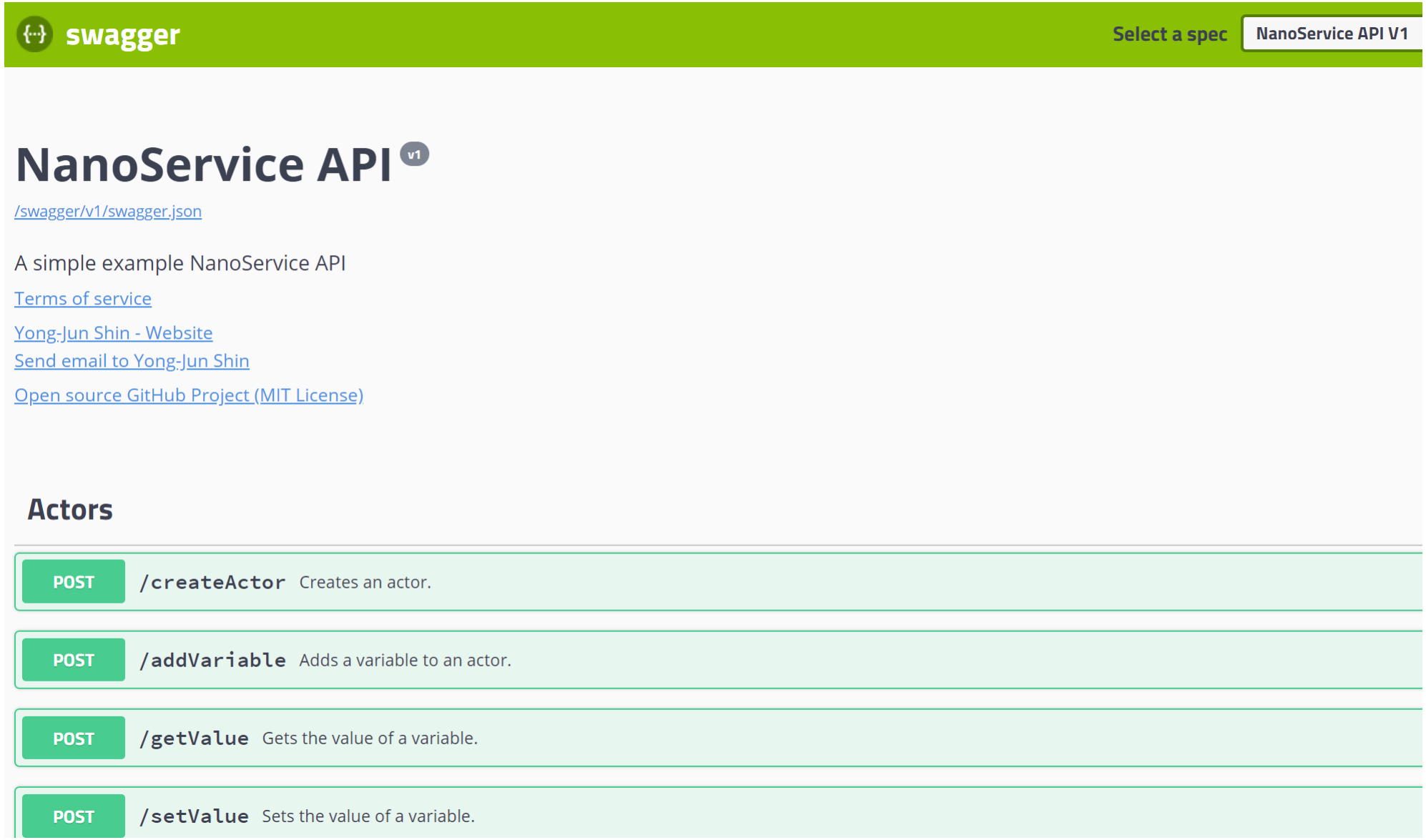
Nanoservice-enabled event-driven clinical decision support REST APIs

