StylesObject for Dynamic Maps

anastasia-ms

# Schema reference guide for the StylesObject in Dynamic Maps

This article is a reference guide to the JSON schema and syntax for the StylesObject. The StylesObject is a StyleObject array representing stateset styles.

## StyleObject

A StyleObject is expressed either as a [BooleanTypeStyleRule](#booleantypestylerule) or a [NumericTypeStyleRule](#numerictypestylerule).

A BooleanTypeStyleRule defines a boolean *state* and associated colors for true and false values. A NumericTypeStyleRule defines a numeric *state* and associated colors for numeric ranges.

The JSON below shows a BooleanTypeStyleRule named occupied and a NumericTypeStyleRule named temperature.

"styles": [  
 {  
 "keyname": "occupied",  
 "type": "boolean",  
 "rules": [  
 {  
 "true": "#FF0000",  
 "false": "#00FF00"  
 }  
 ]  
 },  
 {  
 "keyname": "temperature",  
 "type": "number",  
 "rules": [  
 {  
 "range": {  
 "minimum": 50,  
 "exclusiveMaximum": 70  
 },  
 "color": "#343deb"  
 },  
 {  
 "range": {  
 "maximum": 70,  
 "exclusiveMinumum": 30  
 },  
 "color": "#eba834"  
 }  
 ]  
 }  
]

## NumericTypeStyleRule

A NumericTypeStyleRule is a type of [StyleObject](#styleobject) that defines a numeric *state* and associated colors for numeric ranges.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Description | Required |
| keyName | string | The *state* or dynamic property name. A keyName should be unique inside StyleObject array. | Yes |
| type | string | Value is “numeric”. | Yes |
| rules | [NumberRuleObject](#numberruleobject)[] | Any number of numeric style ranges with associated colors. | Yes |

### Example of NumericTypeStyleRule

The following JSON illustrates a NumericTypeStyleRule *state* named temperature. In this example, the [NumberRuleObject](#numberruleobject) contains two defined temperature ranges and their associated color styles. If the temperature range is 50-69, the display should use the color #343deb. If the temperature range is 31-70, the display should use the color #eba834.

{  
 "keyname": "temperature",  
 "type": "number",  
 "rules": [  
 {  
 "range": {  
 "minimum": 50,  
 "exclusiveMaximum": 70  
 },  
 "color": "#343deb"  
 },  
 {  
 "range": {  
 "maximum": 70,  
 "exclusiveMinumum": 30  
 },  
 "color": "#eba834"  
 }  
 ]  
}

## NumberRuleObject

A NumberRuleObject consists of a [RangeObject](#rangeobject) and a color attribute. If the *state* value falls into the range, its color for display will be the color specified in the color attribute.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Description | Required |
| range | [RangeObject](#rangeobject) | The [RangeObject](#rangeobject) defines a set of logical range conditions which, if true, change the display color of the *state* to the color specified in the color attribute | Yes |

| color | string | The color to use when state value falls into the range. The color attribute is a JSON string in any one of following formats:

HTML-style hex values

RGB (“#ff0”, “#ffff00”, “rgb(255, 255, 0)”)

RGBA (“rgba(255, 255, 0, 1)”)

HSL(“hsl(100, 50%, 50%)”)

HSLA(“hsla(100, 50%, 50%, 1)”)

Predefined HTML colors names, like yellow and blue.

| Yes |

### Example of NumberRuleObject

The following JSON shows a RangeObject. In order for the *state* to fall within the range, its value must be inclusively within a range of 50-69. If it falls within the range, the *color* for display will be #343deb.

{  
 "range": {  
 "minimum": 50,  
 "exclusiveMaximum": 70  
 },  
 "color": "#343deb"  
}

## RangeObject

The RangeObject defines a numeric range value of a [NumberRuleObject](#numberruleobject). For the *state* value to fall into the range, all defined conditions must hold true. For an example, see [NumberRuleObject](#numberruleobject).

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Description | Required |
| minimum | double | All the number x that x ≥ minimum. | No |
| maximum | double | All the number x that x ≤ maximum. | No |
| exclusiveMinumum | double | All the number x that x > exclusiveMinumum. | No |
| exclusiveMaximum | double | All the number x that x < exclusiveMaximum. | No |

### Example of RangeObject

The following JSON shows a RangeObject. In order for the *state* to fall within the range, its value must be inclusively within a range of 50-69.

{  
 "range": {  
 "minimum": 50,  
 "exclusiveMaximum": 70  
 }  
}

## BooleanTypeStyleRule

A BooleanTypeStyleRule defines a boolean *state* and associated colors for true and false values.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Description | Required |
| keyName | string | The *state* or dynamic property name. A keyName should be unique inside style array. | Yes |
| type | string | Value is “boolean”. | Yes |
| rules | [BooleanRuleObject](#booleanruleobject)[1] | A boolean pair with colors for true and false *state* values. | Yes |

### Example of BooleanTypeStyleRule

The following JSON illustrates a BooleanTypeStyleRule *state* named occupied. The [BooleanRuleObject](#booleanruleobject) defines colors for true and false values.

{  
 "keyname": "occupied",  
 "type": "boolean",  
 "rules": [  
 {  
 "true": "#FF0000",  
 "false": "#00FF00"  
 }  
 ]  
}

## BooleanRuleObject

A BooleanRuleObject defines colors for true and false values..

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Description | Required |

| true | string | The color to use when the *state* value is true. The color to use when state value falls into the range. The color attribute is a JSON string in any one of following formats:

HTML-style hex values

RGB (“#ff0”, “#ffff00”, “rgb(255, 255, 0)”)

RGBA (“rgba(255, 255, 0, 1)”)

HSL(“hsl(100, 50%, 50%)”)

HSLA(“hsla(100, 50%, 50%, 1)”)

Predefined HTML colors names, like yellow and blue.

Yes |  
false | string | The color to use when the *state* value is false. | Yes |

### Example of BooleanRuleObject

{  
 "true": "#FF0000",  
 "false": "#00FF00"  
}

## Next Steps