Introdução as tecnologias JSON e RESTful

Leonardo Ribeiro Machado
Orientado por: Francisco José Silva e Silva

Laboratório de Sistemas Distribuídos Inteligentes (LSDi)
Universidade Federal do Maranhão (UFMA)
http://www.lsdi.ufma.br

Setembro de 2018

Sumário

Sumario

Introdução

- O conceito de serviços em uma aplicação é comum nas aplicações modernas.
- Diferente de componentes tradicionais, serviços têm algumas características únicas que lhes permitem participar como parte de uma arquitetura orientada a serviços.

Sumario

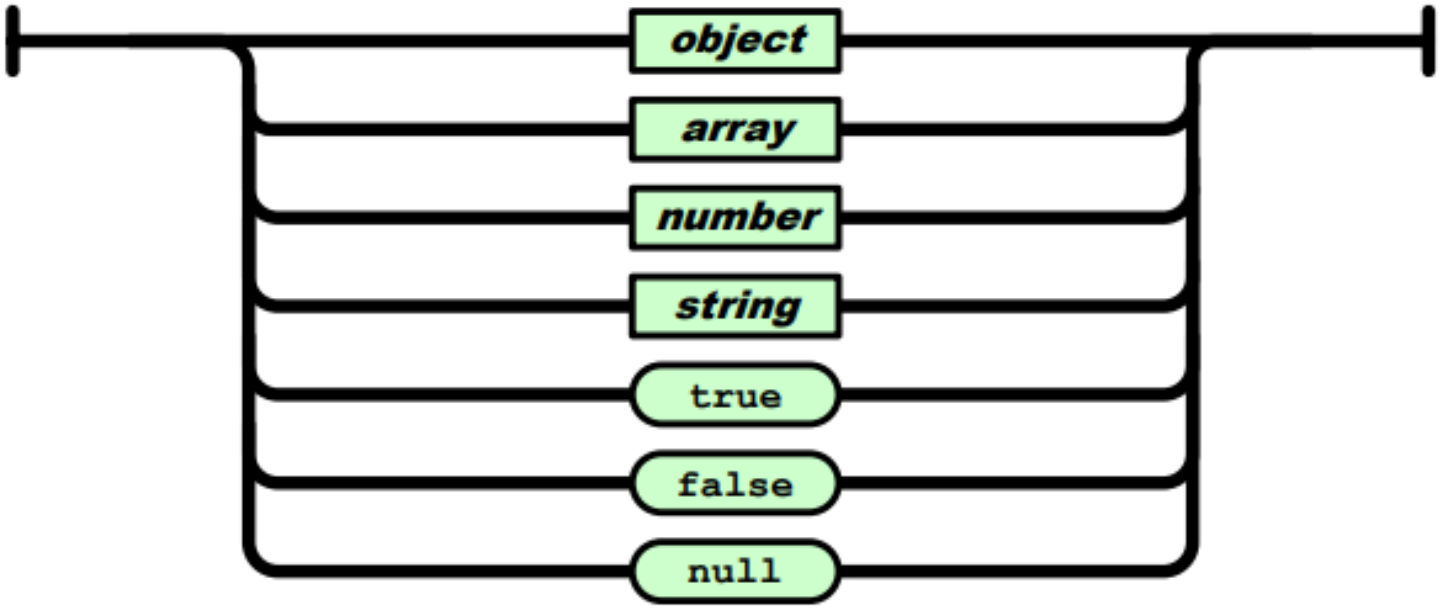
JavaScript Object Notation, JSON [JSON.org, 1999]

- Um formato leve de troca de dados.
- Fácil de ser gerado e analisado por máquinas.
- Baseia-se num subconjunto da Linguagem de Programação JavaScript, Padrão ECMA-262 3ª Edição - Dezembro de 1999.
- O JSON define um pequeno conjunto de regras de estruturação para a representação portátil de dados estruturados [International, 2017].

