


Jakarta EE Web Total


Processamento com a API JSON-P




Tópicos Abordados




- O formato JSON
- JSON-P
 - Object Model API
 - Streaming API



O Formato JSON



- Formato de troca de dados
 - Muito usado em web services e outras aplicações que se comunicam pela internet
- Inspirado no JavaScript
 - JavaScript **O**bject **N**otation
- Baseado em texto
- Possui 2 estruturas de dados
 - Objetos
 - Conjunto de pares de chave e valor
 - Arrays
 - Lista de valores



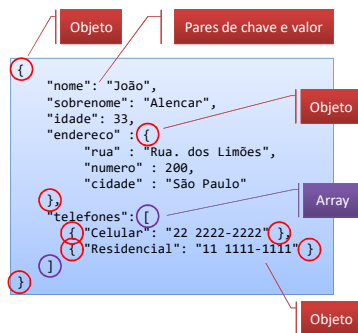
O Formato JSON



- Possui 2 tipos de dados
 - string
 - number
- JSON x XML
 - JSON é mais compacto
 - XML possui recursos para validação de dados (XML Schema e DTD)



Exemplo de Documento JSON



JSON-P



- API do Java EE para processamento de documentos JSON (geração e parsing)
- 2 APIs
 - Object Model API
 - Cria uma árvore do documento inteiro em memória
 - Streaming API
 - Baseado em eventos



JSON-P: Object Model API



- Criar um documento JSON

```
JsonObject rootObj = Json.createObjectBuilder()
    .add("nome", "João Almeida")
    .add("idade", 30)
    .add("endereco", Json.createObjectBuilder()
        .add("rua", "Rua dos Abacates")
        .add("numero", 50).build())
    .build();
```

- Gravar em uma saída

```
StringWriter out = new StringWriter();
try (JsonWriter jsonWriter = Json.createWriter(out)) {
    jsonWriter.writeObject(rootObj);
}
```



JSON-P: Object Model API



- Parse de um documento JSON

```
JsonObject rootObj;
StringReader in = new StringReader(jsonStr);
try (JsonReader jsonReader = Json.createReader(in)) {
    rootObj = jsonReader.readObject();
}

String nome = rootObj.getString("nome");
int idade = rootObj.getInt("idade");
```



JSON-P: Streaming API



- Criar um documento JSON

```
StringWriter out = new StringWriter();
try (JsonGenerator g = Json.createGenerator(out)) {
    g.writeStartObject()
        .write("nome", "Pedro Silva")
        .write("idade", 32)
        .writeStartObject("endereco")
            .write("rua", "Rua do Java")
            .write("numero", 100)
        .writeEnd()
    .writeEnd();
}
```



- Parse de um documento JSON

```
JsonParser parser = Json.createParser(new StringReader(jsonStr));
while (parser.hasNext()) {
    Event event = parser.next();

    switch (event) {
        case KEY_NAME: /* ... */ break;
        case VALUE_STRING: /* ... */ break;
        case VALUE_NUMBER: /* ... */ break;
        case VALUE_TRUE: /* ... */ break;
        case VALUE_FALSE: /* ... */ break;
        case VALUE_NULL: /* ... */ break;
        case START_ARRAY: /* ... */ break;
        case END_ARRAY: /* ... */ break;
        case START_OBJECT: /* ... */ break;
        case END_OBJECT: /* ... */ break;
    }
}
```



- Object Model API

- Vantagens

- Manipulação dos dados é mais fácil e intuitiva

- Desvantagem

- É mais lenta e ocupa mais memória

- Streaming API

- Vantagem

- É mais rápida e não ocupa tanta memória

- Desvantagem

- A manipulação dos dados é pouco intuitiva e mais complexa