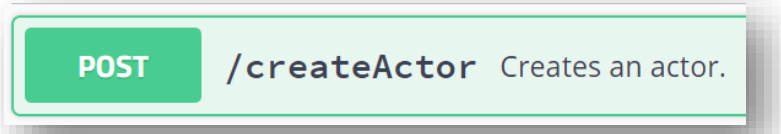
Hyponatremia Diagnosis Demo

For more information about Nanoservice, please visit https://csml.uconn.edu/

Step 1: Go to http://csmlab7.uconn.edu/swagger/



Step 2 [createActor]: Click



swagger

Select a spec NanoService API V1

NanoService API v1

/swagger/v1/swagger.json

A simple example NanoService API

Terms of service

Yong-Jun Shin - Website

Send email to Yong-Jun Shin

Open source GitHub Project (MIT License)

Actors

POST /createActor Creates an actor.

POST /addVariable Adds a variable to an actor.

POST /getValue Gets the value of a variable.

POST /setValue Sets the value of a variable.

Step 3 [createActor]: Follow the instruction and create a patient (actor) with a unique ID (actorId).

Sample request body (requestBody):

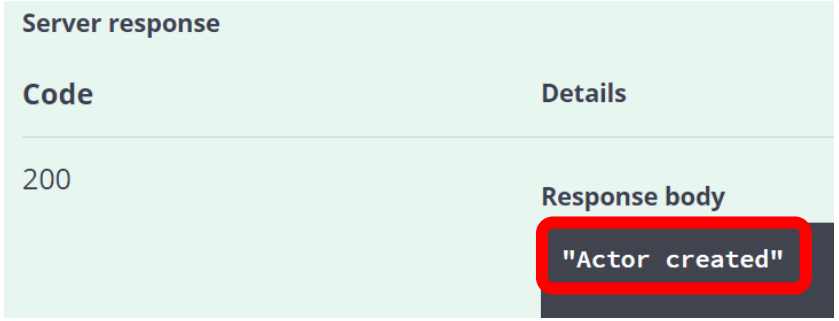
```
{
   "actorId": "patient032904475"
}
```

- "actionId": string

To try this service:

1. Click [Try it out] button (white).
2. Type your request body into “Example Vaule | Model” textbox (white). A sample request body is shown above.
3. Click [Execute] button (blue).
4. Check “Response body” (ignore “Code” for now). If you see "Actor created", your request is processed successfully. Otherwise, you will get an error messsage.

For more information about Service Fabric Actors, please see:
https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-actors-introduction



Service Fabric Actors

https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-actors-introduction

Step 4 [addVariable]: Add a variable (“bloodSodium”) to the previously created patient actor. Let the variable value be “unknown” initially

POST `/addVariable` Adds a variable to an actor.

Sample request body (requestBody):

```
{
   "actorId": "patient032904475",
   "variable": "bloodSodium",
   "value": "unknown"
}
```

- "actionId": string
- "variable": string
- "value": string, int, float, bool

To try this service:

1. Click [Try it out] button (white).
2. Type your request body into “Example Vaule | Model” textbox (white). A sample request body is shown above.
3. Click [Execute] button (blue).
4. Check “Response body” (ignore “Code” for now). If you see "Variable added", your request is processed successfully. Otherwise, you will get an error messsage.

Response body

```
"Variable added"
```

Step 5 [getValue]: Get the value of the variable “bloodSodium” which is “unknown”.

POST `/getValue` Gets the value of a variable.

Sample request body (requestBody):

```
{
   "actorId": "patient032904475",
   "variable": "bloodSodium"
}
```

- "actionId": string
- "variable": string

To try this service:

1. Click [Try it out] button (white).
2. Type your request body into “Example Vaule | Model” textbox (white). A sample request body is shown above.
3. Click [Execute] button (blue).
4. Check “Response body” (ignore “Code” for now). If you see the value retrieved, your request is processed successfully. Otherwise, you will get an error messsage.

