Highcharts React Integration Migration Guide

A guide for migrating from `highcharts-react-official` v3.x to `@highcharts/react`

Overview

This migration guide covers the transition from the `highcharts-react-official` NPM package (v3.x) to the new official `@highcharts/react` package. The migration process ensures access to the latest features, improvements, and official support while maintaining existing chart functionality.

What's Covered

- Step-by-step migration process
- Breaking changes and how to handle them
- Code examples for common scenarios
- Advanced feature migration strategies
- Troubleshooting common issues

Installation Instructions

Follow these steps to replace `highcharts-react-official` with `@highcharts/react` in your project.

Step 1: Remove the Old Package

```
npm uninstall highcharts-react-official
```

Step 2: Install the New Package

```
npm install @highcharts/react
```

Step 3: Update Import Statements

Replace all import statements in your React components:

```
// Before (highcharts-react-official)
import Highcharts from 'highcharts';
import HighchartsReact from 'highcharts-react-official';

// After (@highcharts/react) - Highcharts import no longer required in most cases
import HighchartsReact from '@highcharts/react';
```

Step 4: Verify Core Dependencies

Ensure you have compatible versions of required dependencies:

```
npm install highcharts@^12.2.0 react@^18.0.0
```

Step 5: Clean Build Cache (Recommended)

Clear your build cache to avoid module resolution issues:

For Standard React Projects

```
# Clear npm cache and reinstall node_modules
npm cache clean --force
rm -rf node_modules
npm install
```

For Vite Projects

```
# Clear npm cache, remove caches, and reinstall
npm cache clean --force
rm -rf node_modules .vite
npm install
```

For Next.js Projects

```
# Clear npm cache, remove caches, and reinstall
npm cache clean --force
rm -rf node_modules .next
npm install
```

Version Requirements

The `@highcharts/react` package requires **Highcharts v12.2 or newer**. Ensure your Highcharts version meets this requirement before migration.

Minimum Requirements

- **Highcharts**: v12.2+
- **React**: v18.0.0+
- **Node.js**: v14.0.0+

Verification

Check your current Highcharts version:

```
npm list highcharts
```

If you need to upgrade Highcharts:

```
npm install highcharts@latest
```

Breaking Changes

The migration from `highcharts-react-official` to `@highcharts/react` includes several important changes. Review each change and apply the necessary updates to your code.

Import Path Changes

Change Required: Update all import statements

```
// Before
import HighchartsReact from 'highcharts-react-official';

// After - Simplified import, no Highcharts dependency needed
import HighchartsReact from '@highcharts/react';
```

Component Props Interface

Change: The `highcharts` prop is now optional:

```
// Before (both props required)
<HighchartsReact
  highcharts={Highcharts}
  options={chartOptions}
  ref={chartRef}
/>

// After (only options required)
<HighchartsReact
  options={chartOptions}
  ref={chartRef}
/>
```

Module Resolution

Potential Impact: Some bundlers may require configuration updates

If you encounter module resolution issues: 1. Clear your build cache (see Installation Instructions) 2. Update your bundler configuration to resolve the new package name 3. Check for any explicit aliases that reference the old package

Code Examples

This section provides before/after examples for common chart scenarios. Each example shows the minimal changes required for migration.

Example 1: Basic Chart

Before (`highcharts-react-official`):

```
import React from 'react';
import Highcharts from 'highcharts';
import HighchartsReact from 'highcharts-react-official';

const BasicChart = () => {
  const options = {
    title: { text: 'Basic Chart' },
    series: [{
      data: [1, 2, 3, 4, 5]
    }]
  };

  return <HighchartsReact highcharts={Highcharts} options={options} />;
};
```

After (`@highcharts/react`):

```
import React from 'react';
import { Chart, Series, Title } from '@highcharts/react';

const BasicChart = () => {
  return (
    <Chart>
      <Title>Basic Chart</Title>
      <Series type="line" data={[1, 2, 3, 4, 5]} />
    </Chart>
  );
};
```

Example 2: Multiple Series Chart

Before (`highcharts-react-official`):

```
import React from 'react';
import Highcharts from 'highcharts';
import HighchartsReact from 'highcharts-react-official';

const MultiSeriesChart = () => {
  const options = {
    title: { text: 'Revenue by Quarter' },
    xAxis: {
      categories: ['Q1', 'Q2', 'Q3', 'Q4']
    },
    series: [{
      name: '2023',
      data: [100, 120, 140, 160]
    }, {
```

```
      name: '2024',
      data: [110, 130, 150, 170]
    }]
  };

  return <HighchartsReact highcharts={Highcharts} options={options} />;
};
```

