



PathStyleExtension

The `PathStyleExtension` adds selected features to the `PathLayer` and composite layers that render the `PathLayer`, e.g. `PolygonLayer` and `GeoJsonLayer`.

Note: In v8.0, the `getDashArray` and `dashJustified` props are removed from the `PathLayer` and moved into this extension.

CSS

JS

Result

EDIT ON
CODEPEN

Resources

1x

0.5x

0.25x

Rerun

```
import {PolygonLayer} from '@deck.gl/layers';
import {PathStyleExtension} from '@deck.gl/extensions';

const layer = new PolygonLayer({
  id: 'polygon-layer',
  data,
  ...
  getDashArray: [3, 2],
  dashJustified: true,
  dashGapPickable: true,
  extensions: [new PathStyleExtension({dash: true})]
});
```

Installation

To install the dependencies from NPM:

```
npm install deck.gl
# or
npm install @deck.gl/core @deck.gl/layers @deck.gl/extensions
```

```
import {PathStyleExtension} from '@deck.gl/extensions';
new PathStyleExtension({});
```

To use pre-bundled scripts:

```
<script src="https://unpkg.com/deck.gl@8.0.0/dist.min.js"></script>
<!-- or -->
<script src="https://unpkg.com/@deck.gl/core@8.0.0/dist.min.js"></script>
<script src="https://unpkg.com/@deck.gl/layers@8.0.0/dist.min.js"></script>
<script src="https://unpkg.com/@deck.gl/extensions@8.0.0/dist.min.js"></script>
```

```
new deck.PathStyleExtension({});
```

Constructor

```
new PathStyleExtension({dash});
```

- `dash` (Boolean) - add capability to render dashed lines. Default `false`.
- `highPrecisionDash` (Boolean) - improve dash rendering quality in certain circumstances. Note that this option introduces additional performance overhead, see "Remarks" below. Default `false`.
- `offset` (Boolean) - add capability to offset lines. Default `false`.

Layer Properties

When added to a layer via the `extensions` prop, the `PathStyleExtension` adds the following

properties to the layer:

`getDashArray` (**Function|Array**)

Must be specified if the `dash` option is enabled.

The dash array to draw each path with: `[dashSize, gapSize]` relative to the width of the path.

- If an array is provided, it is used as the dash array for all paths.
- If a function is provided, it is called on each path to retrieve its dash array. Return `[0, 0]` to draw the path in solid line.
- If this accessor is not specified, all paths are drawn as solid lines.

`dashJustified` (**Boolean, optional**)

- Default: `false`

Only effective if `getDashArray` is specified. If `true`, adjust gaps for the dashes to align at both ends. Overrides the effect of `highPrecisionDash`.

`getOffset` (**Function|Number**)

Must be specified if the `offset` option is enabled.

The offset to draw each path with, relative to the width of the path. Negative offset is to the left hand side, and positive offset is to the right hand side. `0` extrudes the path so that it is centered at the specified coordinates.

- If a number is provided, it is used as the offset for all paths.
- If a function is provided, it is called on each path to retrieve its offset.

`dashGapPickable` (**Boolean, optional**)

- Default `false`

Only effective if `getDashArray` is specified. If `true`, gaps between solid strokes are pickable. If `false`, only the solid strokes are pickable.

Remarks

Limitations

WebGL has guaranteed support for up to 16 attributes per shader. The current implementation of `PathLayer` uses 13 attributes. Each one of the options of this extension adds one more attribute. In other words, if all options are enabled, the layer will not be able to use other extensions.

Tips on Rendering Dash Lines

There are three modes to render dash lines with this extension:

1. Default: dash starts from the beginning of each line segment
2. Justified: dash is stretched to center on each line segment
3. High precision: dash is evaluated continuously from the beginning of a path

	<code>highPrecisionDash</code> <code>dashJustified</code>	<code>highPrecisionDash</code> <code>dashJustified</code>	<code>highPrecisionDash</code> <code>dashJustified</code>
Relatively long, disjoint lines			
Polyline with sharp corners			
Smooth curves with tiny segments			

The above table illustrates the visual behavior of the three modes.


The default mode works best if the data consists of long, disjoint paths. It renders dashes at exactly the defined lengths.

The justified mode is guaranteed to render sharp, well-defined corners. This is great for rendering polyline shapes. However, the gap size may look inconsistent across line segments due to stretching.

The high precision mode pre-calculates path length on the CPU, so it may be slower and use more resources for large datasets. When a path contains a lot of short segments, this mode yields the best result.

Source

[modules/extensions/src/path-style](#)

 [Edit this page](#)