

## How to write Arden Syntax MLMs: SIRS as an example

Medical Knowledge Development

Medexter Healthcare

Borschkegasse 7/5

A-1090 Vienna, Austria

[www.medexter.com](http://www.medexter.com)

September 9, 2016

The following instructions show how to implement automated notifications if a systemic inflammatory response syndrome (SIRS) is detected in a patient.

The definition of the systemic inflammatory response syndrome (SIRS) is as follows (or similar):

### **SIRS Notification**

ALERT if  $\geq 2$  Criteria

- Temperature  $> 38^{\circ}\text{C}$  ( $100.4^{\circ}\text{F}$ ) or  $< 36^{\circ}\text{C}$  ( $96.8^{\circ}\text{F}$ )  
and/or
- Heart rate  $> 90$  beats per minute  
and/or
- Respiratory rate  $> 20$  breaths per minute or arterial carbon dioxide tension ( $\text{PaCO}_2$ )  $< 32$  mm Hg  
and/or
- white blood cell count ( $>12,000/\mu\text{L}$  or  $< 4,000/\mu\text{L}$  or  $>10\%$  immature [band] forms)

**Aim:** If the definition above is met by patient data, an alert should be generated automatically.

**Method:** Arden Syntax medical logic modules (MLMs)

**Requirements:** A Database with two tables, an ArdenSuite IDE, ArdenSuite Server with DB Connector, a REST client. Patient data is made available.

In order to illustrate the possible levels of complexity in writing Arden Syntax MLMs, we will show 6 variants for completing the task. The provided MLMs (SIRS Notification 1-6) progress from a basic version (SIRS Notification 1) to a more complex, feature-filled version (SIRS Notification 6). The 6 MLMs differ in terms of receiving patient data, database interaction, and avoiding overalerting.

Starting with the first MLM version, specific features are either changed or added from one version to the next:

### **SIRS Notification 1**

This MLM checks the 4 SIRS notification criteria and sends an alert if 2 or more criteria are met. It uses 6 input parameters selected from a database with only one entry for each patient id (IDPatient). All laboratory limits are included in the respective if-statements.

### **SIRS Notification 2**

Instead of including laboratory limits directly in the if-statements, this MLM stores all limits in variables created in the data slot.

### **SIRS Notification 3**

In addition to storing the laboratory limits as variables in the data slot, this MLM also uses a database table with multiple entries (different date and/or time) for each patient ID (IDPatient). The MLM reads all entries with the same patient ID but uses only the most recent entry for evaluation.

### **SIRS Notification 4**

Besides checking for the most recent patient ID entry, this MLM also checks for overalerting. If the definition for sending a SIRS alert is met by the most recent data, the MLM checks all entries of the last 24 hours. If one of these entries also meets the SIRS criteria, no alert is sent. This ensures that the MLM sends not more than one alert per 24 hours.

### **SIRS Notification 5**

Whereas in the previous MLM versions the patient ID was defined in the MLM's data slot, this MLM receives the patient ID through the REST call. The patient ID is used to gather entries from the database.

## SIRS Notification 6

This MLM uses no database. All necessary values are being sent along with the REST call.

### SIRS Notification 1 and SIRS Notification 2

The 6 provided MLMs have the suffix .mlm. They can be adjusted using the ArdenSuite IDE. Before sending the REST call, the MLMs have to be compiled with the ArdenSuite IDE (compiled MLMs have the suffix .gz) and uploaded to the ArdenSuite Server. For simply opening .mlm files to see their content, you can also use a standard text editor.

## Create Database

The first step is to create a database with patients and values. The database for the SIRS notification can either be a new database or just 2 new tables within an already existing database. The name of the database and the table names can be chosen freely. The name of the table however has to be the same as in the MLM's SQL-Statement. The database name is required for setting up the database connection with the ArdenSuite Server's Database Connector (DB Connector).