Response body

```
"unknown"
```

Step 6 [setValue]: Set the value of the variable “bloodSodium” to 109. The unit is mEq/L and not included.

POST `/setValue` Sets the value of a variable.

Sample request body (requestBody):

```
{
   "actorId": "patient032904475",
   "variable": "bloodSodium",
   "value": 109
}
```

- "actionId": string
- "variable": string
- "value": string, int, float, bool

To try this service:

1. Click [Try it out] button (white).
2. Type your request body into “Example Vaule | Model” textbox (white). A sample request body is shown above.
3. Click [Execute] button (blue).
4. Check “Response body” (ignore “Code” for now). If you see "Value set", your request is processed successfully. Otherwise, you will get an error messsage.

Response body

```
"Value set"
```

Note this new value is also published to **Azure Event Grid.**

Step 7 [getValue]: Get the value of the variable “bloodSodium” which is now changed to 109.

POST `/getValue` Gets the value of a variable.

Sample request body (requestBody):

```
{
   "actorId": "patient032904475",
   "variable": "bloodSodium"
}
```

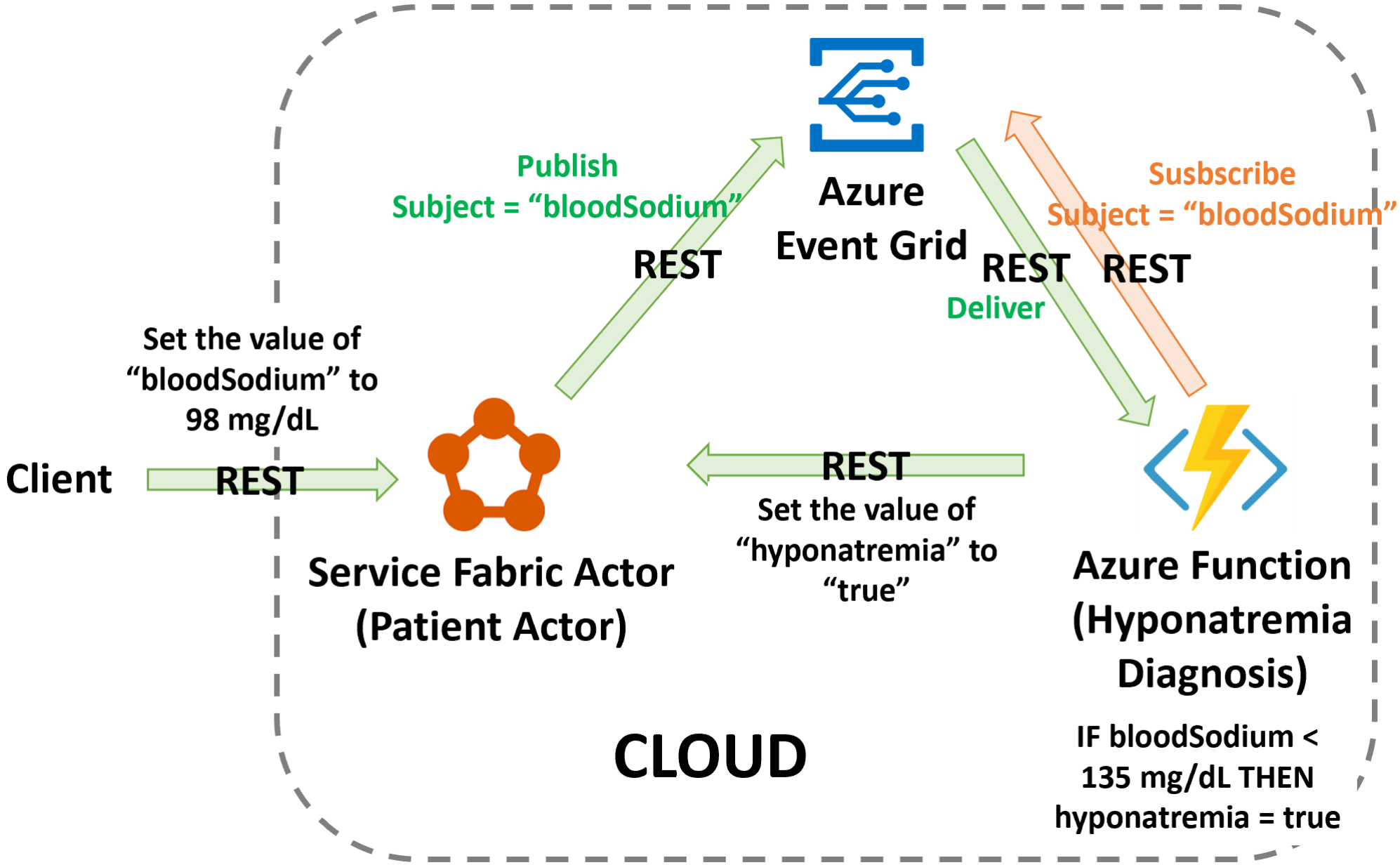
- "actionId": string
- "variable": string

