Sistemas Distribuídos & Mobile

Professores: Bonato

Assunto: REST E JSON

Conceitos básicos

REST

REST, ou Transferência de Estado Representativo (*REpresentational State Transfer*) é uma maneira simples de realizar interações em sistemas independentes. Sua popularidade vem crescendo desde 2005, o que inspira o design de serviços, como por exemplo a API do Twitter. Isso está relacionado ao fato do REST permitir a interação sem stress com diversos clientes, tanto em dispositivos móveis quanto qualquer outro website/serviço. Na teoria, REST não tem nada a ver com web, mas o conceito é amplamente utilizado na área, e foi inspirado pelo HTTP. Como resultado, onde existe HTTP, o REST pode ser utilizado.

A alternativa é criar convenções relativamente complexas em cima do HTTP. As vezes, isso nos leva a novas linguagens baseadas no XML. O melhor exemplo disso é o SOAP. Você precisa aprender por completo um conjunto de novas convenções, mas você nunca vai utilizar o HTTP com todos os seus recursos. Como o REST foi inspirado pelo HTTP e utiliza sua base, é melhor entendermos primeiro como o HTTP de fato funciona, revendo o material da aula 2 deste curso antes de continuar a leitura.

Verbos HTTP

Cada requisição HTTP está atrelada a um verbo HTTP, ou método, no cabeçalho de requisição. São as letras maiúsculas no inicio do cabeçalho. Por exemplo,

GET / HTTP/1.1
Significa que o método GET está sendo utilizado, enquanto

DELETE /clientes/ana HTTP/1.1
significa que o método DELETE está sendo utilizado.

Os verbos HTTP dizem ao servidor o que ele deve fazer com a informação identificada na URL.

Os verbos HTTP dizem ao servidor o que ele deve fazer com a informação identificada na URL. A requisição pode por opção conter informações adicionais no corpo/body, que podem ser necessárias para executar a operação - por exemplo, informações que você quer guardar com o recurso.

Como neste curso já criamos diversos formulários em HTML, estamos familiarizados com dois dos mais importantes verbos HTTP: GET e POST. Mas, existem muitos outros verbos HTTP disponíveis. Os mais importantes para criar uma API RESTful são GET, POST, PUT, e DELETE. Outros métodos estão disponíveis, como o HEAD e OPTIONS, mas a utilização destes é mais rara (se você quer saber mais sobre métodos HTTP, acesse a RFC 2616 em http://www.ietf.org/rfc/rfc2616.txt).

Os verbos HTTP são associados então com os tradicionais CRUD para a criação da aplicação, conforme abaixo:

- Create (insert) = POST
- Retrieve (select) = GET
- Update = PUT
- Delete = DELETE

Servlet

O modo mais simples de se implementar um serviço REST em Java EE é usando um servlet, já que este possui métodos para todas os verbos HTTP. Leia o material da aula 4 deste curso e depois veja o exemplo abaixo.

JSON

JSON (JavaScript Object Notation - Notação de Objetos JavaScript) é uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar. Está baseado em um subconjunto da linguagem de programação JavaScript, Standard ECMA-262 3a Edição -Dezembro - 1999. 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.

JSON está constituído em duas estruturas:

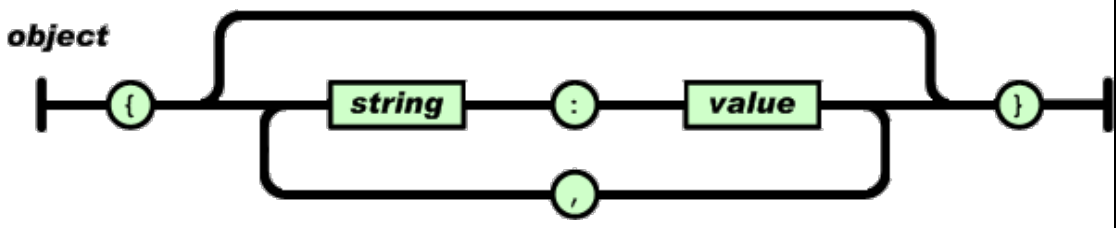
- Uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um object, record, struct, dicionário, hash table, keyed list, ou arrays associativos.

- Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como um array, vetor, lista ou sequência.