After (`@highcharts/react`):

```
import React from 'react';
import { Chart, Series, Title, XAxis } from '@highcharts/react';

const MultiSeriesChart = () => {
  return (
    <Chart>
        <Title>Revenue by Quarter</Title>
        <XAxis categories={['Q1', 'Q2', 'Q3', 'Q4']} />
        <Series type="column" data={[100, 120, 140, 160]} options={{ name: "2023" }} />
        <Series type="column" data={[110, 130, 150, 170]} options={{ name: "2024" }} />
    </Chart>
    );
};
```

Example 3: Chart with Custom Configuration

Before (`highcharts-react-official`):

```
import React from 'react';
import Highcharts from 'highcharts';
import HighchartsReact from 'highcharts-react-official';

const CustomChart = () => {
  const options = {
    chart: {
      type: 'area',
      backgroundColor: '#f8f9fa'
    },
    title: { text: 'Sales Trend' },
    yAxis: {
      title: { text: 'Sales ($)' }
    },
    series: [{
      name: 'Sales',
      data: [1000, 1200, 1100, 1300, 1500]
    }]
  };
```

```
  return <HighchartsReact highcharts={Highcharts} options={options} />;
};
```

After (`@highcharts/react`):

```
import React from 'react';
import { Chart, Series, Title, YAxis } from '@highcharts/react';

const CustomChart = () => {
    return (
        <Chart
            options={{
                chart: {
                    backgroundColor: "#f8f9fa"
                }
            }}>
            <Title>Sales Trend</Title>
            <YAxis>Sales ($)</YAxis>
            <Series
                type="area"
                data={[1000, 1200, 1100, 1300, 1500]}
                options={{
                    name: "Sales"
                }}
            />
        </Chart>
    );
};
```

Accessing Highcharts Instance

The new `@highcharts/react` package provides several ways to access and configure the underlying Highcharts instance.

Accessing the Global Highcharts Instance

Before (`highcharts-react-official`):

```
import Highcharts from 'highcharts';
import HighchartsReact from 'highcharts-react-official';

// Set global options
Highcharts.setOptions({
  chart: { animation: false }
});

const options = {
```

```
  title: { text: 'My Chart' },
  series: [{
    type: 'line',
    data: [1, 2, 3, 4, 5]
  }]
};

// Use in component
<HighchartsReact highcharts={Highcharts} options={options} />
```

After (`@highcharts/react`):

```
import { Chart, Series, Title, Highcharts } from '@highcharts/react';

// Access built-in Highcharts instance directly
Highcharts.setOptions({
  chart: { animation: false }
});

// Use in component - no need to pass Highcharts
<Chart>
  <Title>My Chart</Title>
  <Series type="line" data={[1, 2, 3, 4, 5]} />
</Chart>
```

Accessing Individual Chart Instances

Before (`highcharts-react-official`):

```
import React, { useRef, useEffect } from 'react';
import Highcharts from 'highcharts';
import HighchartsReact from 'highcharts-react-official';

const ChartWithRef = () => {
  const chartRef = useRef(null);

  const options = {
    title: { text: 'Chart with Ref' },
    series: [{
      type: 'line',
      data: [1, 2, 3, 4, 5]
    }]
  };

  useEffect(() => {
    if (chartRef.current) {
      const chart = chartRef.current.chart;
```

```
      // Access chart methods
      chart.reflow();
    }
  }, []);

  return (
    <HighchartsReact
      highcharts={Highcharts}
      options={options}
      ref={chartRef}
    />
  );
};
```

After (`@highcharts/react`):

```
import type { HighchartsReactRefObject } from '@highcharts/react'

import React, { useRef, useEffect } from 'react';
import { Chart, Series, Title } from '@highcharts/react';

const ChartWithRef = () => {
    const chartRef = useRef<HighchartsReactRefObject>(null);

    useEffect(() => {
        if (chartRef.current?.chart) {
            // Access chart instance
            chartRef.current.chart.reflow();
        }
        if (chartRef.current?.container) {
            // Access container element
            console.log('Container:', chartRef.current.container);
        }
    }, []);

    return (
        <Chart ref={chartRef}>
            <Title>Chart with Ref</Title>
            <Series type="line" data={[1, 2, 3, 4, 5]} />
        </Chart>
    );
};
```

