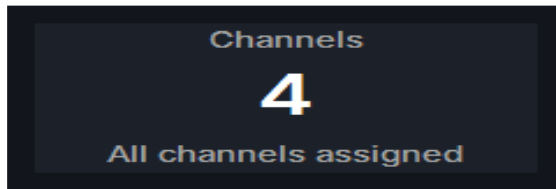


CHANNELS



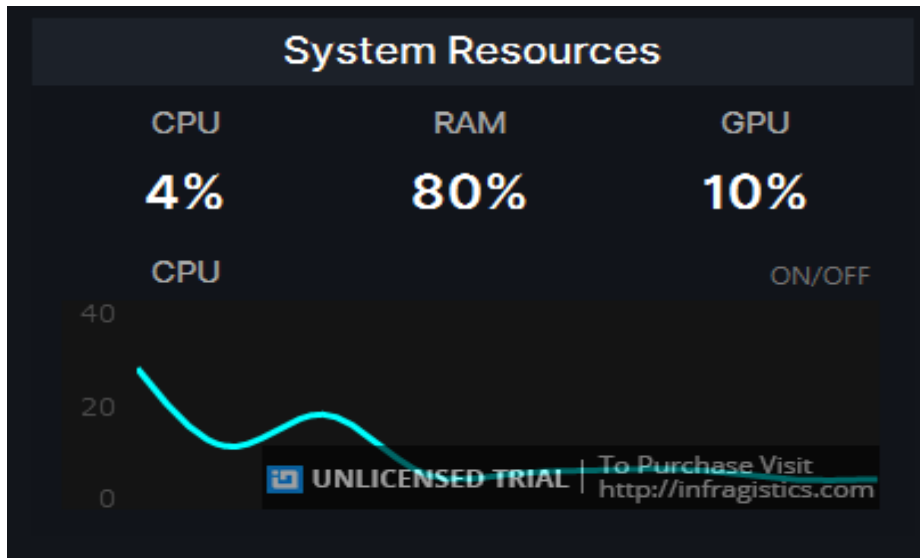
Api call - `Channels/count` and you get `ChannelListCountDto`

```
public class ChannelListCountDto
{
    public int? TotalChannels { get; set; }
    public int? Connected { get; set; }
    public int? Disconnected { get; set; }
}
```

Middle text is `TotalChannels`, and text down, I used this method

```
if (channels?.Connected.HasValue == true && channels?.TotalChannels.HasValue ==
true)
{
    if (channels.Connected == channels.TotalChannels)
    {
        cardChannels.Invoke(new Action(() => cardChannels.TextDown = "All channels
assigned"));
    }
    else
    {
        var disconnected = channels.Disconnected ?? 0;
        cardChannels.Invoke(new Action(() => cardChannels.TextDown =
${disconnected} channels not assigned));
    }
}
else
{
    cardChannels.Invoke(new Action(() => cardChannels.TextDown = "No data"));
}
```

SYSTEM RESOURCES



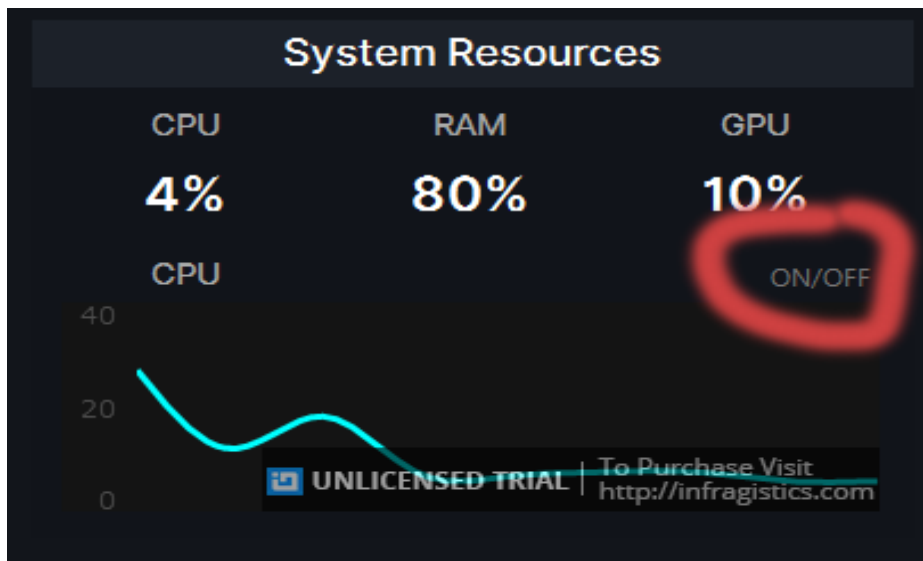
`ServiceMessages.Metrics` is the message SignalR gets

```
CommandDto? comm = data.ToObject<CommandDto>();

SystemMetricsDto? dddata = comm.data.ToObject<SystemMetricsDto>();
if (dddata != null)
{
    systemResources1.SetData(dddata);
}





internal void SetData(SystemMetricsDto? dddata)
{
    if (dddata != null)
    {
        lblCPU.Text = dddata.cpuPercent + "%";
        lblGPU.Text = dddata.gpuPercent + "%";
        lblRam.Text = dddata.ramPercent + "%";

        AddCpu(dddata.cpuPercent); - this is % for chart
    }
}
```



