1. INTRODUÇÃO



## 2. DESCRIÇÃO DOS SERVIÇOS

2.1. Sintaxe

2.2. Dados e Formatos

Este manual tem por objetivo apresentar o formato utilizado para a estrutura da informação no Portal Oficial desta Entidade. A PROCEDE BAHIA deseja uma excelente leitura e se coloca à disposição para novos esclarecimentos.



**JSON** (JavaScript Object Notation) é um modelo para armazenamento e transmissão de informações no formato texto. Apesar de muito simples, tem sido bastante utilizado por aplicações Web devido a sua capacidade de estruturar informações de uma forma bem mais compacta, tornando mais rápido a análise dessas informações.

Este artigo apresenta o formato utilizado para a estrutura da informação no Portal Oficial dessa entidade e está dividido da seguinte forma. A Seção 2.1 descreve a sintaxe da linguagem. E a Seção 2.2 os Dados e Formatos.

## DESCRIÇÃO DOS SERVIÇOS



A Lei Federal nº 12.527/11 no Art. 8º, §3º, inciso III exige o acesso automatizado por sistemas externos em formatos abertos, estruturados e legíveis por máquina. Para cumprir esta exigência da Lei, foi adotado o JSON (Java Script Object Notation), que é um formato leve para troca de dados, ou seja, através dele é possível importar informações presentes no Portal Oficial desta entidade para outros sistemas.

JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo C++, C#, Java, JavaScript, Perl, Python e muitas outras. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados.

## 2.1 SINTAXE

A ideia utilizada pelo JSON para representar informações é tremendamente simples: para cada valor representado, atribui-se um nome (ou rótulo) que descreve o seu significado. Esta sintaxe é derivada da forma utilizada pelo JavaScript para representar informações. Por exemplo, para representar o ano de 2017, utiliza-se a seguinte sintaxe:

**Listagem 1:** Representando o ano de 2017

```
"ano": 2017
```

Um par nome/valor deve ser representado pelo nome entre aspas duplas, seguido de dois pontos, seguido do valor. Os valores podem possuir apenas **3 tipos básicos**: numérico (inteiro ou real), booleano e string. As Listagens 2, 3, 4 e 5 apresentam exemplos. Observe que os valores do tipo string devem ser representados entre aspas.

**Listagem 2:** Representando um número real

```
"altura": 1.72
```

**Listagem 3:** Representando uma string

```
"site": "www.procedebahia.com.br"
```

**Listagem 4:** Representando um número negativo

```
"temperatura": -2
```

**Listagem 5:** Representando um valor booleano

```
"casado": true
```

A partir dos tipos básicos, é possível construir **tipos complexos**: array e objeto. Os arrays são delimitados por colchetes, com seus elementos separados entre vírgulas. As listagens 6 e 7 mostram exemplos.

**Listagem 6:** Array de Strings

```
["RJ", "SP", "MG", "ES"]
```

**Listagem 7:** Matriz de Inteiros

```
[
  [1,5],
  [-1,9],
  [1000,0]
]
```

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! O exemplo da Listagem 8 mostra a representação dos dados de um filme.

#### Listagem 8: Objeto

```
{
  "titulo": "JSON x XML",
  "resumo": "o duelo de dois modelos de representação de informações",
  "ano": 2017,
  "genero": ["aventura", "ação", "ficção"]
}
```

É possível representar mais de um objeto ou registro de uma só vez. Um exemplo é apresentado na Listagem 9, onde dois filmes são representados em um array.

#### Listagem 9: Array de objetos

```
[
  {
    "titulo": "JSON x XML",
    "resumo": "o duelo de dois modelos de representação de informações",
    "ano": 2017,
    "genero": ["aventura", "ação", "ficção"]
  },
  {
    "titulo": "JSON James",
    "resumo": "a história de uma lenda do velho oeste",
    "ano": 2017,
    "genero": ["western"]
  }
]
```

Por fim, é importante citar que a palavra-chave "null" deve ser utilizada para a representação de valores nulos (Listagem 10).

#### Listagem 10: Representando um valor nulo

```
"site":null
```

## 2.2 DADOS E FORMATOS

VALOR	SÍMBOLO	FORMATO
Numérico	N	Inteiro ou Real
Booleano	B	
String	S	
Nulo	NU	

### RELATÓRIO ESIC

NOME	TIPO DE DADO
Órgão	S
Recebidas	N
Atendidas	N
Indeferidas	N
PF	N
PJ	N
Residente	N
Não Residente	N

### ACESSO DA INFORMAÇÃO

NOME	TIPO DE DADO
Título	S
Texto	S

### LDO – LEIS DE DIRETRIZES ORÇAMENTÁRIAS

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## LOA – LEI DE ORÇAMENTO ANUAL

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## PPA – PLANO PLURIANUAL

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## PRESTAÇÃO DE CONTAS ANUAIS

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## CONTRATOS

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## LICITAÇÕES

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## CÓDIGOS

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## ESTATUTOS

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## LEI ORGÂNICA DO MUNICÍPIO

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## CONVÊNIOS

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## PORTARIAS

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## RESUMO DA EXECUÇÃO ORÇAMENTÁRIA

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S

## RELATÓRIO DE GESTÃO FISCAL

NOME	TIPO DE DADO
Nº	N
Data	S
Objeto	S
Arquivo	S