HyJavaApex Full Documentation

the java api for apexcharts

OVERVIEW

The HyJavaApex Java library is a pure Java wrapper for the ApexCharts JavaScript library.
It allows your Java-based web applications to configure ApexCharts charts using only Java methods. There is no need to write any JavaScript.
The library will generate the required JavaScript code that ApexCharts needs to render the charts.

The Apexchart class is the starting point for all charts.

To define the contents of a chart, you simply have to instantiate the Apexchart class and use its setter methods to define the chart options and chart data as required. For details on the effects of each chart option refer to the ApexCharts API reference. The naming of the chart options in the HyJavaApex library in most cases exactly follows the ApexCharts JavaScript API.

In the example below we are defining a bar chart.
Instantiate the Apexchart class and then get a reference to ChartOptions. You then set the chart options as required.

```
public class YourBarChart {

    public Apexchart configure() {
        Apexchart apexchart = new Apexchart();
        ChartOptions chartOptions = apexchart.getChartOptions();

        chartOptions.getChart().setHeight(450).setType(ChartType.BAR);
        chartOptions.getPlotOptions().getBar().setHorizontal(true);
        chartOptions.getDataLabels().setEnabled(false);

        set all your required chart options......

        return apexchart;
    }

}
```

JAVADOC API

In most cases the HyJavaApex API exactly matches the ApexCharts API. There is no need to learn another API. The HyJavaApex API can be viewed online or refer to the API jar provided in the product download.

DEMO WEB APPLICATION

To give you a kick start using the library, the demo web application provides thorough examples of more than 60 different charts including Java source code for each example. Most of the ApexCharts Demos from the ApexCharts Demos page are included in this application.
The Java source code for the demo application including all chart examples is available from the Other Downloads page.

HOW TO USE THE LIBRARY

Java

Include the HyJavaApex jar in your project.

Configure your chart and generate the JavaScript code for the chart options.

```
Apexchart apexchart = new YourBarChart().configure();
String chartOptionsJs = apexchart.chartOptionsToJs();
```

The YourBarChart class will look something like this:

```
public class YourBarChart {

    public Apexchart configure() {
        Apexchart apexchart = new Apexchart();
        ChartOptions chartOptions = apexchart.getChartOptions();

        chartOptions.getChart().setHeight(450).setType(ChartType.BAR);
        chartOptions.getPlotOptions().getBar().setHorizontal(true);
        chartOptions.getDataLabels().setEnabled(false);

        set all your required chart options......

        return apexchart;
    }

}
```

JavaScript

Include required JavaScript on your page as per ApexCharts documentation.

```
var chart = new ApexCharts(document.querySelector("#chart"), ${chartOptionsJs});
chart.render();
```

Your Java code will pass the chartOptionsJs strings to the markup variable ${chartOptionsJs}.
The method used to pass the generated chart options from the Java to the web page will vary depending on your java web framework. *(Refer to Java Web Frameworks below).*

Web Page

Your markup will look something like this:

```
<script src="https://cdn.jsdelivr.net/npm/apexcharts"></script>

<div id="chart" style="max-width: 650px; margin: 35px auto; height: 800px;"></div>

<script>
var chart = new ApexCharts(document.querySelector("#chart"), ${chartOptionsJs});
chart.render();
</script>
```

That is all that is required to get a working Bar chart example.

MAIN LIBRARY CLASSES

ChartOptions Class

The ChartOptions class maps directly to the ApexCharts chart options in the ApexCharts API.
All ApexCharts chart options will be configured using setter methods on this class.

Apexchart Class

The Apexchart class wraps ChartOptions classes into one convenient class to make usage as simple as possible.
This class handles chart theming; the setting of chart options; generates JavaScript for ChartOptions.
All ApexCharts chart options can be configured by getting a reference to ChartOptions from this class via getChartOptions().
Usage of the Apexchart class has been described in examples above.

ApexchartRenderer Class

The HyJavaApex library performance is extremely fast right out of the box.
In cases where thousands of charts are generated, for example batch processes, some improvements can be found using ApexchartRenderer.toJs(...) rather than the Apexchart.chartOptionsToJs().
Instantiate the ApexchartRenderer class once and then use it to generate the chart options JavaScript for multiple charts.
Note that there are no improvements when generating single charts.

