

# Advanced Dashboard Charts

Repo: php-adv-charts

Creiamo una dashboard di grafici utilizzando tutti i linguaggi visti fino ad ora. Html, css, javascript e php.

## Tecnologie da utilizzare e architettura:

La parte frontend deve utilizzare HTML e Javascript utilizzando la libreria ChartJS.

I dati riguardanti i grafici saranno salvati in PHP e verranno passati a javascript attraverso una chiamata ajax.

La soluzione da adottare è quindi quella di avere i seguenti file:

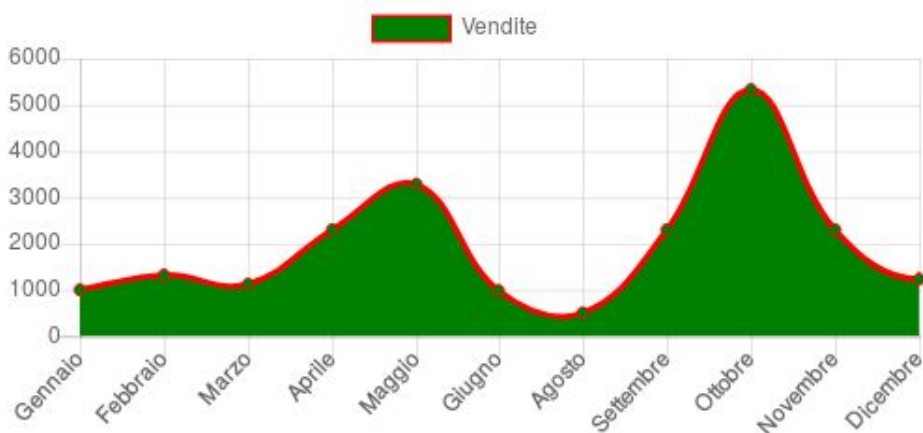
- Il file **index.php** che contiene la parte di HTML
- Il file **database.php** che conterrà i dati
- Il file **server.php** che esporrà i dati via ajax al front end (divisibile anche in diversi file, uno per tipo di grafico)
- Eventuale file **style.css**

## Step 1

Il file **data.php** è il seguente:

```
<?php
    $data = [1000,1322,1123,2301,3288,988,502,2300,5332,2300,1233,2322];
?>
```

Si vuole creare un grafico a linea che mostri l'andamento mensile delle vendite, ogni numero fornito è il valore totale delle vendite per un mese, partendo da gennaio



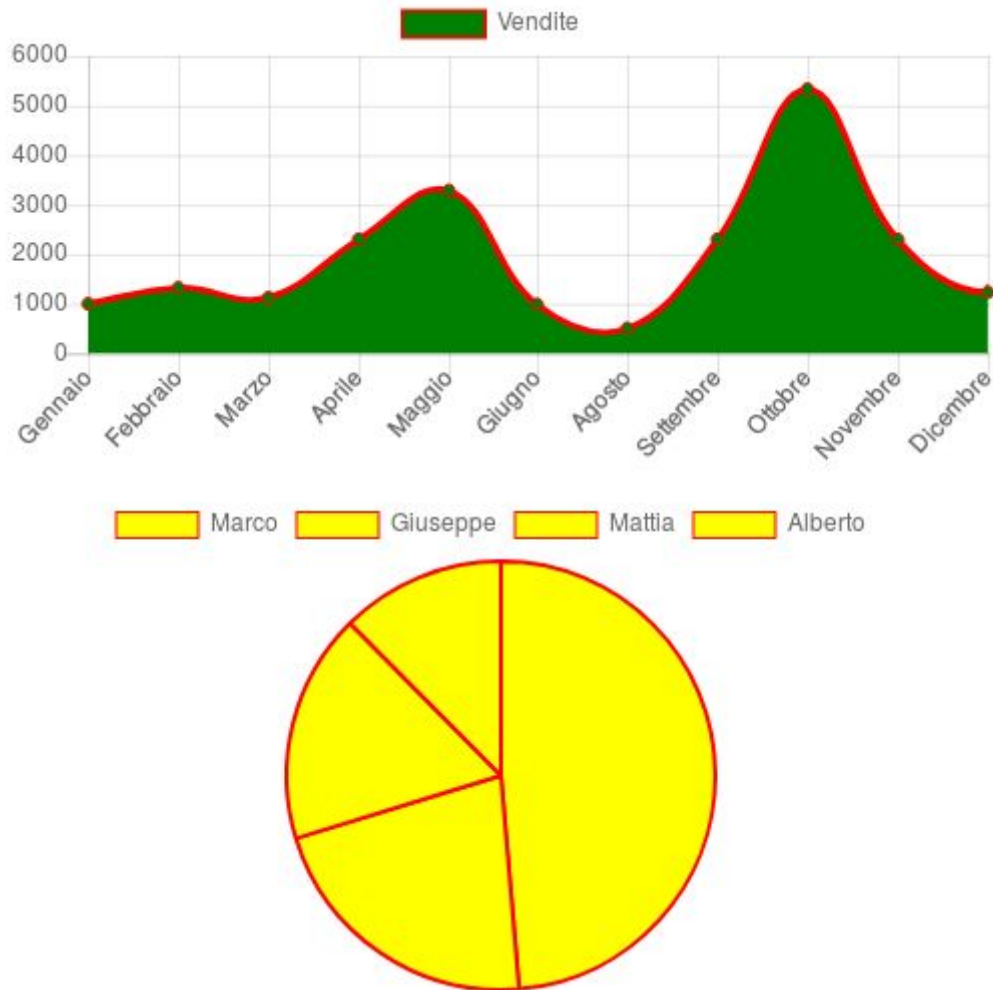
## Step 2

Il file **data.php** è il seguente:

```
<?php
    $graphs = [
        'fatturato' => [
            'type' => 'line',
            'data' => [1000,1322,1123,2301,3288,988,502,2300,5332,2300,1233,2322]
        ],
        'fatturato_by_agent' => [
            'type' => 'pie',
            'data' => [
                'Marco' => 9000,
                'Giuseppe' => 4000,
                'Mattia' => 3200,
                'Alberto' => 2300
            ]
        ]
    ];
?>
```

Si vogliono creare due grafici utilizzando i dati contenuti dentro al file **data.php**. Questa volta il file contiene non solo i dati ma anche il tipo di grafico atteso.