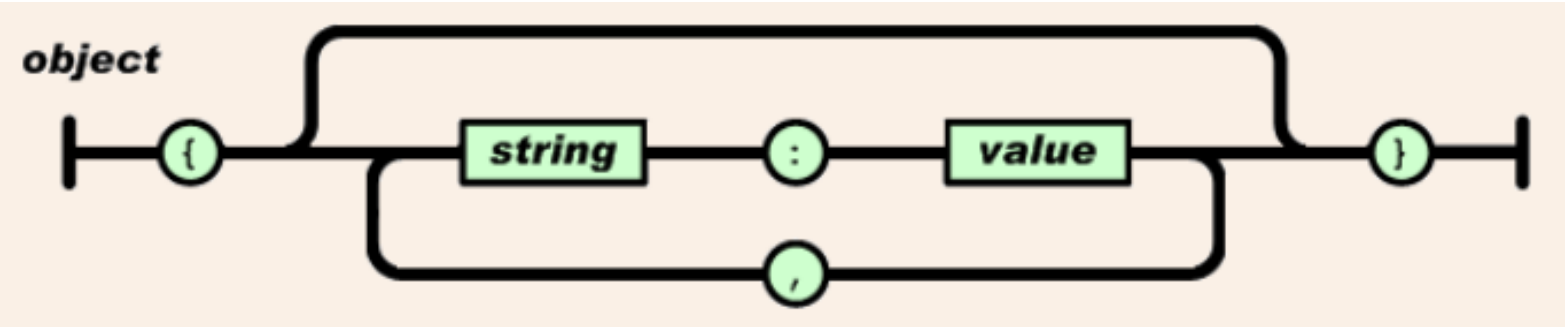
JSON - Estrutura

- Construído em duas estruturas:
  - Uma coleção de pares **nome/valor**.
  - Uma **lista ordenada de valores**.

Valores permitidos



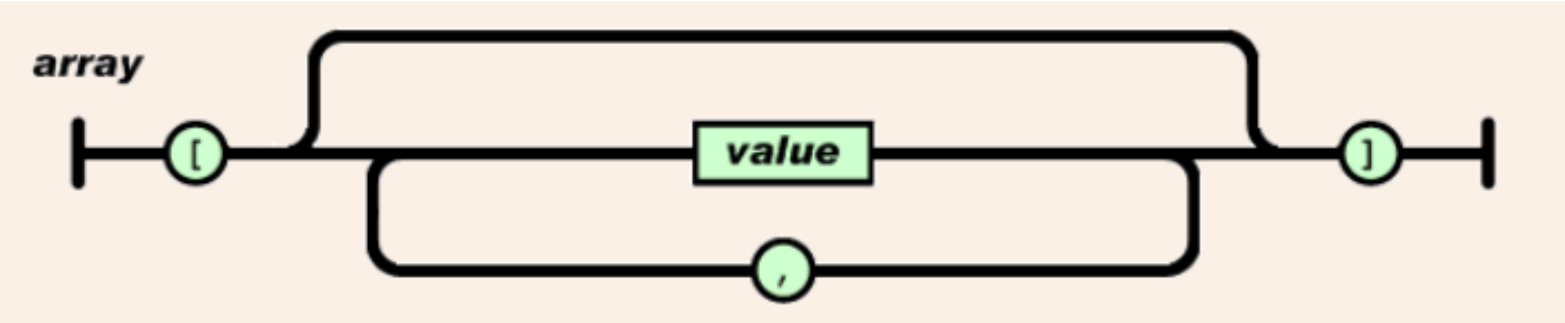
Básica estrutura - nome/valor



Básica estrutura - exemplo


```
{
    "id":37,
    "cpf":"50868667315",
    "senha":"3094c93bb3a98e8c393f1d81c2da903d",
    "pessoa":{
        "id":11025,
        "nome":"JOSE DEDE RIBAMAR CORREA PEREIRA",
        "documentoPessoal":{
            "id":3302,
            "cpf":"50868667315"
        }
    }
}
```


Estrutura de listas ordenada de valores



Estrutura de listas ordenada de valores - exemplo

```
{
    "name":"John",
    "age":30,
    "cars": [
        { "name":"Ford", "models":[ "Fiesta", "Focus", "Mustang" ] },
        { "name":"BMW", "models":[ "320", "X3", "X5" ] },
        { "name":"Fiat", "models":[ "500", "Panda" ] }
    ]
}
```