The following database table is the one used for the provided MLMs:

Database name: SIRSDB

Table name: values12

IDPatient	temperature	heartRate	respRate	PaCO2	WBcellCount	immatureBand
123	39	89	23	33	3965	16
124	37	92	15	29	11194	9
125	35	82	21	32	13652	10

## Connect Database to ArdenSuiteServer

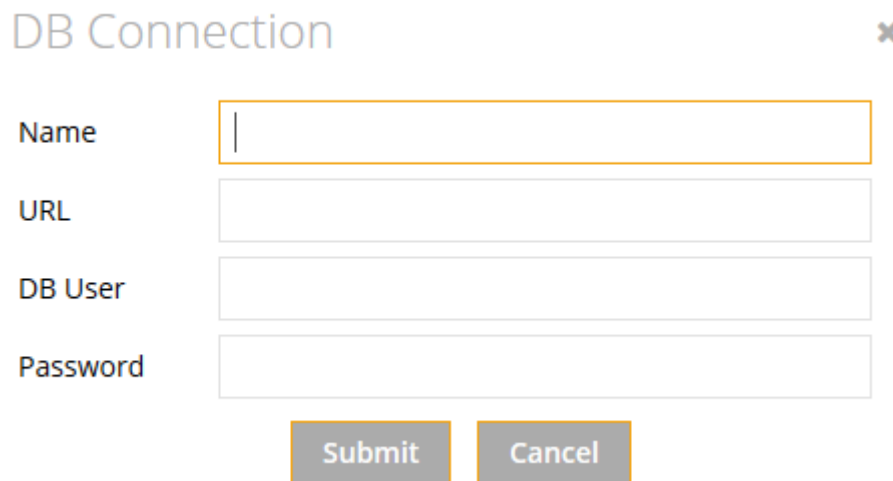
The next step is to connect the database with the ArdenSuite Server. Start the ArdenSuite Server and create the connection in the DB Connector tab. Click on the *Add Connction* button on the bottom left to open the window shown below. Choose a name for your connection. Use the URL, DB User, and password from your created database:

Name: Connection name

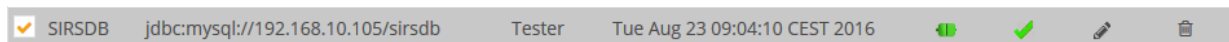
URL: URL of database and table (e.g., jdbc:mysql://IPAdressOfDataBase/TableName)

DB User: Your database username

Password: Corresponding database password



Afterwards you will see the connection in the list of database connections. Activate the connection by clicking on the respective symbol. Then it should look like this:



## Define REST Call

The last step is to trigger the MLM with a REST call. You can use any REST client. Here, the *RESTClient* addon for Firefox will be used as an example:

<https://addons.mozilla.org/de/firefox/addon/restclient/?src=search>

In order to send a correct REST request, the following headers and body are required. The request has to be sent with the method "POST".

Header:

