

Appunti sviluppo

Appunti vari sviluppo progetto MAPO

MP/MON

Attualmente è un applicazione MVC asp.net c#, NON interattiva (display distribuiti stato produzione), effettua refresh timer-based.

NON ottimizzata (per diminuire le chiamate) e genera traffico e carico sul server

Parzialmente mitigato il carico su server con

- caching tramite redis dei dati elaborati
- ottimizzaizone stored
- caching a livello DB dei dati x stored

Obiettivi

Si prefigge di migliorare l'applicazione per 2 motivi

- performances
- reattività (logica push e non pull)
- semplificazione del metodo di peronalizzazione dei dati mostrati ai singoli clienti

Per i primi 2 punti si intende lavorare con un prototipo VUE.JS / REACT con cui valutare un approccio differente per l'applicazione che preveda di pubblicare, in modalità push dal server, i dati in modo che siano aggiornati sul client AL MOMENTO DELLA PRODUZIONE degli stessi (o nel minor tempo possibile/col minor carico possibile sul server)

Si intendeo fornire i dati (push o pull) tramite un a api REST che alla chiamata fornisca un traccato di TUTTI i valori di tutte le macchine (o in alternativa dell'unica macchina richiesta, a seconda del VERB usato)

A livello di configurazione, si vuole passare dall'attuale datamodel "tipizzato" in cui ogni variabile è indicata con il nome esatto proveniente dalla abse dati ad un insieme id variabili indicate da un dizionario KVP (chiave/valore) in cui l'oggetto della view da mostrare utilizzerà dei generici segnaposti (es: da NumPezzi --> label02) e a livello di configurazione backend, tramite LUT (lookupTable) andrà a mappare, per il singolo cliente, valori e label desiderate (tramite un file json serializzato sul DB, personalizzabile per SINGOLA MACCHINA)

WebAPI recupero dati

Per il recupero dei dati di una SINGOLA macchina si intende impiegare il seguente metodo:

```
CALL: http://{server_url}/MSE/getData/{cod_macchina}
```

che fornirà in risposta una stringa JSON (per singola macchina):

```
[{
  "RowNum": 1,
  "lastUpdate": "2019-04-03T18:33:14.24",
  "IdxMacchina": "SIM_DP_01",
  "CodMacchina": "SIMDP1",
  "Nome": "IOB SimDP1",
  "url": "Steamware.png",
  "idxODL": 2208,
  "CodArticolo": "027309",
  "NumPezzi": 500,
  "TCAssegnato": 1.08300000,
  "DataInizioODL": "2019-03-25T18:41:36.657",
  "Semaforo": "sVe",
  "idxStato": 13,
  "DescrizioneStato": "lavorazione",
  "durata": 557.0,
  "PezziProd": 20956,
  "PezziConf": 0,
  "TempoOn": 0.00000000,
  "TempoAuto": 0.00000000,
  "TempoRun": 0.00000000,
  "TCMedio": 0.00000000,
  "TCLav": 0.00000000,
  "TCEff": 0.00000000,
  "TCMedioRT": 0.00000000,
  "TCLavRT": 0.00000000,
  "TCEffRT": 0.00000000,
  "Disegno": ""
}]
```

Per il recupero dei dati si intende impiegare il seguente metodo (per ottenere TUTTE le macchine):

```
CALL: http://{server_url}/MSE/getData
```