OPTIONS WITH MULTIPLE DATA TYPES

Where a chart option has more than one data type, eg. Number and String the Java API will include one method per type.
Setter methods will be overloaded. Getter methods will include a suffix.

```
chartOptions.getChart().setWidth(25);
chartOptions.getChart().setWidth("50%");

chartOptions.getChart().getWidthAsNumber();
chartOptions.getChart().getWidthAsString();
```

MAIN OPTION DATA TYPES

Color Class

To set a color string value use: new Color("#558F3B"); or new Color().setColor("#558F3B");
To set a JavaScript color value use: new Color().setColorValue("jsvalue");
To set a RGB color value use: new Color(255, 255, 255); or new Color().setRGB(255, 255, 255);
To set a RGBA color value use: new Color(255, 255, 255, 0.5); or new Color().setRGBA(255, 255, 255, 0.5);
140 Web color code strings are also defined in this class. Eg. new Color(Color.ALICEBLUE).

Function Class

To set the body of a Function use: new Function("functionbodystring"); or new Function().setFunctionBody("functionbodystring");
To add parameters to the function use: new Function("functionbodystring", "parameter"); or addParameter("parameter");

General Class

To set a JavaScript value use: new General("jsvalue"); or new General().setJsValue("jsValue");

JSON

ApexCharts call back functions that are inline functions are not strict JSON. JSON does not allow inline functions.

If you serialise/deserialise in your Java you may get an invalid JSON error.

The library gives you the option to treat the inline functions as strings. They can then be serialised/deserialised.
As ApexCharts expects a function you will need to convert them back in your JavaScript before rendering the chart.
The HyJavaApex library provides a JavaScript function: *hyjavaChartsInflateFunctions(yourChartConfig)* to convert them.

The JavaScript can be downloaded from https://www.hyjavacharts.com/content/js/hyjavacharts-inflate-1.0.0.zip and included in your markup.

The Function class provides a method setEmitAsString(boolean emitAsString) – will emit this function as a string from the Apexchart.chartOptionsToJs() method.

The Apexchart class provides a static method setEmitFunctionsAsString(boolean emitFunctionsAsString) – will emit all functions as strings from the Apexchart.chartOptionsToJs() method.

ARRAYS

There are methods for each chart option array to help simplify usage where there is only 1 element in the array.
In the example the getYaxisSingle() method will add a new Yaxis object to the yaxis array and operate on that object.

```
Yaxis yaxis = new Yaxis();
yaxis.setReversed(true);
chartOptions.getYaxis().add(yaxis);
   OR
chartOptions.getYaxisSingle().setReversed(true);
```

THEMES

If your Java code uses the chartOptions.getColors() method you will also need to set the Theme in your Java prior to calling the method.
eg. *apexchart.setTheme(ApexchartTheme.DEFAULT);*

If you have created your own theme you will need to create a Theme class as follows:
(1) create a new theme class implementing Theme.
(2) add the method getColors() to the class returning your theme colors.

```
public class YourNewTheme implements Theme {
  public List<Color> getColors() {
    return new ArrayList(Arrays.asList(
      new Color(Color.LIGHTSKYBLUE), new Color(Color.LIGHTGREEN), new Color(Color.LIGHTSALMON),
      new Color(Color.LIGHTCORAL), new Color(Color.LIGHTSTEELBLUE), new Color(Color.LIGHTGRAY),
      new Color(Color.LIGHTPINK), new Color(Color.LIGHTSEAGREEN), new Color(Color.LIGHTCYAN),
      new Color(Color.LIGHTBLUE), new Color(Color.LIGHTYELLOW), new Color(Color.LIGHTSLATEGRAY)
    ));
  }
}
```

You can also use color strings ie. new Color("#87cefa").
To use the theme: *apexchart.setTheme(new YourNewTheme());*

CHART DATA SOURCES

Chart data sources can be databases, web services, files or any resource your application server has access to.

JAVA VERSION

The HyJavaApex library requires Java 8 or greater. There are no external dependencies.

API ISSUES

Every endeavor has been made to ensure the HyJavaApex Java API matches the ApexCharts JavaScript API. Please contact us if there is a chart option missing from the Java API or if an option data type does not match the ApexCharts API. We will publish an updated API as soon as possible.

Generic Chart Options

To allow the developer to continue until the Java API is updated, the method *setString name, Object value)* is available on every ApexChart model class. This method allows you to configure any name/value pairs.

```
// If an option "allowAnimation" Boolean was missing from Chart in the API or the data type is incorrect
// you can still include the option as shown below
chartOptions.getChart().set("allowAnimation", true);

// If the option missing is a JSON object, you can use the General class.
chartOptions.getChart().set("someobject", new General("{ ... ... }"));
```

SUPPORT

For any questions or issues on the HyJavaApex API, please contact our support team via the Contact Us area on the home page.
Please refer to the ApexCharts API for usage of each chart option or contact ApexCharts for support of the ApexCharts product.

Copyright © 2026 Powerbuilt Systems. All rights reserved.