JSON - Bibliotecas Java

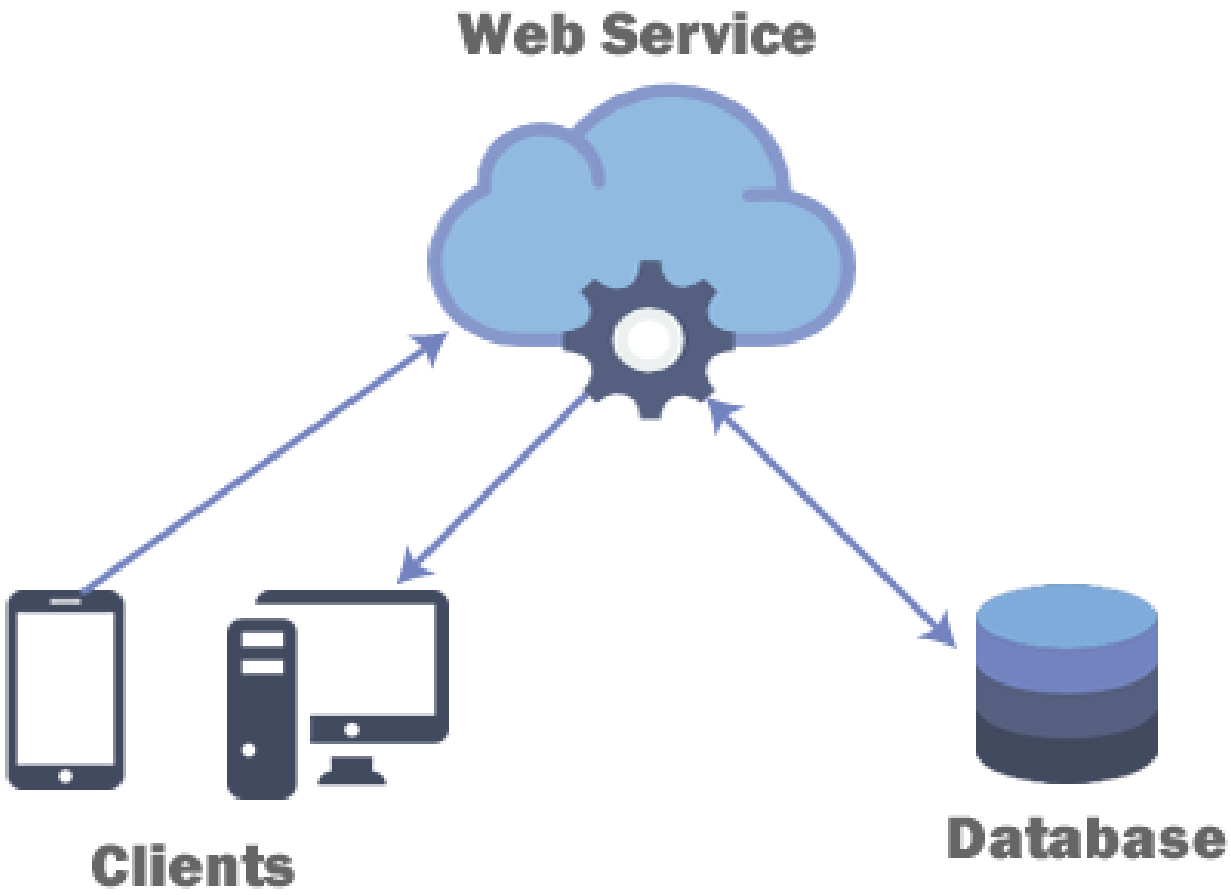
- JSON-java
- google-gson
- JON tools
- Apache johnzon

Sumario

Web Service

- É um sistema de software interoperável projetado para suportar a interação máquina a máquina em uma rede [Note, 2004].
- Os Web Services são considerados aplicações modulares autocontidos e auto descritivos que podem ser publicadas, localizadas e chamadas pela Web [Rao and Su, 2004].

