~~~
```mq4
#property indicator_chart_window

extern int AsiaStartHour = 0;    // Asia session start hour (broker time)
extern int AsiaEndHour = 6;      // Asia session end hour (broker time)
extern int LondonStartHour = 7;  // London session start hour (broker time)
extern int LondonEndHour = 15;   // London session end hour (broker time)
extern int NYStartHour = 12;     // New York session start hour (broker time)
extern int NYEndHour = 20;       // New York session end hour (broker time)

void OnInit()
{
    IndicatorShortName("Session Indicator");
}

void OnCalculate(const int rates_total, const int prev_calculated, const datetime &time[], const double &open[], const
double &high[],
                 const double &low[], const double &close[], const long &tick_volume[], const long &volume[], const int
&spread[])
{
    int counter = prev_calculated;
    datetime lastTime = 0;

    for (int i = counter; i < rates_total; i++)
    {
        if (TimeHour(time[i]) >= AsiaStartHour && TimeHour(time[i]) < AsiaEndHour)
        {
            // Asia session
            if (TimeHour(time[i]) == AsiaStartHour && TimeHour(lastTime) != AsiaStartHour)
                SetIndexBuffer(0, i);

            if (TimeHour(time[i]) == AsiaEndHour && TimeHour(lastTime) != AsiaEndHour)
                SetIndexBuffer(1, i);
        }
        else if (TimeHour(time[i]) >= LondonStartHour && TimeHour(time[i]) < LondonEndHour)
        {
            // London session
            if (TimeHour(time[i]) == LondonStartHour && TimeHour(lastTime) != LondonStartHour)
                SetIndexBuffer(2, i);

            if (TimeHour(time[i]) == LondonEndHour && TimeHour(lastTime) != LondonEndHour)
                SetIndexBuffer(3, i);
        }
        else if (TimeHour(time[i]) >= NYStartHour && TimeHour(time[i]) < NYEndHour)
        {
            // New York session
            if (TimeHour(time[i]) == NYStartHour && TimeHour(lastTime) != NYStartHour)
                SetIndexBuffer(4, i);

            if (TimeHour(time[i]) == NYEndHour && TimeHour(lastTime) != NYEndHour)
                SetIndexBuffer(5, i);
        }

        lastTime = time[i];
    }

    PlotIndexSetInteger(0, PLOT_DRAW_BEGIN, 0);
    PlotIndexSetInteger(1, PLOT_DRAW_BEGIN, 0);
    PlotIndexSetInteger(2, PLOT_DRAW_BEGIN, 0);
    PlotIndexSetInteger(3, PLOT_DRAW_BEGIN, 0);
    PlotIndexSetInteger(4, PLOT_DRAW_BEGIN, 0);
    PlotIndexSetInteger(5, PLOT_DRAW_BEGIN, 0);

    PlotIndexSetString(0, PLOT_LABEL, "Asia Start Candle");
    PlotIndexSetString(1, PLOT_LABEL, "Asia End Candle");
    PlotIndexSetString(2, PLOT_LABEL, "London Start Candle");
    PlotIndexSetString(3, PLOT_LABEL, "London End Candle");
    PlotIndexSetString(4, PLOT_LABEL, "NY Start Candle");
    PlotIndexSetString(5, PLOT_LABEL, "NY End Candle");
~~~

```
   PlotIndexSetInteger(0, PLOT_STYLE, STYLE_SOLID);
   PlotIndexSetInteger(1, PLOT_STYLE, STYLE_SOLID);
   PlotIndexSetInteger(2, PLOT_STYLE, STYLE_SOLID);
   PlotIndexSetInteger(3, PLOT_STYLE, STYLE_SOLID);
   PlotIndexSetInteger(4, PLOT_STYLE, STYLE_SOLID);
   PlotIndexSetInteger(5, PLOT_STYLE, STYLE_SOLID);

   PlotIndexSetInteger(0, PLOT_DRAW_BEGIN, 1);
   PlotIndexSetInteger(1, PLOT_DRAW_END, 1);
   PlotIndexSetInteger(2, PLOT_DRAW_BEGIN, 1);
   PlotIndexSetInteger(3, PLOT_DRAW_END, 1);
   PlotIndexSetInteger(4, PLOT_DRAW_BEGIN, 1);
   PlotIndexSetInteger(5, PLOT_DRAW_END, 1);

   PlotIndexSetInteger(0, PLOT_LINE_COLOR, clrYellow);
   PlotIndexSetInteger(1, PLOT_LINE_COLOR, clrYellow);
   PlotIndexSetInteger(2, PLOT_LINE_COLOR, clrLime);
   PlotIndexSetInteger(3, PLOT_LINE_COLOR, clrLime);
   PlotIndexSetInteger(4, PLOT_LINE_COLOR, clrBlack);
   PlotIndexSetInteger(5, PLOT_LINE_COLOR, clrBlack);
}
```

2) Indicator for the First 4 Hours of Asia Session:

This indicator will draw horizontal lines for the high and low of the first 4 hours of the Asia session. These lines will be visible on lower time frames as well.

```mq4
#property indicator_chart_window

extern int AsiaStartHour = 0;      // Asia session start hour (broker time)
extern int AsiaDurationHours = 4;  // Duration of the Asia session (in hours)

void OnInit()
{
   IndicatorShortName("Asia Session Indicator");
}

void OnCalculate(const int rates_total, const int prev_calculated, const datetime &time[], const double &open[], const
double &high[],
                const double &low[], const double &close[], const long &tick_volume[], const long &volume[], const int
&spread[])
{
   datetime asiaStartTime = 0;
   datetime asiaEndTime = 0;
   int asiaStartIndex = -1;

   for (int i = 0; i < rates_total; i++)
   {
      if (TimeHour(time[i]) >= AsiaStartHour && asiaStartTime == 0)
      {
         asiaStartTime = time[i];
         asiaStartIndex = i;
         asiaEndTime = TimeAdd(asiaStartTime, AsiaDurationHours * 3600);
      }

      if (time[i] >= asiaStartTime && time[i] <= asiaEndTime)
      {
         // Draw horizontal lines for the high and low
         if (i == asiaStartIndex)
            ObjectCreate("AsiaHighLine", OBJ_HLINE, 0, time[i], high[i]);

         if (i == asiaStartIndex + (AsiaDurationHours - 1) * PeriodSeconds(PERIOD_H1))
            ObjectCreate("AsiaLowLine", OBJ_HLINE, 0, time[i], low[i]);
      }
   }
```

```
    ObjectSet("AsiaHighLine", OBJPROP_COLOR, clrRed);
    ObjectSet("AsiaHighLine", OBJPROP_STYLE, STYLE_SOLID);
    ObjectSet("AsiaLowLine", OBJPROP_COLOR, clrRed);
    ObjectSet("AsiaLowLine", OBJPROP_STYLE, STYLE_SOLID);
  }
```