Setting Custom Highcharts Instance

For advanced use cases where you need a custom Highcharts build:

Before (`highcharts-react-official`):

```
import Highcharts from 'highcharts/highcharts';
import HighchartsReact from 'highcharts-react-official';
import 'highcharts/modules/exporting';
import 'highcharts/modules/accessibility';

const options = {
  title: { text: 'Custom Chart' },
  series: [{
    type: 'line',
    data: [1, 2, 3, 4, 5]
  }]
};

// Use custom Highcharts instance
<HighchartsReact highcharts={Highcharts} options={options} />
```

After (`@highcharts/react`):

```
import { Chart, Series, Title, setHighcharts } from '@highcharts/react';
import Highcharts from 'highcharts/highcharts';
import 'highcharts/modules/exporting';
import 'highcharts/modules/accessibility';

// Set custom Highcharts instance globally
setHighcharts(Highcharts);

// All charts will now use your custom instance
export function ChartWithCustomHC() {
    return (
        <Chart>
            <Title>Custom Chart</Title>
            <Series type="line" data={[1, 2, 3, 4, 5]} />
        </Chart>
    );
}
```

Using Chart Methods and Events

Before (`highcharts-react-official`):

```
import Highcharts from 'highcharts';
import HighchartsReact from 'highcharts-react-official';

const options = {
  chart: {
    events: {
```

```
      load: function() {
        console.log('Chart loaded:', this);
      }
    }
  },
  series: [{
    data: [1, 2, 3],
    events: {
      click: function(e) {
        console.log('Series clicked:', e);
      }
    }
  }]
};

<HighchartsReact highcharts={Highcharts} options={options} />
```

After (`@highcharts/react`):

```
import { Chart, Series } from '@highcharts/react';

<Chart options={{
  chart: {
    events: {
      load: function() {
        console.log('Chart loaded:', this);
      }
    }
  }
}}>
  <Series
    type="line"
    data={[1, 2, 3]}
    options={{
      events: {
        click: function(e) {
          console.log('Series clicked:', e);
        }
      }
    }}
  />
</Chart>
```

Advanced Features

This section covers migration strategies for advanced Highcharts features including custom modules, themes, and plugins.

Custom Modules and Specialized Charts

The `@highcharts/react` package provides specialized components for different chart types:

```
// Stock Charts
import { StockChart, StockSeries } from '@highcharts/react/Stock';

export function StockExample() {
    return (
        <StockChart>
            <StockSeries
                type="candlestick"
                data={[
                    [1609459200000, 100, 110, 90, 105],
                    [1609545600000, 105, 115, 95, 110],
                    [1609632000000, 110, 120, 100, 115],
                ]}
            />
        </StockChart>
    );
}
```

```
import { Chart } from '@highcharts/react';
import { VennSeries } from '@highcharts/react/series/Venn';

export function VennExample() {
  return (
    <Chart>
      <VennSeries data={[
        { sets: ['A'], value: 4 },
        { sets: ['B'], value: 1 },
        { sets: ['A', 'B'], value: 1 }
      ]} />
    </Chart>
  );
}
```

```
import { MapsChart } from '@highcharts/react/Maps';
import { MapSeries } from '@highcharts/react/series/Map';

import mapData from '@highcharts/map-collection/custom/scandinavia.geo.json' with { type: '

export function MapExample() {
  return (
    <MapsChart
      options={{
```

```
        chart: { map: mapData }
      }}
    >
      <MapSeries
        data={[
          { 'hc-key': 'no', value: 1 },
          { 'hc-key': 'dk', value: 2 },
          { 'hc-key': 'se', value: 3 }
        ]}
      />
    </MapsChart>
  );
}
```

Server-Side Rendering (SSR)

SSR compatibility can be achieved using dynamic imports or client-side only components:

```
// Next.js App Router - Mark component as client-side
'use client';

import { Chart, Series, Title } from '@highcharts/react';

export default function ChartComponent({ data, title }) {
  return (
    <Chart>
      <Title>{title}</Title>
      <Series type="line" data={data} />
    </Chart>
  );
}

// For Pages Router or when dynamic loading is preferred
import dynamic from 'next/dynamic';

const Chart = dynamic(() => import('./Chart'), {
  ssr: false
});

export default function ChartPage({ data, title }) {
  return <Chart data={data} title={title} />;
}
```

Last updated: 2025-10-08