che fornirà in risposta una stringa JSON di tipo array:

```
[{
  "RowNum": 1,
  "lastUpdate": "2019-04-03T18:33:14.24",
  "IdxMacchina": "SIM_DP_01",
  "CodMacchina": "SIMDP1",
  "Nome": "IOB SimDP1",
  "url": "Steamware.png",
  "idxODL": 2208,
  "CodArticolo": "027309",
  "NumPezzi": 500,
  "TCAssegnato": 1.08300000,
```

```

        "DataInizioODL": "2019-03-25T18:41:36.657",
        "Semaforo": "sVe",
        "idxStato": 13,
        "DescrizioneStato": "lavorazione",
        "durata": 557.0,
        "PezziProd": 20956,
        "PezziConf": 0,
        "TempoOn": 0.00000000,
        "TempoAuto": 0.00000000,
        "TempoRun": 0.00000000,
        "TCMedio": 0.00000000,
        "TCLav": 0.00000000,
        "TCEff": 0.00000000,
        "TCMedioRT": 0.00000000,
        "TCLavRT": 0.00000000,
        "TCEffRT": 0.00000000,
        "Disegno": ""
    }, {
        "RowNum": 2,
        "lastUpdate": "2019-04-03T18:33:50.167",
        "IdxMacchina": "SIM_DP_02",
        "CodMacchina": "SIMDP2",
        "Nome": "IOB SimDP2",
        "url": "Steamware.png",
        "idxODL": 2211,
        "CodArticolo": "005123",
        "NumPezzi": 100000,
        "TCAssegnato": 1.00000000,
        "DataInizioODL": "2019-03-25T18:44:37.3",
        "Semaforo": "sRo",
        "idxStato": 15,
        "DescrizioneStato": "allarme CN",
        "durata": 547.0,
        "PezziProd": 24151,
        "PezziConf": 1954,
        "TempoOn": 0.00000000,
        "TempoAuto": 0.00000000,
        "TempoRun": 0.00000000,
        "TCMedio": 0.00000000,
        "TCLav": 0.00000000,
        "TCEff": 0.00000000,
        "TCMedioRT": 0.00000000,
        "TCLavRT": 0.00000000,
        "TCEffRT": 0.00000000,
        "Disegno": ""
    }, {
        "RowNum": 3,
        "lastUpdate": "2019-04-03T18:33:28.75",
        "IdxMacchina": "SIMUL_01",
        "CodMacchina": "SIM05",
        "Nome": "IOB SIM 01",
        "url": "Steamware.png",
        "idxODL": 2220,
        "CodArticolo": "020293",

```

```
    "NumPezzi": 15000,  
    "TCAssegnato": 0.00000000,  
    "DataInizioODL": "2019-03-25T18:54:57.877",  
    "Semaforo": "sVe",  
    "idxStato": 13,  
    "DescrizioneStato": "lavorazione",  
    "durata": 568.0,  
    "PezziProd": 63929,  
    "PezziConf": 0,  
    "TempoOn": 0.00000000,  
    "TempoAuto": 0.00000000,  
    "TempoRun": 0.00000000,  
    "TCMedio": 0.00000000,  
    "TCLav": 0.00000000,  
    "TCEff": 0.00000000,  
    "TCMedioRT": 0.18630215,  
    "TCLavRT": 0.18499994,  
    "TCEffRT": 0.00000000,  
    "Disegno": "179L1000"  
  }  
}
```

Attuale struttura pagina SINGOLA macchina

Ogni singolo impianto è attualmente mostrato con una pagina razor

- che si adatta alla larghezza dello schermo
- che permette di specificare il NUM MAX di colonne da visualizzare tra 1 e 6
- che utilizza un criterio di flashing (tramite 2 css alternati ogni secondo quando il colore NON E' verde)

... così composta:

```
@model IEnumerable<MP_MON.Models.MappaStatoExpl>  
  
<div class="row statusMap mx-2 mt-3 mb-1">  
  
  @{  
    // salvo variabili num blocchi x riga  
    int numBlock = 0;  
    int maxCol = 6;  
    try  
    {  
      maxCol = ViewBag.maxCol;  
    }  
    catch  
    {  
      maxCol = 6;  
    }  
    if (maxCol == 0)  
    {  
      maxCol = 6;  
    }  
  }  
</div>
```

```

    }
}

@foreach (var item in Model)
{
    // fix codice semaforo: sVe -> Ve, sRo ->Ro...
    string codSemafor = item.Semaforo;
    if (codSemafor.Length == 3)
    {
        codSemafor = codSemafor.Substring(1, 2);
    }
    string cssStatus = ViewBag.baseCss + codSemafor;
    // calcolo durata...
    TimeSpan TCass = TimeSpan.FromMinutes((double)item.TCAssegnato);
    TimeSpan TCLav = TimeSpan.FromMinutes((double)item.TCLavRT);
    // converto a stringa!
    string TCAssegnato = TCass.ToString(@"mm\:ss");
    string TCLavorato = TCLav.ToString(@"mm\:ss");

    // verifico se mostrare articolo o disegno...
    string sArticolo = "";
    if (SteamWare.memLayer.ML.CRS("sART") == "CodArticolo")
    {
        sArticolo = item.CodArticolo;
    }
    else
    {
        sArticolo = item.Disegno;
    }
    // se fosse vuoto mostro cmq codArt tra parentesi...
    if (sArticolo == "")
    {
        sArticolo = string.Format("[{0}]", item.CodArticolo);
    }
    // se NON trova update da oltre "keepAliveMin" / 2 minuti rosso lo status
    comunicazione
    string cssComStatus = cssStatus;
    string showComErr = "invisible";
    if (DateTime.Now.Subtract((DateTime)item.lastUpdate).TotalSeconds >
        SteamWare.memLayer.ML.CRI("keepAliveMin") * 60 / 2)
    {
        cssComStatus = ViewBag.baseCss + "Ro";
        showComErr = "visible";
    }
    // verifico SE è disabilitata modalità animazione -> blink a stati (e refresh
    1s)
    if (SteamWare.memLayer.ML.CRS("doAnimate") == "1")
    {
        // blink se secondo pari...
        DateTime adesso = DateTime.Now;
        int resto = 0;
        Math.DivRem(adesso.Second, 2, out resto);
    }
}