Authorization: username and password of the ArdenSuite Server

Accept: application/json

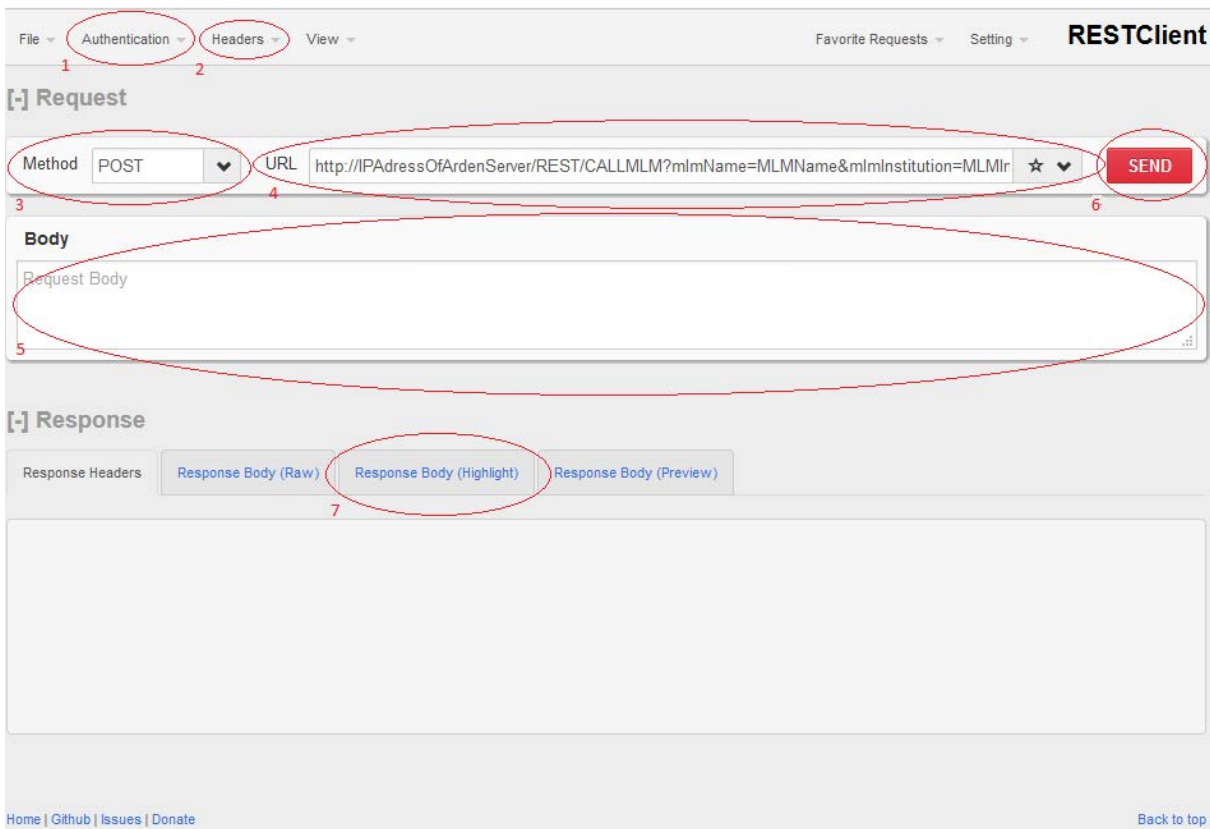
Content-Type: application/json

Body:

```
{  
  "type": "number",  
  "primarytime": null,  
  "applicability": 1,  
  "value": 0  
}
```

(Which number you put in "value" is irrelevant for MLM versions 1-4, but it has to be filled.)

The creation of a REST request is shown below as a step-by-step instruction using the RESTClient addon:



1. Click *Authentication* and select *Basic Authentication*. Insert the username and the password of your ArdenSuite Server.
2.
  - a. Click *Headers* and select *Custom Header*. Insert the following:  
Name: Accept  
Value: application/json
  - b. Click again *Headers* and select *Custom Header*. Insert the following:  
Name: Content-Type  
Value: application/json
3. Select the method „POST“

4. Insert the URL with the IP address of the ArdenSuite Server, the name of the corresponding MLM and the institution (change the parts in blue):

http://*IPAddressOfArdenServer*/rest/CALLMLM?mlmName=*MLMName*&mlmInstitution=*MLMInstitution*

5. Insert the following in the body slot:

```
{
  "type": "number",
  "primarytime": null,
  "applicability": 1,
  "value": 0
}
```

6. Click *SEND*.
7. This is where the response will be displayed. This part is not visible before clicking *SEND*.

### SIRS Notification 3 and SIRS Notification 4

For MLM version 3 and the following versions, a new database table with multiple entries per patient is required.

## Create New Database Table

Create a new table like the one shown below. The table name has to match the name within the MLM. There are no changes necessary concerning the CB Connector or the REST request.

The following database table is the one used for the provided MLMs:

Database name: SIRSDB

Table name: values3

IDPatient	Date	temperature	heartRate	respRate	PaCO2	WBcellCount	immatureBand
123	2016-08-22 09:30:27	39	89	23	33	3965	16
123	2016-08-22 10:22:38	37	80	17	35	11256	12
123	2016-08-22 15:29:23	37	110	25	34	13654	8
123	2016-08-21 14:18:22	39	80	16	31	7563	11
124	2016-08-22 13:23:34	35	82	21	32	13652	10
124	2016-08-22 11:50:12	37	85	19	33	7569	11
125	2016-08-22 18:31:24	35	82	21	32	13652	10

## SIRS Notification 5

This MLM version differs from the previous ones only in the way it receives the patient ID. This MLM receives the patient ID through the REST call, whereas in MLM versions 1-4, the patient ID was defined in the MLM's data slot.

