

```

package main.java.com.wfdai.weatherforecastdai.main;

import java.text.SimpleDateFormat; import main.java.com.wfdai.weatherforecastdai.main.weather.Weather;
import java.util.Date; import main.java.com.wfdai.weatherforecastdai.main.KPI.RegistoKPI;

/** * Transforma dados e Objectos em JSON na especificação PPMP @author
daniel */ public class Parser {

String parsedMessage;

public Parser() {
}

/**
 * Dá return de uma String com a mensagem em JSON...
 *
 * @return mensagem em JSON
 */
public String getParsedMessage() {
    return parsedMessage;
}

private final SimpleDateFormat rfc3339 = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'");

/**
 * Transforma os Parametros numa mensagem PPMP.
 *
 * @param weather Objecto com os dados meteorologicos recolhidos.
 */
public void setParser(Weather weather) {
    parsedMessage
        = "{\n"
        + "  \"content-spec\": \"urn:spec://eclipse.org/unide/measurement-message#v2\", \n"
        + "  \"device\": {\n    \"deviceID\": \"servidorjava\" \n  }, \n"
        + "  \"measurements\": [\n"
        + "    {\n"
        + "      \"ts\": \"\" + rfc3339.format(weather.getDataDados()) + "\", \n"
        + "      \"series\": {\n"
        + "        \"$_time\": [ 0 ], \n"
        + "        \"DirecaoVento\": [ \" + weather.getDirecaoVento() + \" ], \n"
        + "        \"VelocidadeVento\": [ \" + weather.getVelocidadeVento() + \" ], \n"
        + "        \"Temperatura\": [ \" + weather.getTemperatura() + \" ], \n"
        + "        \"Pressao\": [ \" + weather.getPressao() + \" ], \n"
        + "        \"Humidade\": [ \" + weather.getHumidade() + \" ], \n"
        + "        \"Visibilidade\": [ \" + weather.getVisibilidade() + \" ], \n"
        + "        \"NascerSol\": [ \"\" + weather.getNascerSol() + \"\" ], \n"
        + "        \"PorSol\": [ \"\" + weather.getPorSol() + \"\" ] \n"
        + "      } \n    } \n  ] \n}";
}

```

```

}

/**
 * Transforma os Parametros numa mensagem PPMP.
 *
 * @param historico Objecto com o conjunto de dados de meteorologicas
 * recolhidos.
 */
public void setParser(Historico historico) {
    parsedMessage
        = "{\n"
        + "  \"content-spec\": \"urn:spec://eclipse.org/unide/measurement-message#v2\", \n"
        + "  \"device\": {\n    \"deviceId\": \"servidorjava\" \n  }, \n"
        + "  \"measurements\": [\n"
        + "    {\n"
        + "      \"ts\": \"\" + rfc3339.format(new Date()) + "\", \n"
        + "      \"series\": {\n"
        + "        \"$_time\": \" + historico.getDataDados().toString() + ", \n"
        + "        \"DirecaoVento\": \" + historico.getDirecaoVento().toString() + ", \n"
        + "        \"VelocidadeVento\": \" + historico.getVelocidadeVento().toString() + ", \n"
        + "        \"Temperatura\": \" + historico.getTemperatura().toString() + ", \n"
        + "        \"Pressao\": \" + historico.getPressao() + ", \n"
        + "        \"Humidade\": \" + historico.getHumidade().toString() + ", \n"
        + "        \"Visibilidade\": \" + historico.getVisibilidade().toString() + ", \n"
        + "        \"NascerSol\": \" + historico.getNascerSol().toString() + ", \n"
        + "        \"PorSol\": \" + historico.getPorSol().toString() + "\n"
        + "      } \n    } \n  ] \n}";
}

/**
 * Transforma os Parametros numa mensagem PPMP.
 *
 * @param ts timestamp do alerta
 * @param code codigo do alerta
 * @param title Titulo do alerta
 * @param description Descrição do alerta
 */
public void setParser(Date ts, String code, String title, String description) {
    parsedMessage
        = "{\n"
        + "  \"content-spec\": \"urn:spec://eclipse.org/unide/measurement-message#v2\", \n"
        + "  \"device\": {\n    \"deviceId\": \"servidorjava\" \n  }, \n"
        + "  \"messages\": [\n"
        + "    {\n"
        + "      \"ts\": \"\" + rfc3339.format(ts) + "\", \n"
        + "      \"code\": \"\" + code + "\", \n"

```

```

        + "        \"title\": \"\" + title + "\",\n"
        + "        \"description\": \"\" + description + "\"\n"
        + "    }\n    }\n  }\n";
    }

/**
 * Transforma os Parametros numa mensagem PPMP.
 *
 * @param registoKPI Objecto com o conjunto de dados de KPI's recolhidos.
 */
public void setParser(RegistoKPI registoKPI) {
    parsedMessage
        = "{\n"
        + "  \"content-spec\": \"urn:spec://eclipse.org/unide/measurement-message#v2\", \n"
        + "  \"device\": {\n    \"deviceId\": \"servidorjava\"\n  },\n"
        + "  \"measurements\": [\n"
        + "    {\n"
        + "      \"ts\": \"\" + rfc3339.format(new Date()) + "\",\n"
        + "      \"series\": {\n"
        + "        \"$_time\": \" + registoKPI.getTime().toString() + ",\n"
        + "        \"Uptime\": \" + registoKPI.getUptime().toString() + ",\n"
        + "        \"Total Clientes\": \" + registoKPI.getTotalClients().toString() + ",\n"
        + "        \"Clientes Ativos\": \" + registoKPI.getActiveClients().toString() + ",\n"
        + "        \"Mensagens\": \" + registoKPI.getMessages().toString() + ",\n"
        + "        \"Subscrições\": \" + registoKPI.getSubscriptions() + ",\n"
        + "        \"Carga do sistema nos últimos 5 minutos\": \" + registoKPI.getReceiv\n"
        + "        \"Carga do sistema nos últimos 15 minutos\": \" + registoKPI.getReceiv\n"
        + "        \"Bytes enviados nos últimos 15 minutos\": \" + registoKPI.getBytesSer\n"
        + "      }\n    }\n  ]\n}";
    }

/**
 * Transforma os Parametros numa mensagem PPMP.
 *
 * @param erro Objecto com o conjunto de dados de erros recolhidos.
 */
public void setParser(GestorErros erro) {
    parsedMessage
        = "{\n"
        + "  \"content-spec\": \"urn:spec://eclipse.org/unide/measurement-message#v2\", \n"
        + "  \"device\": {\n    \"deviceId\": \"servidorjava\"\n  },\n"
        + "  \"measurements\": [\n"
        + "    {\n"
        + "      \"ts\": \"\" + rfc3339.format(new Date()) + "\",\n"
        + "      \"series\": {\n"
        + "        \"$_time\": \" + erro.getTime().toString() + ",\n"

```

```

+ "        \"Erros\": \" + erro.getErros().toString() + \"\n\"
+ "    }\n    }\n  ]\n}";
}
}

```