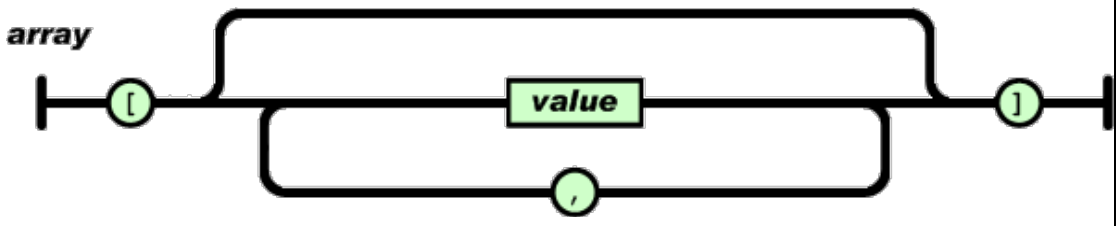
Estas são estruturas de dados universais. Virtualmente todas as linguagens de programação modernas as suportam, de uma forma ou de outra. É aceitável que um formato de troca de dados que seja independente de linguagem de programação se baseie nestas estruturas.

Em JSON, os dados são apresentados desta forma:

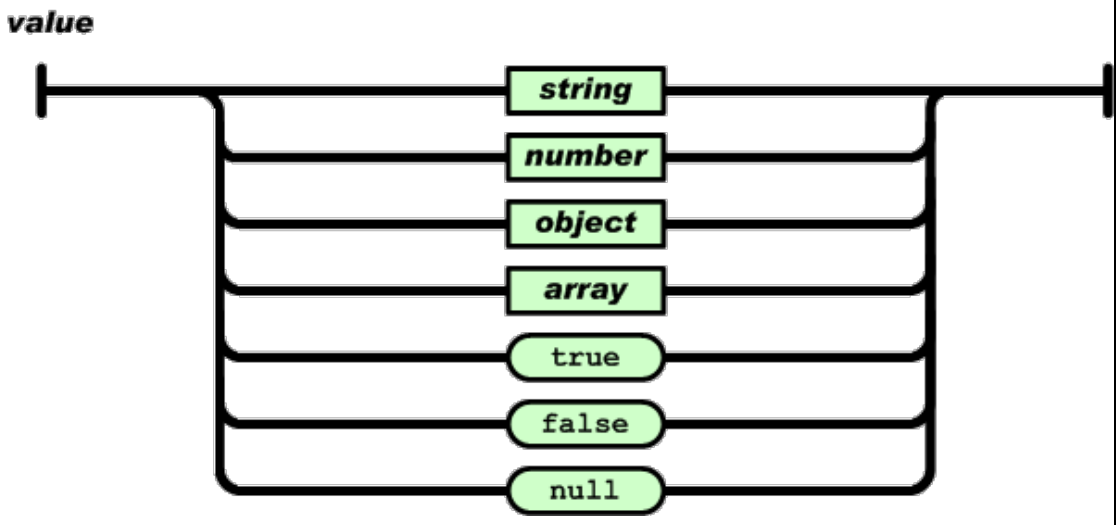
Um objeto é um conjunto desordenado de pares nome/valor. Um objeto começa com { (chave de abertura) e termina com } (chave de fechamento). Cada nome é seguido por : (dois pontos) e os pares nome/valor são seguidos por , (vírgula).



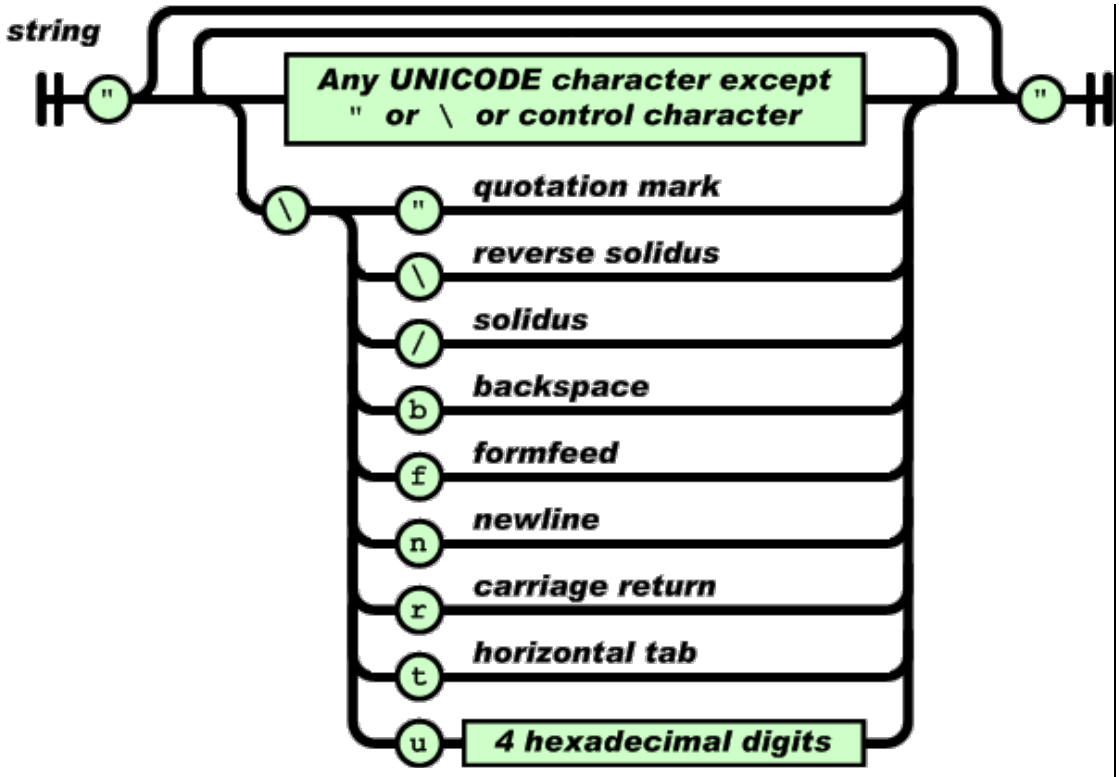
Um array é uma coleção de valores ordenados. O array começa com [(colchete de abertura) e termina com] (colchete de fechamento). Os valores são separados por , (vírgula).