## Adjust REST Call

When defining the REST call, only *step 5* is different. Here, change *value* to the [patient ID](#) (e.g., patient ID = [124](#)) in the body slot:

5. Insert the following in the *Body* slot. Change "value" to patient ID.

```
{ "type": "number",  
  "primarytime": null,  
  "applicability": 1,  
  "value": 124  
}
```

File Authentication Headers View Favorite Requests Setting RESTClient

[-] Request

Method POST URL http://IPAddressOfArdenServer/REST/CALLMLM?mlmName=MLMName&mlmInstitution=MLMlr SEND

Body

Request Body

[-] Response

Response Headers Response Body (Raw) Response Body (Highlight) Response Body (Preview)

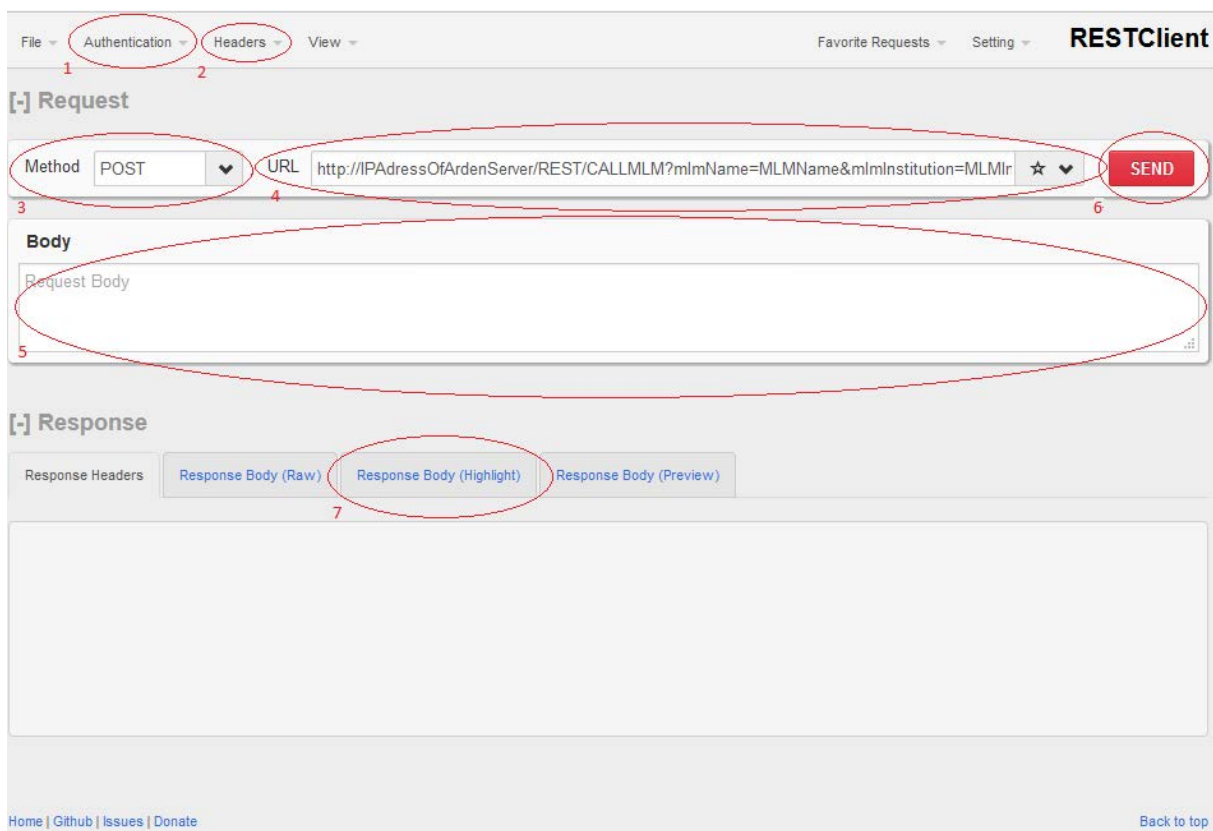
Home | Github | Issues | Donate Back to top

## SIRS Notification 6

This MLM does not use a database. It receives all patient information through the REST call.