To try this service:

1. Click [Try it out] button (white).
2. Type your request body into “Example Vaule | Model” textbox (white). A sample request body is shown above.
3. Click [Execute] button (blue).
4. Check “Response body” (ignore “Code” for now). If you see the value retrieved, your request is processed successfully. Otherwise, you will get an error messsage.

Response body

```
109
```

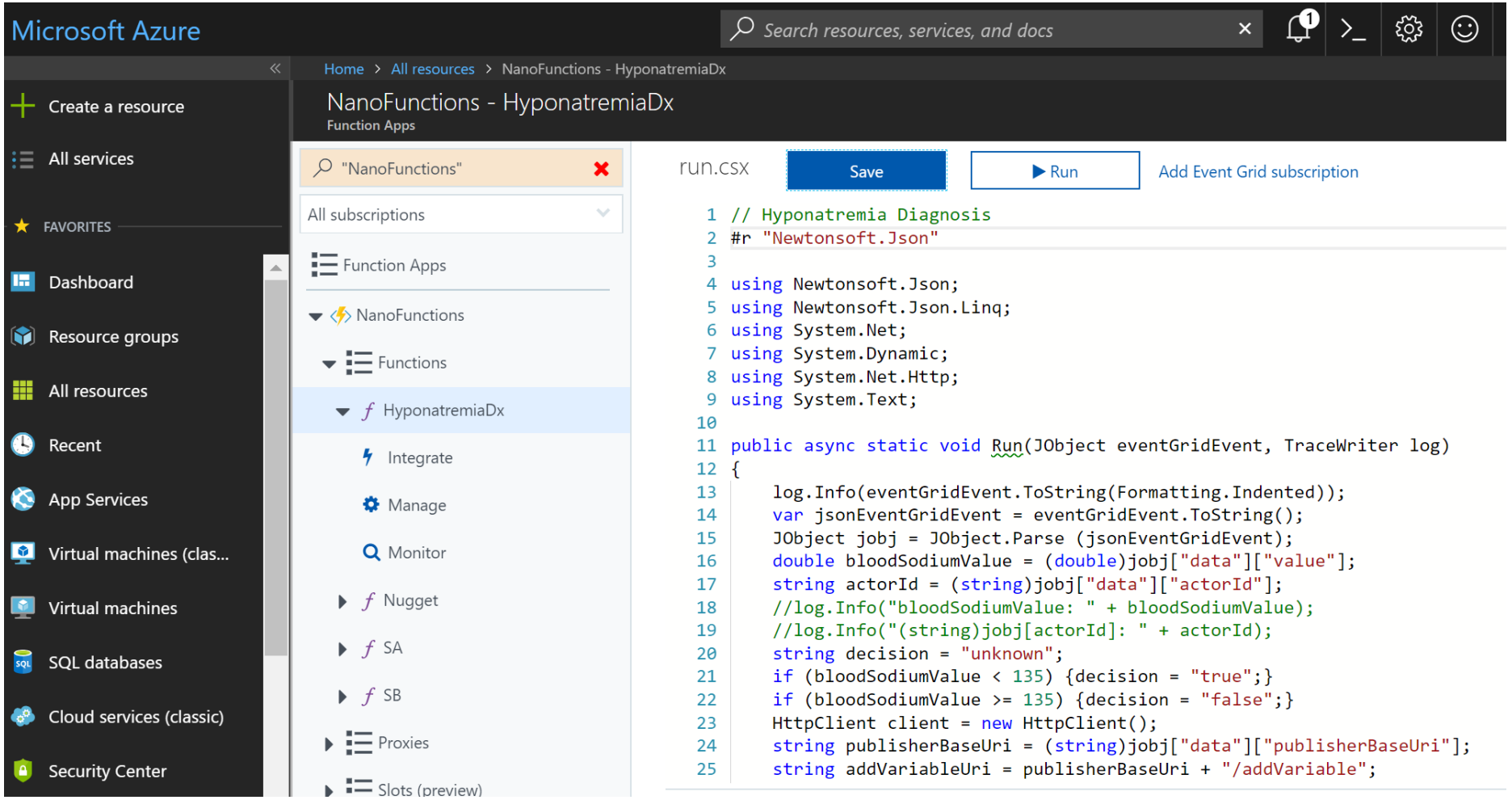

Azure
Event Grid
Publish
Subject = "bloodSodium"
REST
Susbscribe
Subject = "bloodSodium"
REST
REST
Deliver
Set the value of
"bloodSodium" to
98 mg/dL
Client
REST
Service Fabric Actor
(Patient Actor)
REST
Set the value of
"hyponatremia" to
"true"
Azure Function
(Hyponatremia
Diagnosis)
IF bloodSodium <
135 mg/dL THEN
hyponatremia = true
CLOUD

Code for publishing an event to **Azure Event Grid**

```
public async Task<string> PublishToAzureEventGridAsync(JObject data)
{
    string AzureEventGridTopicEndPoint = "https://topic.eastus-1.eventgrid.azure.net/api/events?api-version=2018-01-01";
    string AzureEventGridTopicAccessKey = "9UGRYFbXX3Pqr8yTp2vvhgvNBr8HO0HSWza/PMdxu/0=";
    string uri = AzureEventGridTopicEndPoint;
    string topicSubject = (string)data.SelectToken("variable");

    // Event data schema (Azure Event Grid)
