Chapter 7: JSON – JAVA XML AND JSON

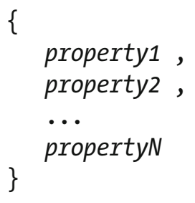
1. What is JSON?

JSON (JavaScript ObjectNotation) is a language-independent data format that expresses JSON objects as human-readable lists of properties (name-value pairs).

Although derived from a nonstrict subset of JavaScript, code to parse JSON objects into equivalent language-dependent objects is available in many programming languages.

1. JSON Syntax Tour

The JSON data format presents a JSON object as a brace-delimited and comma-separated list of properties:



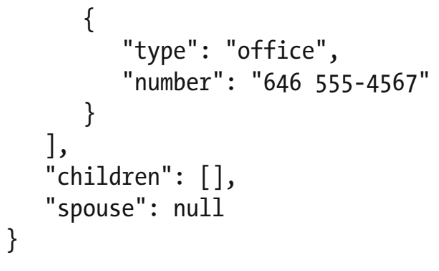
A comma is not placed after the final property. For each property, the name is expressed as a string that’s typically quoted (and by a pair of double quotes). The name string is followed by a coloncharacter, which is followed by a value of a specific type ("name": "JSON", for example)

JSON supports the following six types:

* Number: A signed decimal number that may contain a fractional part and may use exponential (E) notation. JSON doesn’t permit non numbers (such asNaN), nor does it make any distinction between integer andfloating-point. Furthermore, JSON doesn’t recognize the octal and hexadecimal formats. (Although JavaScriptuses adouble precision floating-point formatfor all numeric values, other languages implementing JSONmay encode numbers differently.)
* String: A sequence of zero or more Unicode characters. Strings are delimited with double quotes and support a backslash escapingsyntax.
* Boolean: Either of the values true or false.
* Array: An ordered list of zero or more values, each of which may be of any type. Arrays use square bracket notation with elements being comma-separated
* Object: An unordered collection of properties where the names (also called keys) are strings. Because objects are intended to represent associative arrays, it’s recommended, although not required, that each key be unique within an object. Objects are delimited with braces and use commas to separate each property.Within each property the colon character separates the key from its value
* Null: An empty value

Specify a JSON object:

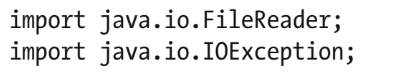


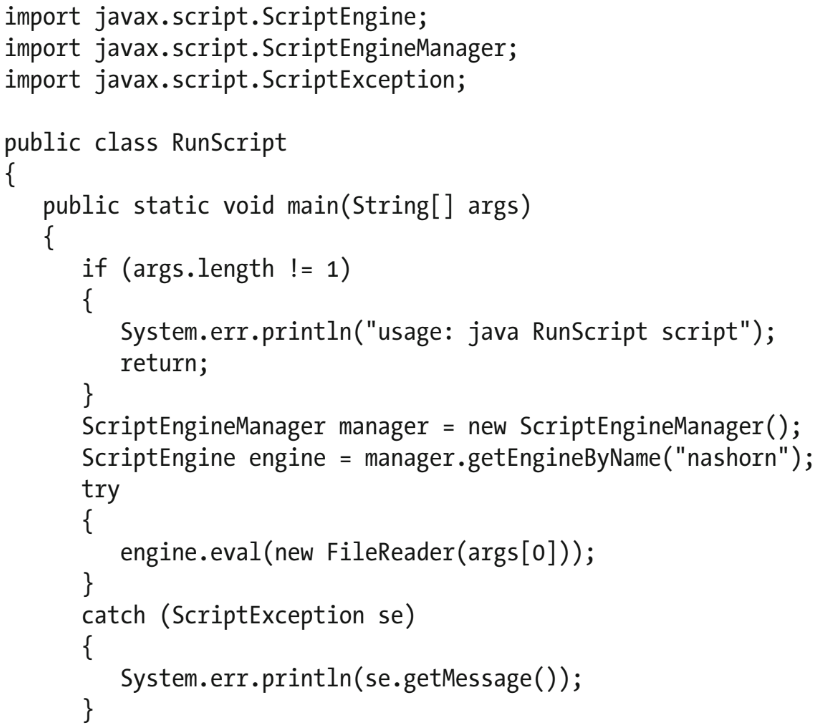


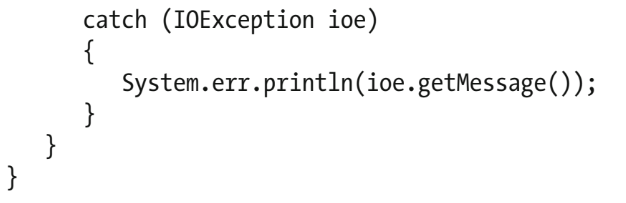
In this example, the object consists of eight properties with the following keys:

* firstNameidentifies a person’s first name and is of type string.
* lastNameidentifies a person’s last name and is of type string.
* isAliveidentifies a person’s alive status and is of type Boolean.
* ageidentifies the age of a person and is of type number.
* addressidentifies a person’s location and is of type object. Within this object are four properties (of typestring): streetAddress,city,state, andpostalCode.
* phoneNumbersidentifies a person’s phone numbers and is of type array. Within the array are two objects; each object consists of type and number properties (of type string).
* childrenidentifies a person’s children (if any) and is of type array.
* spouseidentifies a person’s partner and is empty.

1. Demonstrating JSON with JavaScript







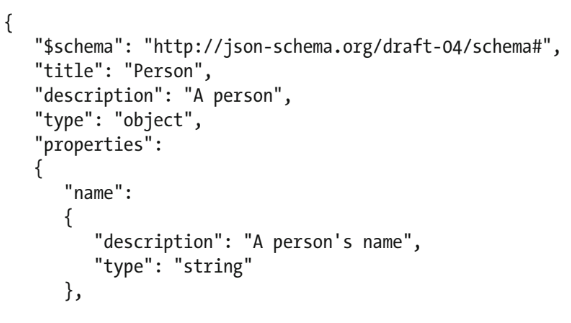
* The ScriptEngineManagerobject ’s ScriptEnginegetEngineByName(String shortName)method is called to obtain a scriptengine corresponding to the desiredshortNamevalue.
* ScriptEngine declares several eval() methods for evaluating a script.main() invokes the Object eval(Reader reader) method to read the script from itsjava.io.FileReaderobject argument and (assuming that java.io.IOException isn’t thrown) then evaluate the scrip

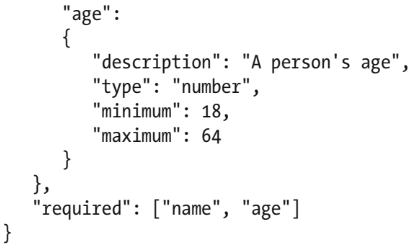
1. Validating JSON Objects

Validation is typically performed in the context of JSON Schema.

JSON Schema is a grammar language for defining the structure, content, and (to some extent) semantics of JSON objects. It lets you specify meta data (data about data) about what an object’s properties mean and what values are valid for those properties

Example of a JSON based schema





* The $schema keyword states that this schema is written according to the draft version 4 specification.
* The title keyword identifies the JSON object being validated by this schema. In this case, a Person object is being validated.
* The description keyword provides a description of the Person object. As with title, description adds noconstraint to the data being validated.
* The type keyword signifies that the containing object isa JSON object (via theobjectvalue). Also, it identifies property types (such as string and number).
* The propertie skeyword introduces an array of the properties that can appear in the JSON object. These properties are identified as name and age. Each property is further described by an object that provides a description keyword to describe the property and a type keyword to identify the type of value that can be assigned to the property. This is a constraint: you must assign a string to name and a number to age. For the age property, minimum and maximum keywords are specified to provide additional constraints: the number assigned to agemust range from 18 through 64.
* The required keyword introduces an array that identifies those properties that must be present in the JSON object. In the example, both name and age are required properties