# Manuale d’uso

## Requisiti

Per poter utilizzare la libreria in una pagina web c’è la necessità che venga incluso nella pagina sia la libreria chart.js che il framework jQuery.

Quindi necessità di queste due inclusioni nel header:

<**head**>

<**script src="path\_to\_your\_js/jquery-3.2.1.min.js"**></**script**>  
 <**script src="path\_to\_your\_js/Chart.js"**></**script**>  
</**head**>

## Importazione

Per importare nella applicazione web la libreria bisgona importare i seguenti file:

<**head**>

<**script src="path\_to\_your\_js/jquery-3.2.1.min.js"**></**script**>  
 <**script src="path\_to\_your\_js/Chart.js"**></**script**>

<**script src="path\_to\_your\_js/EasyGraph.js"**></**script**>  
</**head**>

## Inserimento di un grafico

Inserimento di un grafico nella pagina web è necessario solo inserire una riga come questa nella posizione che si vuole avere il grafico:

<**easygraph id="unique\_id" chartType="chart\_type" json="url\_to\_json\_data" filters="filter\_name[,filter\_name2]"**></**easygraph**>

### unique\_id

Id unico che identifica l’elemento

### chart\_type

Tipo di grafico che si vuole creare.

Può assumere i seguenti valori:

* bar
* line
* radio
* pie
* doughnut
* polarArea
* bubble
* scatter

### url\_to\_json\_data

URL o PATH da dove verranno caricati i dati da visualizzare nel grafico. Il contenuto del documento caricato deve essere in formato JSON e avere la struttura corretta per il filtro o per il grafico che si vuole creare.

### filter\_name

I filtri da applicare si dovranno specificare in questo attributo in ordine di applicazione e divisi da virgola.

## Requisiti del backend

Il documento che viene caricato dal backend deve essere un oggetto JSON e avere la struttura corretta per il grafico o il filtro che la utilizzerà.

Per dettagli su tutte le feature dei grafici e le varie impostazioni che si possono specificare si rimanda alla documentazione di chart.js: <http://www.chartjs.org/docs/latest/charts/>

La struttura minima dovrà essere come segue:

### Bar, Line, Doughnut, Pie, PolarArea

{  
 **"labels"**: [**"Red"**, **"Blue"**, **"Yellow"**, **"Green"**, **"Purple"**, **"Orange"**],  
 **"datasets"**: [{  
 **"label"**: **"# of Votes"**,  
 **"data"**: [12, 19, 3, 5, 2, 3]  
 }  
}

### Bubble

{  
 **"datasets"**: [  
 {  
 **"label"**: **"First Dataset"**,  
 **"data"**: [  
 {**"x"**: 20, **"y"**: 30, **"r"**: 5},  
 {**"x"**: 40, **"y"**: 10, **"r"**: 10},  
 {**"x"**: 10, **"y"**: 5, **"r"**: 5},  
 {**"x"**: 15, **"y"**: 15, **"r"**: 10},  
 {**"x"**: 25, **"y"**: 10, **"r"**: 15},  
 {**"x"**: 30, **"y"**: 15, **"r"**: 10}  
 ]  
 }  
 ]  
}

### Scatter

{  
 **"datasets"**: [  
 {  
 **"label"**: **"First Dataset"**,  
 **"data"**: [  
 { **"x"**: 20, **"y"**: 30},  
 {**"x"**: 40, **"y"**: 10},  
 {**"x"**: 10, **"y"**: 5},  
 {**"x"**: 15, **"y"**: 15},  
 {**"x"**: 25, **"y"**: 10},  
 {**"x"**: 30, **"y"**: 15}  
 ]  
 }  
 ]  
}

## Creazione nuovi filtri

Se si vogliono inserire nuovi filtri che fanno al caso specifico di una situazione si possono aggiungere andando a specificare una funzione nel seguente modo:

EasyGraphFilter.myNewFilter = **function** (data){

**// codice che modifica il data**

**return** data;

}

Il parametro “*data*” è obbligatorio come anche il valore che ritorna il quale dovrà essere nel formato giusto per il filtro successivo o per il grafico che si vuole creare.