Web Service



Arquitetura do Webservice

- Spring Boot
- MVC
- Java Persistence API

Representational State Transfer, REST

- Estilo arquitetônico para sistemas hipermídia distribuídos.
- O REST não está vinculado a nenhuma tecnologia ou plataforma específica - é simplesmente uma maneira de projetar coisas para funcionar como a Web [Microsoft, 2008].
- O REST usa os métodos básicos de interação remota interna do HTTP (**PUT, POST, GET e DELETE**) aplicando sua semântica pretendida para acessar qualquer recurso referenciável para a URL [Garriga et al., 2016].

RESTful

- Serviços da Web que utilizam o modelo REST.
- Os serviços da Web RESTful são identificados por URIs, que oferecem um espaço de endereçamento global para recursos e serviços [Sheng et al., 2014].

Verbos HTTP

- **GET:**
  - Solicita uma representação específica de um recurso.
  - Solicitações só devem recuperar dados.
- **PUT:**
  - Crie ou atualiza um recurso com a representação fornecida.
- **DELETE:**
  - Exclui o recurso especificado.
- **POST:**
  - Submete dados a serem processados pelo recurso identificado.
- **PATH:**
  - Aplica modificações parciais em um recurso.

Ferramentas

- Postman
- IntelliJ IDEA
- Android Studio
- Retrofit

Principais erros e retornos

Status Range	Description	Examples
100	Informational	100 Continue
200	Successful	200 OK
201	Created	
202	Accepted	
300	Redirection	301 Moved Permanently
304	Not Modified	
400	Client error	401 Unauthorized
402	Payment Required	
404	Not Found	
405	Method Not Allowed	
500	Server error	500 Internal Server Error
501	Not Implemented	

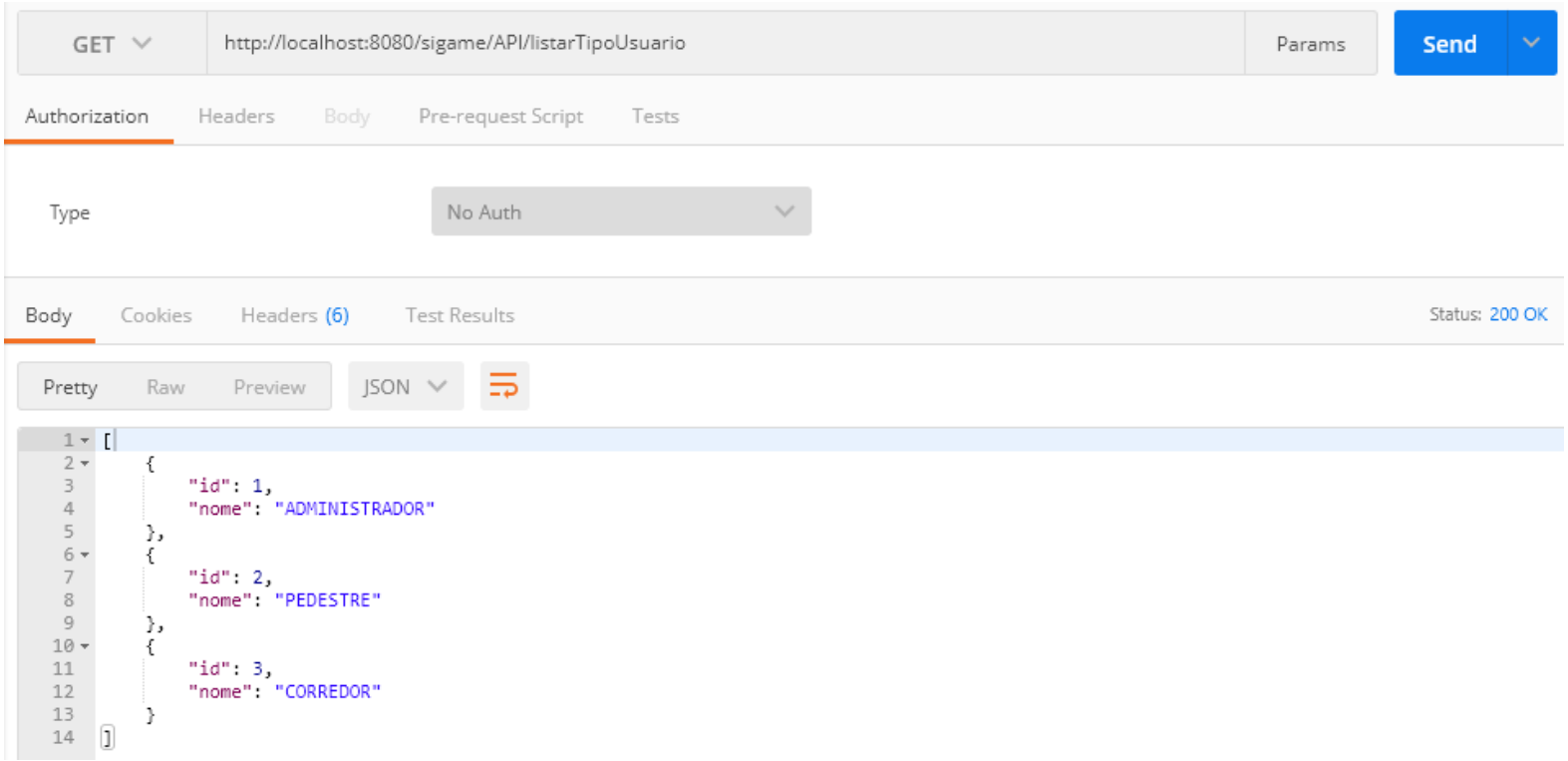
GET - exemplo de código Java

```java
@RequestMapping(value = "/listarUsuario", method = RequestMethod.GET)
public ResponseEntity<List<Usuario>> listarUsuario() {
    List<Usuario> usuarios = new ArrayList<>();
    usuarios = usuarioService.getAll();

    return new ResponseEntity<List<Usuario>>(usuarios, HttpStatus.OK);

}
```