For turn on / off resources, send to API (so not SignalR) `Systemsettings/settelemetrycontrol`

ERRORS

Errors		
	Audio format mismatch	I9 ULTIMATE
	23.12.2025	
	Video engine crash	I9 ULTIMATE
	22.12.2025	
	Buffer underrun	I9 ULTIMATE
	21.12.2025	
	Stream connection lost	I9 ULTIMATE
	21.12.2025	

UNLICENSED TRIAL | To Purchase Visit <http://infragistics.com>

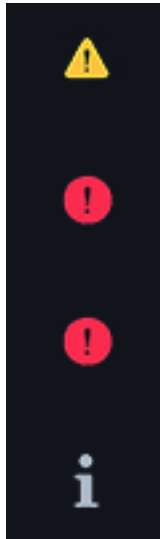
API call: `Errors/lastweek`

3 type of errors

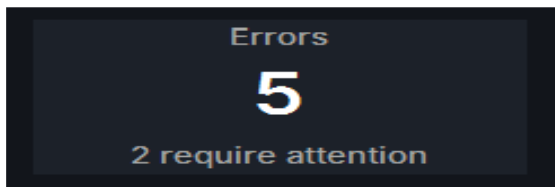
0 - error

1 - warning

2 - information



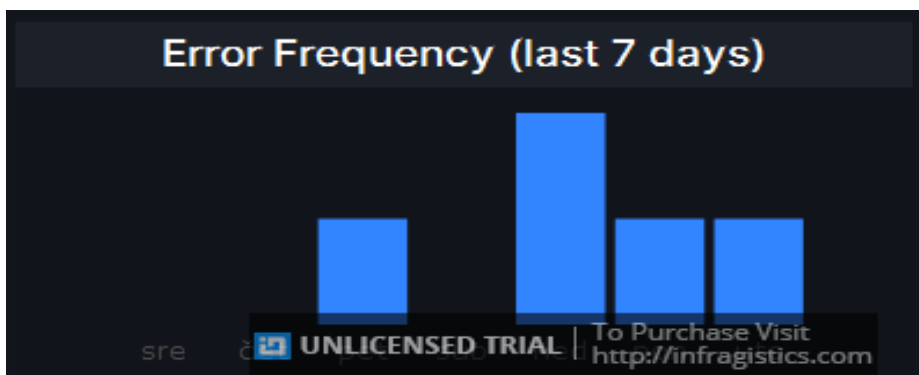
Api returns `ErrorListDto` and there you get Type, Time, Description and PlayoutName



Middle number is Count of `ErrorListDto`, and text down is number of errors (Type = 0)

```
int temp = Errors.Where(p => p.Type == 0).Count();
```

```
if (temp > 0) error = $"{temp.ToString()} require attention"; else error = "No  
critical errors";
```



For this graph, I use `ErrorFrequencyDto`, so this is the code. We use same `ErrorListDto` we got for Errors grid. So this must execute after filling the grid

```
var today = DateTime.Today;

// 1. generate 7 days list
var days = Enumerable.Range(0, 7)
    .Select(i => today.AddDays(-i))
    .OrderBy(d => d)
    .ToList();

// 2. group errors by time
var grouped = errors
    .Where(e => e.Time.Date >= today.AddDays(-6)) // zadnjih 7 dana
    .GroupBy(e => e.Time.Date)
    .ToDictionary(g => g.Key, g => g.Count());

// 3. make final list for chart
errorlist = days
    .Select(day => new ErrorFrequencyDto
    {
        Day = day.ToString("ddd"),
        Count = grouped.ContainsKey(day) ? grouped[day] : (int?)null
    })
    .ToList();
}
```

So you have grouped data for each day, just put to chart.

MEDIA ASSETS



API call `Assets/count`

And we get `SystemMetricsDto` (didn't want to make new dto just for this, I will probably do it later)

Number of assets (middle number) is `cpuPercent` value of `SystemMetricDto`, and text down is `gpuPercent` of `SystemMetricDto`. This is my code

```
cardAssets.TextCenter = assets.cpuPercent.ToString(); //number in the middle
if (assets.gpuPercent == 0)
{
    cardAssets.TextDown = "No change this week"; //text down
}
else
{
    if (assets.gpuPercent > 0)
    {
        cardAssets.TextDown = "+" + assets.gpuPercent.ToString() + " this week";
    }
}
```

```
else
{
    cardAssets.TextDown = "-" + assets.gpuPercent.ToString() + " this week";
}
}
```