Json example 1

```
{
    " sto re": {
        " boo k": [
           {
                " cat ego ry": " ref ere nce ",
                " aut hor ": " Nigel Rees",
                " tit le": " Sayings of the
Centur y",
                " pri ce": 8.95
           },
           {
                " cat ego ry": " fic tio n",
                " aut hor ": " Evelyn Waugh",
                " tit le": " Sword of Honour ",
                " pri ce": 12.99
           },
           {
                " cat ego ry": " fic tio n",
                " aut hor ": " Herman Melvil le",
                " tit le": "Moby Dick",
                " isb n": " 0-5 53- 213 11- 3",
                " pri ce": 8.99
           },
           {
                " cat ego ry": " fic tio n",
                " aut hor ": "J. R. R. Tolkie n",
                " tit le": "The Lord of the Rings",
                " isb n": " 0-3 95- 193 95- 8",
                " pri ce": 22.99
           }
       ],
        " bic ycl e": {
            " col or": " red ",
```

Json example 1 (cont)

```
>     "price": 19.95
  }
 }
}
```

Extract value from Example 1

- Read the json: String --> JsonPath

```
JsonPath jsonPath = JsonPa th.f ro m(res);
```

- First category

```
jsonPa th.g et Str ing ("st ore.bo ok[ 0].c at -
ego ry");
//refe rence
```

- All category

```
jsonPa th.g et Lis t("s tor e.b ook.ca teg ory ");
//[ref erence, fiction, fiction, fiction]
```

- All author that have price > 10

```
jsonPa th.g et Lis t("s tor e.b ook.fi ndAll
{it.price > 10}.au tho r");
//[Evelyn Waugh, J. R. R. Tolkien]
```

- All author that have category = fiction

```
jsonPa th.g et Lis t("s tor e.b ook.fi ndAll
{it.ca tegory == 'ficti on' }.a uth or");
//[Evelyn Waugh, Herman Melville, J. R. R.
Tolkien]
```

- Count number of author

```
jsonPa th.g et Int ("st ore.bo ok.a ut hor.si -
ze( )");
//4
```

- First author that have price > 10

```
jsonPa th.g et Str ing ("st ore.bo ok.find
{it.price > 10}.au tho r");
//Evelyn Waugh
```

- Color of bicycle

```
jsonPa th.g et Str ing ("st ore.bi cyc le.c ol -
or");
//red
```

Convert to Array or List of Object

```
- Response
[
   {
        " id": " 11",
        " nam e": " abc "
   }
]
- POJO
@Data
public class Root {
        private String id;
        private String name;
}
- Convert to object
Root root = given( ....e xtr act ().a s( Roo -
t[].cl ass )[0];
or
Root root = given( ....e xtr act ().j so nPa -
th( ).g etL ist ("", Root.c las s).g et(0);
```

Json Example 2 - extract

```
{
    " foo.ba r.b az": {
        " 0.2.0": " tes t"
   }
}
JsonPa th.f ro m(j son ).g etS tri ng( " 'fo o.b -
ar.b az '.' 0.2.0' ")
//test
{
    " a-b ": " min us",
    " a.b ": " dot ",
    " a.b -c": " bot h"
}
JsonPa th.f ro m(j son ).g etS tri ng( " 'a.b -
c '");
//both
```

Json Example 2 - extract (cont)

```
> {
  "map": {
    "true": 12.3,
    "false": 15.0
  }
}
JsonPath.from(json).getFloat("map.'false'")
//15.0f
JsonPath.from(json).getFloat("map.'true'")
//12.3f
```

Json Example 3 - extract

```
{
    " sem est er": "Fall 2015",
    " gro ups ": [
       {
            " sit eUr l": " htt p:/ /cp hbu sin -
ess jb.c lo uda pp.n et /CA 2/",
            " err or": "NO AUTHOR /CL ASS -IN FO"
       },
       {
            " sit eUr l": " htt p:/ /ca 2-e bsk -
i.r hcl oud.co m/C A2N ew/ ",
            " aut hor s": " Ebbe, Kasper, Christ -
off er",
            " cla ss": "A klasse n",
            " gro up": " Gruppe: Johns Llama
Herders A/S"
       },
       {
            " sit eUr l": " htt p:/ /ca 2-c hri -
sli nd.r hc lou d.c om/ CA2 Fin al/ ",
            " err or": "NO AUTHOR /CL ASS -IN FO"
       },
       {
            " sit eUr l": " htt p:/ /ca 2-p ern -
ill e.r hcl oud.co m/N YCA 2/",
            " aut hor s": " Marta, Jeanette,
Pernil le",
            " cla ss": "DAT A",
```

Json Example 3 - extract (cont)

```
>      "group": "Group: MJP"
   },
   {
     "siteUrl": "https://ca2-afn.rhcloud.com:8443/company.jsp",
     "error": "NO AUTHOR/CLASS-INFO"
   },
   {
     "siteUrl": "http://ca-smcphbusiness.rhcloud.com/ca2/index.jsp",
     "authors": "Mikkel, Steffen, B Andersen",
     "class": "A Class Computer Science",
     "group": "1"
   }
 ]
}
JsonPath.from(json).getList("groups.getAt('class')");
//[null, A klassen, null, DAT A, null, A Class Computer Science]
```

- Additional: how to move null from a list

```
list.removeAll(Collections.singleton(null));
System.out.println("list = " + list);
//[A klassen, DAT A, A Class Computer Science]
```

Special cases

```
//Multiple nested arrays
JsonPa th.f ro m(r esp onse)
.get("c omm uni cat ion Pol icy.ru les.fl atten()
   .fi ndAll {it.test1 == 'TRANS ACT IONAL'}
   .va lue.fl att en( ).f indAll {it.ch annel ==
'EMAIL '}.v al ue");
System.ou t.p rin tln (va lue);
//Check contains
" res pon se.d at a.t ask s.f indAll{
        it.t ri gge red _by.co nta ins (re stA -
ssu red Jso nRo otO bje ct.r es pon se.d at a.t -
ask s.find{
        it.n am e.e qua ls( 'In ves tig ate Sug -
ges tio ns' )}.id) }.name "
restAs sur edJ son Roo tObject --> refer to root of
json
```