

2016 json-schema json-ld primers

```
1 // primer-json-schema.js
2
3 // FISHC01LBC
4 // TCAACYAATCAYAAAGATATYGGCAC
5 //
6 // FISHC01HBC
7 // ACTTCYGGGTGCCRAARAATCA
8
9 {
10   "id": 1,
11   "name": "FISHC01LBC",
12   "description": "Fish C01 barcoding primer",
13   "sequence": "TCAACYAATCAYAAAGATATYGGCAC",
14   "orientation": [
15     "forward",
16     "reverse"
17   ]
18 }
19
20 {
21   "type": "object",
22   "properties": {
23     "id": {
24       "type": "string",
25       "description": "unique id",
26       "minLength": 0,
27       "maxLength": 255
28     },
29     "name": {
30       "type": "string",
31       "description": "short name",
32       "minLength": 4,
33       "maxLength": 32
34     },
35     "description": {
36       "type": "string"
37     },
38     "sequence": {
39       "type": "string",
40       "description": "sequence"
41     },
42     "orientation": {
43       "type": "string",
44       "default": "forward",
45       "enum": [
46         "forward",
47         "reverse"
48       ]
49     }
50   },
51   "required": [
52     "id",
53     "name",
54     "sequence",
55   ]
56 }
```



```

57         "orientation"
58     ]
59 }
60
61
62 {
63     "$id": "http://example.com/example.json",
64     "type": "object",
65     "definitions": {},
66     "$schema": "http://json-schema.org/draft-06/schema#",
67     "properties": {
68         "id": {
69             "$id": "/properties/id",
70             "type": "integer",
71             "title": "id",
72             "description": "id string",
73             "default": ""
74         },
75         "name": {
76             "$id": "/properties/name",
77             "type": "string",
78             "title": "The Name Schema",
79             "description": "An explanation about the purpose of this instance.",
80             "default": "",
81             "examples": [
82                 "FISHC01LBC"
83             ]
84         },
85         "description": {
86             "$id": "/properties/description",
87             "type": "string",
88             "title": "The Description Schema",
89             "description": "An explanation about the purpose of this instance.",
90             "default": "",
91             "examples": [
92                 "Fish C01 barcoding primer"
93             ]
94         },
95         "sequence": {
96             "$id": "/properties/sequence",
97             "type": "string",
98             "title": "The Sequene Schema",
99             "description": "An explanation about the purpose of this instance.",
100             "default": "",
101             "examples": [
102                 "TCAACYAATCAYAAAGATATYGGCAC"
103             ]
104         },
105         "orientation": {
106             "$id": "/properties/orientation",
107             "type": "array",
108             "uniqueItems": false,
109             "items": {
110                 "$id": "/properties/orientation/items",
111                 "type": "string",
112                 "title": "orientation",
113                 "description": "forward or reverse strand",
114                 "default": "forward",
115                 "examples": [
116                     "forward"
117                 ]

```

```
118     "enum": [
119         "forward, reverse"
120     ]
121 }
122 }
123 },
124 "required": [
125     "id",
126     "name",
127     "sequence"
128 ]
129 }
130
131 const mobx = require("mobx")
132 const { types } = require('mobx-state-tree');
133 const jsonSchemaToTypes = require('jsonschema-to-mobx-state-tree')(types);
134
135 const eventSchema = {
136     type: "object",
137     title: "Event",
138     properties: {
139         title: {
140             type: "string"
141         },
142         public: {
143             type: "boolean",
144             default: false
145         },
146         time: {
147             type: "object",
148             properties: {
149                 start: {
150                     type: "string",
151                     format: "datetime"
152                 },
153                 end: {
154                     type: "string",
155                     format: "datetime"
156                 }
157             },
158             required: ["start"]
159         }
160     },
161     required: ["title", "public"]
162 };
163
164 const primer = {
165     "type": "object",
166     "properties": {
167         "id": {
168             "type": "string",
169             "description": "unique id",
170         },
171         "name": {
172             "type": "string",
173             "description": "short name",
174             "minLength": 4,
175             "maxLength": 32
176         },
177         "description": {
178             "type": "string"
```

```

179     },
180     "sequence": {
181       "type": "string",
182       "description": "sequence"
183     },
184     "orientation": {
185       "type": "string",
186       "default": "forward",
187       "enum": [
188         [
189           "forward",
190           "reverse"
191         ]
192       ]
193     }
194   },
195   "required": [
196     "id",
197     "name",
198     "sequence",
199     "orientation"
200   ]
201 }
202
203 const eventModel = jsonSchemaToTypes(primer);
204 console.log(eventModel);
205
206 // https://npm.runkit.com/jsonschema-to-mobx-state-tree

```

json-ld_simple_protocol_flow.json

```

1 <script type="application/ld+json">
2 [{
3   "@context":
4   {
5     "schema": "http://schema.org",
6     "name": {
7       "@id": "http://schema.org/name",
8       "@type": "@id"
9     },
10    "input": {
11      "@id": "http://schema.org/url",
12      "@type": "@id"
13    },
14    "content": {
15      "@id": "http://schema.org/code",
16      "@type": "@id"
17    }
18  },
19  "name": "calculate transformation efficiency",
20  "input": "http://fab.bio/100ideas/pcr/1/1sjf2h2",
21  "content": "content",
22  "dateCreated": "2013-02-14T13:15:03-08:00"
23 }]
24 </script>
25
26
27 /*#####*/

```

```

28
29
30 <script type="application/ld+json">
31 {
32   "@context": "http://schema.org",
33   "@type": "Recipe",
34   "name": "calculate transformation efficiency",
35   "url": "https://www.example-petstore.com/",
36   "dateCreated": "2013-02-14T13:15:03-08:00"
37 }
38 </script>
39
40
41 /*#####*/
42
43 <script type="application/ld+json">
44 [{
45   "@context":
46   {
47     "name": {
48       "@id": "http://schema.org/name",
49       "@type": "@id"
50     },
51     "operation": "http://schema.org/Recipe",
52     "input": {
53       "@id": "http://schema.org/url",
54       "@type": "@id"
55     },
56     "content": {
57       "@id": "http://schema.org/code",
58       "@type": "@id"
59     }
60   },
61   "@type": "operation",
62   "name": "calculate transformation efficiency",
63   "input": "http://fab.bio/100ideas/pcr/1/1sjf2h2",
64   "content": "content",
65   "dateCreated": "2013-02-14T13:15:03-08:00"
66 }]
67 </script>
68
69
70 /*#####*/
71
72
73 {
74   "@context": {
75     "name": "http://fab.bio/schema/0.1/name",
76     "homepage": {
77       "@id": "http://xmlns.com/foaf/0.1/workplaceHomepage",
78       "@type": "@id"
79     },
80     "Person": "http://xmlns.com/foaf/0.1/Person"
81   },
82   "@id": "http://me.example.com",
83   "@type": "Person",
84   "name": "John Smith",
85   "homepage": "http://www.example.com/"
86 }
87
88 /*#####*/

```