```

```

        if (resto == 0)
        {
            cssStatus += "_b";
            if (cssComStatus.EndsWith("Ro"))
            {
                cssComStatus += "_b";
            }
        }
    }
}

<div class="col" style="padding:2px;">
    <div class="@cssStatus p-1">
        <div class="row mb-1">
            <div class="col-12">
                <div class="ui-title text-uppercase">@Html.DisplayFor(modelItem =>
item.Nome)</div>
            </div>
        </div>
        <div class="row pt-0 pb-2 px-1 fontSmall">
            <div class="col-4 pr-0">Art.</div>
            <div class="col-8 pl-0 text-right ui-art">@sArticolo</div>
        </div>
        <div class="row pt-0 pb-2 px-1 fontSmall">
            <div class="col-9 text-uppercase"><b>@Html.DisplayFor(modelItem =>
item.DescrizioneStato)</b></div>
            <div class="col-3 pl-0 text-right">@item.durata'</div>
            @*<div class="col-6 pr-0">OEE</div>
            <div class="col-6 pl-0 text-right">xx%</div>*@
            <div class="col-4 pr-0">T.Ciclo</div>
            <div class="col-4 pl-0 text-right">std: @TCAssegnato</div>
            <div class="col-4 pl-0 text-right">act: @TCLavorato</div>
            <div class="col-6 pr-0">Pezzi<sub>(prod/ord)</sub></div>
            <div class="col-6 pl-0 text-right">@Html.DisplayFor(modelItem =>
item.PezziProd) / @Html.DisplayFor(modelItem => item.NumPezzi)</div>
        </div>
    </div>
    <div class="@cssComStatus p-1">
        <div class="row fontSmaller mt-1">
            <div class="col-12">
                <div class="text-right ui-footer px-2">
                    <div class="row">
                        <div class="col-6 text-warning @showComErr">
                            <b>C.101</b>
                        </div>
                        <div class="col-6">
                            @Html.DisplayFor(modelItem => item.lastUpdate)
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
// controllo se ho resto zero --> uso NUOVA riga...
if ((numBlock % maxCol) == maxCol - 1)

```

```

    {
        @Html.Raw("</div><div class=\"row statusMap mx-2 my-1\">");
    }
    // incremento contatore
    numBlock++;
}
@{
    // controllo se NON SONO arrivato in fondo --> aggiungo blocchi!
    int currNum = (numBlock % maxCol);
    while (currNum < (maxCol))
    {
        @Html.Raw("<div class=\"col\" style=\"padding: 2px;\">aaa</div>");
        currNum++;
    }
}
</div>

```

con il seguente estratto css

```

/* area semafori*/
.semBlinkVe,
.semFixVe,
.semFixVe_b,
.semVe,
.semVe_b {
    background: #009036;
    background: rgba(0,255,80,.6);
    color: #FFFFAA;
}

.semBlinkGr,
.semFixGr,
.semFixGr_b,
.semGr,
.semGr_b {
    background-color: #bcbcbc;
    background: rgba(180,180,180,.6);
}

.semGi {
    text-align: left;
    background: #8a8d27;
    background: rgba(230,210,0,.6);
    padding: 0px 4px 0px 4px;
    color: #000;
}

.semGi_b,
.semFixGi,
.semFixGi_b {
    text-align: left;
}

```

```
background: #f9ff18;
background: rgba(255,255,0,.8);
padding: 0px 4px 0px 4px;
color: #333;
}

.semBl {
text-align: left;
background: #000E7A;
background: rgba(0,5,200,.6);
padding: 0px 4px 0px 4px;
color: #959500;
}

.semBl_b,
.semFixBl,
.semFixBl_b {
text-align: left;
background: #243FFF;
background: rgba(60,80,255,.8);
padding: 0px 4px 0px 4px;
color: #ffff32;
}

.semRo {
text-align: left;
background-color: #7a000e;
background: rgba(200,0,5,.6);
padding: 0px 4px 0px 4px;
color: #959500;
}

.semRo_b,
.semFixRo,
.semFixRo_b {
text-align: left;
background-color: #ff243f;
background: rgba(255,60,80,.8);
padding: 0px 4px 0px 4px;
color: #ffff32;
}
```