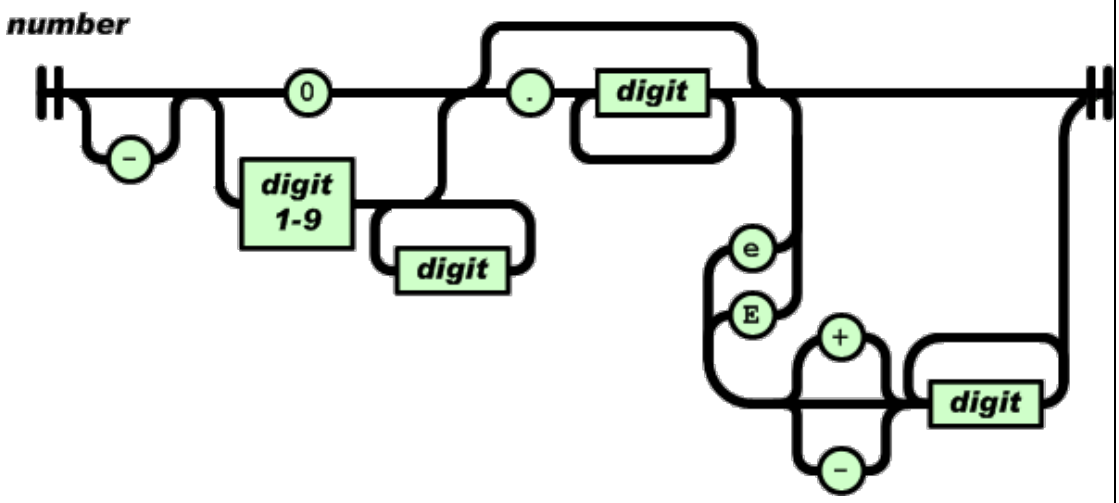
Um valor (value, na imagem acima) pode ser uma cadeia de caracteres (string), ou um número, ou true ou false, ou null, ou um objeto ou um array. Estas estruturas podem estar aninhadas.



Uma string é uma coleção de nenhum ou mais caracteres Unicode, envolvido entre aspas duplas usando barras invertidas como caractere de escape. Um caractere está representando como um simples caractere de string. Uma cadeia de caracteres é parecida com uma cadeia de caracteres em C ou Java.



Um número é similar a um número em C ou Java, exceto quando não se usa os números octais ou hexadecimais.



Espaços em branco podem ser inseridos em qualquer parte dos símbolos.

Exemplo de Arquivo JSon:

```
[
    {
        "fone": "(11) 91234-4330",
        "nome": "Mário de Andrade",
        "id": 10,
        "email": "mda@usjt.br"
    },
    {
        "fone": "(51) 24323-9887",
        "nome": "Mário Quintana",
        "id": 62,
        "email": "mario@poepoa.lit"
    }
]
```

Dica

Para acessar serviços REST use o app Postman, disponível em http://www.getpostman.com/

Bibliografia

Introdução ao REST WS, disponível em http://www.devmedia.com.br/introducao-ao-rest-ws/26964. Acessado em 12/05/2015.

Introdução ao JSON, disponível em http://www.json.org/json-pt.html. Acessado em 12/05/2015.

Como eu expliquei REST para minha esposa, disponível em https://distopico.wordpress.com/traducao-de-how-i-explained-rest-to-my-wife/ . Acessado em 07/06/2017.