### Adjust REST Call

When defining the REST call, again, only *step 5* of the step-by-step instruction above is different. The body has to be changed. Below, 2 examples of the body are provided. The first example has one dataset for one patient, the second example has two datasets (differing in date and/or time) for the same patient.



5. Insert one of the following 2 examples in the *Body* slot. All the patient information is included in the body. "Values" and "primarytimes" can be adjusted:



## REST Body for 1 Dataset:

```
{
  "type": "object",
  "primarytime": "2016-08-23T16:34:57",
  "applicability": 1,
  "declaration": "Patient",
  "fields":
  {
    "temperature":
    {
      "type": "number",
      "primarytime": "2016-08-23T16:34:57",
      "applicability": 1,
      "value": 39
    },
    "heartRate":
    {
      "type": "number",
      "primarytime": "2016-08-23T16:34:57",
      "applicability": 1,
      "value": 100
    },
    "respRate":
    {
      "type": "number",
      "primarytime": "2016-08-23T16:34:57",
      "applicability": 1,
      "value": 21
    },
    "PaCO2":
    {
      "type": "number",
      "primarytime": "2016-08-23T16:34:57",
      "applicability": 1,
      "value": 33
    },
    "WBcellCount":
    {
      "type": "number",
      "primarytime": "2016-08-23T16:34:57",
```

```
"applicability": 1,  
"value": 13000  
  },  
  "immatureBand":  
  {  
    "type": "number",  
    "primarytime": "2016-08-23T16:34:57",  
    "applicability": 1,  
    "value": 11  
  }  
}
```

### REST Body for 2 Datasets:

```
{  
  "type": "list",  
  "primarytime": null,  
  "applicability": 1,  
  "values":  
  [  
    {  
      "type": "object",  
      "primarytime": "2016-08-22T09:30:27",  
      "applicability": 1,  
      "declaration": "Patient",  
      "fields":  
      {  
        "PaCO2":  
        {  
          "type": "number",  
          "primarytime": "2016-08-22T09:30:27",  
          "applicability": 1,  
          "value": 34  
        },  
        "WBcellCount":  
        {  
          "type": "number",  
          "primarytime": "2016-08-22T09:30:27",  
          "applicability": 1,  
          "value": 11  
        }  
      }  
    }  
  ]  
}
```

```
        "value": 7452
      },
      "heartRate":
      {
        "type": "number",
        "primarytime": "2016-08-22T09:30:27",
        "applicability": 1,
        "value": 88
      },
      "immatureBand":
      {
        "type": "number",
        "primarytime": "2016-08-22T09:30:27",
        "applicability": 1,
        "value": 6
      },
      "temperature":
      {
        "type": "number",
        "primarytime": "2016-08-22T09:30:27",
        "applicability": 1,
        "value": 37
      },
      "respRate":
      {
        "type": "number",
        "primarytime": "2016-08-22T09:30:27",
        "applicability": 1,
        "value": 17
      }
    }
  },
  {
    "type": "object",
    "primarytime": "2016-08-22T10:22:38",
    "applicability": 1,
    "declaration": "Patient",
    "fields":
    {
      "PaCO2":
```

```
{
  "type": "number",
  "primarytime": "2016-08-22T10:22:38",
  "applicability": 1,
  "value": 33
},
"WBcellCount":
{
  "type": "number",
  "primarytime": "2016-08-22T10:22:38",
  "applicability": 1,
  "value": 5891
},
"heartRate":
{
  "type": "number",
  "primarytime": "2016-08-22T10:22:38",
  "applicability": 1,
  "value": 110
},
"immatureBand":
{
  "type": "number",
  "primarytime": "2016-08-22T10:22:38",
  "applicability": 1,
  "value": 9
},
"temperature":
{
  "type": "number",
  "primarytime": "2016-08-22T10:22:38",
  "applicability": 1,
  "value": 39
},
"respRate":
{
  "type": "number",
  "primarytime": "2016-08-22T10:22:38",
  "applicability": 1,
  "value": 24
}
```

```
}  
  }  
    }  
  ]  
}
```