GET - exemplo de requisição



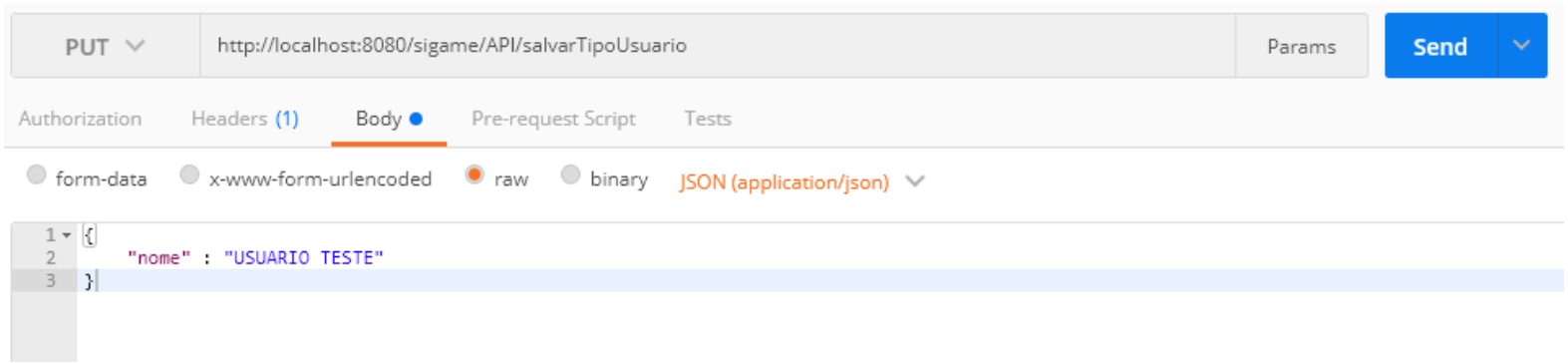
PUT - exemplo de código Java

```
@RequestMapping(value = "/listarUsuario", method = RequestMethod.GET)
public ResponseEntity<List<Usuario>> listarUsuario() {
    List<Usuario> usuarios = new ArrayList<>();
    usuarios = usuarioService.getAll();

    return new ResponseEntity<List<Usuario>>(usuarios, HttpStatus.OK);

}
```