Il primo grafico è lo stesso del punto precedente mentre il secondo è un grafico a torta; per questo grafico i dati forniti non sono solo i “valori” ma anche le “label”



### Step 3

Il file **data.php** è il seguente:

```
<?php
$graphs = [
    'fatturato' => [
        'type' => 'line',
        'data' => [1000,1322,1123,2301,3288,988,502,2300,5332,2300,1233,2322],
        'access' => 'guest'
    ],
    'fatturato_by_agent' => [
        'type' => 'pie',
        'data' => [
            'Marco' => 9000,
            'Giuseppe' => 4000,
            'Mattia' => 3200,
            'Alberto' => 2300
        ]
    ],
];
```

```

        'access' => 'employee'
    ],
    'team_efficiency' => [
        'type' => 'line',
        'data' => [
            'Team1' => [1,0.8,0.7,0.5,0.7,0.8,0.9,0.5,0.6,1,0.3,0.9],
            'Team2' => [0.3,0.6,0.8,0.3,0.6,0.5,0.8,0.7,0.3,0.5,0.6,1],
            'Team3' => [0.2,0.1,0.5,0.1,0.6,0.5,0.4,0.6,0.3,0.4,0.5,0.7],
        ],
        'access' => 'clevel'
    ],
];
?>

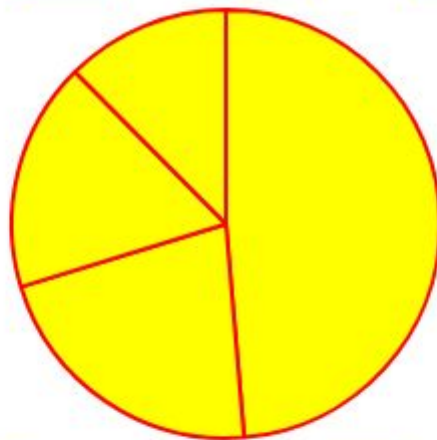
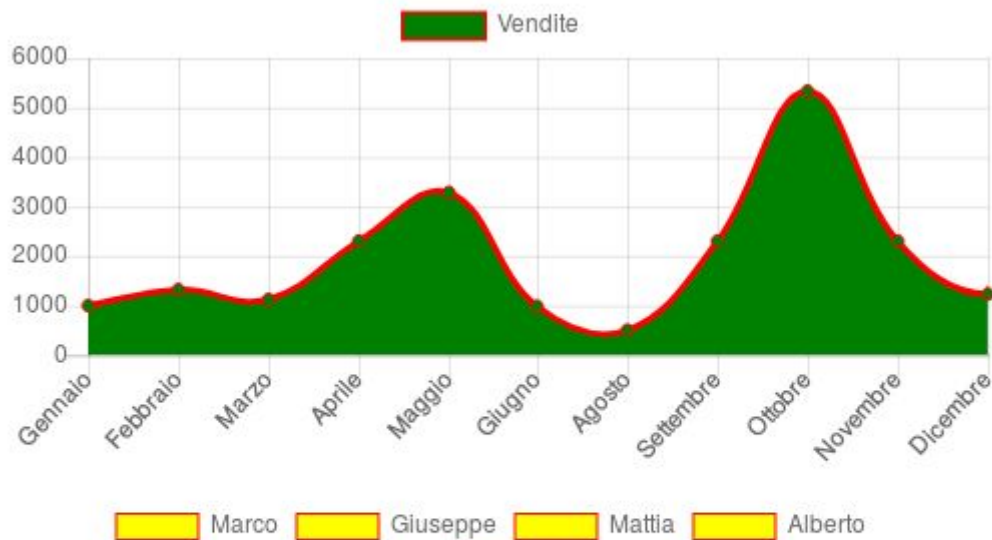
```

Questo file introduce due novità:

- Un nuovo grafico a linea che mostra l'andamento di efficienza dei 3 team mese per mese, una linea per ogni team
- Un elemento access che definisce il livello minimo di chi può vedere quel grafico, con questo significato: i grafici con access *'guest'* possono essere visti da tutti, i grafici con access *'employee'* possono essere visti da chi è *employee* e chi è *clevel*, tutti i grafici con access *'clevel'* possono essere visti solo da chi è *clevel*.

Il livello dell'utente che visita la pagina viene preso da querystring utilizzando \$\_GET['level'] (es: <http://localhost/index.php?level=guest>)

**Risultato per l'accesso clevel:**



### Consigli:

1. Ordine! Con aperture e chiusure di php, apici, etc. diventa difficile perdersi. Meglio formattare e indentare bene prima.
2. A volta avete problemi JS (client-side) altri PHP (server-side): tenete sempre aperta la console e il terminare con `tail -f /var/log/apache2/error.log`