

Using the JSONGroovyBuilder

The extensions.groovy bundle includes a [Groovy Builder](#) which makes it easy to create JSON documents from Sling scripts. This code is inspired by the builder contained in Andres Almiray's json-lib project (<http://json-lib.sourceforge.net/>). However, Sling's builder is **not** compatible with the json-lib builder.

An instance of this class is available to all Groovy scripts as a variable named `jsonBuilder`.

A basic script looks like this:

```
out.write jsonBuilder.json {
    text currentNode.getProperty("text").string
} as String
```

This can be simplified by using JSONGroovyBuilder's write method:

```
jsonBuilder.write(out).json {
}
```

Examples

These examples are taken from the test case: <http://svn.apache.org/viewvc/sling/trunk/bundles/commons/json/src/test/groovy/org/apache/sling/commons/json/groovy/JSONGroovyBuilderTest.groovy?view=markup>.



The JSON output below is formatted for readability. Actual output will be compact.

```
jsonBuilder.json { }
```

produces...

```
{ }
```

```
jsonBuilder.json([ ])
```

produces...

```
[ ]
```

```
jsonBuilder.json {
    string "json"
    integer 1
    bool true
}
```

produces...

```
{
  'string' : 'json',
  'integer' : 1,
  'bool' : true
}
```

```
jsonBuilder.json {
  first { integer 42 }
  second { integer 48 }
}
```

produces...

```
{
  'first' : {
    'integer' : 42
  },
  'second' : {
    'integer' : 48
  }
}
```

```
jsonBuilder.json {
  books {
    book {
      title "The Definitive Guide to Grails"
      author "Graeme Rocher"
    }
    book {
      title "Groovy in Action"
      author "Dierk Konig"
    }
  }
}
```

produces...

```
{
  'books' : {
    'book' : [
      { 'title' : 'The Definitive Guide to Grails', 'author' : 'Graeme Rocher' },
      { 'title' : 'Groovy in Action', 'author' : 'Dierk Konig' }
    ]
  }
}
```

Creating a Wrapper

The `json` pseudo-method of `JSONGroovyBuilder` produces an instance of `org.apache.sling.commons.json.JSONObject` based on the parameters passed to it. If a different pseudo-method is called, the method name will be used to create a wrapper object. For example:

```
builder.books {  
  book {  
    title "The Definitive Guide to Grails"  
    author "Graeme Rocher"  
  }  
  book {  
    title "Groovy in Action"  
    author "Dierk Konig"  
  }  
}
```

produces...

```
{  
  'books' : {  
    'book' : [  
      { 'title' : 'The Definitive Guide to Grails', 'author' : 'Graeme Rocher' },  
      { 'title' : 'Groovy in Action', 'author' : 'Dierk Konig' }  
    ]  
  }  
}
```