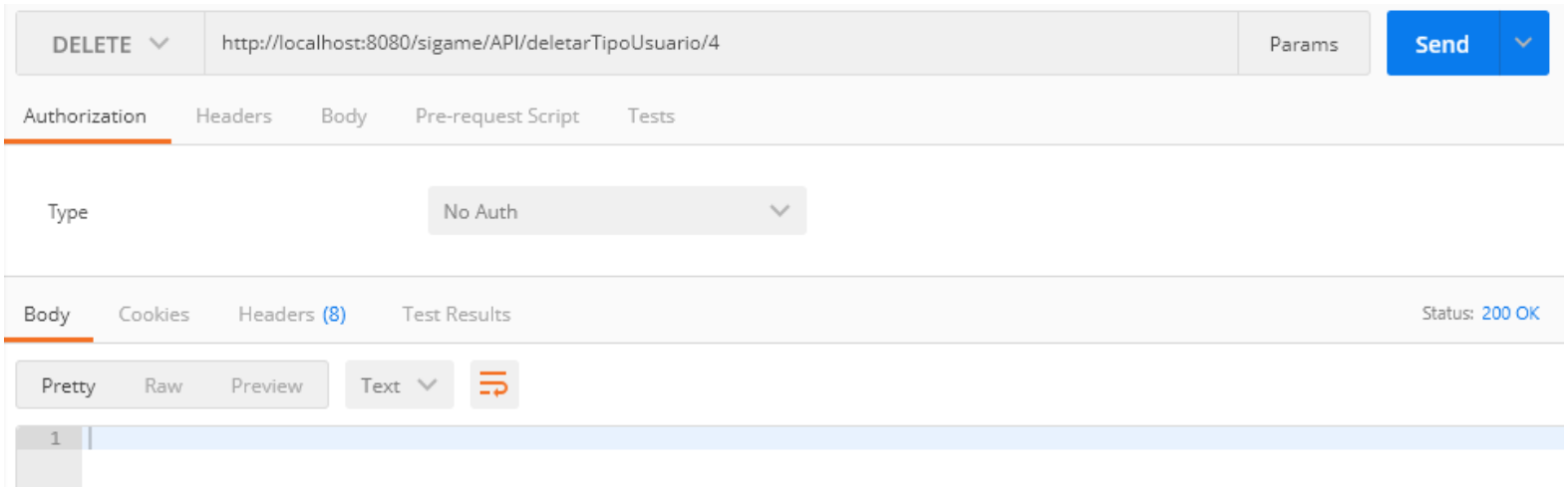
PUT - exemplo de requisição



DELETE - exemplo de código Java

```java
@RequestMapping(value = "/deletarTipoUsuario/{id}", method = RequestMethod.DELETE)
public ResponseEntity deletarTipoUsuario(@PathVariable Long id) {
    tipoUsuarioService.delete(id);
    return new ResponseEntity<>(HttpStatus.OK);
}
```

DELETE - exemplo de requisição




Garriga, M., Mateos, C., Flores, A., Cechich, A., and Zunino, A. (2016).
Restful service composition at a glance: A survey.
Journal of Network and Computer Applications, 60:32–53.


International, E. (2017).
Final draft ecma-404 2nd edition.
http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf.
(Accessed on 08/21/2018).


JSON.org (1999).
Json.
http://www.json.org/.
(Accessed on 08/21/2018).


Microsoft (2008).
A guide to designing and building restful web services with wcf 3.5.

https://msdn.microsoft.com/en-us/library/dd203052.aspx. (Accessed on 08/23/2018).


Note, W. W. G. (2004).
Web services architecture.
https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest.
(Accessed on 08/22/2018).


Rao, J. and Su, X. (2004).
A survey of automated web service composition methods.
In *International Workshop on Semantic Web Services and Web Process Composition*, pages 43–54. Springer.


Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. (2014).
Web services composition: A decade's overview.
Information Sciences, 280:218–238.