    // https://docs.microsoft.com/en-us/azure/event-grid/post-to-custom-topic#event-data

    dynamic requestBody = new ExpandoObject();
    requestBody.id = "notSet";
    requestBody.eventType = "notSet";
    requestBody.subject = topicSubject; // e.g., bloodSodium
    requestBody.eventTime = DateTime.Now;
    requestBody.data = data;
    requestBody.dataVersion = "v1";

    List<dynamic> requestBodyArray = new List<dynamic>();
    requestBodyArray.Add(requestBody);

    HttpClient client = new HttpClient();
    client.DefaultRequestHeaders.Add("aeg-sas-key", AzureEventGridTopicAccessKey);
    var request = new HttpRequestMessage(HttpMethod.Post, uri);
    string jsonRequestBody = JsonConvert.SerializeObject(requestBodyArray);
    request.Content = new StringContent(jsonRequestBody, Encoding.UTF8, "application/json");
    var response = await client.SendAsync(request);
    string stringResponseContent = await response.Content.ReadAsStringAsync();
    return stringResponseContent;
}
```

Source code available at
https://github.com/yshin1209/Nanoservice

Azure Function https://azure.microsoft.com/en-us/services/functions/



This Azure Function (Hyponatremia Diagnosis) subscribes to Azure Event Grid and is triggered whenever a new bloodSodium value is published (new event).

Azure Function https://azure.microsoft.com/en-us/services/functions/

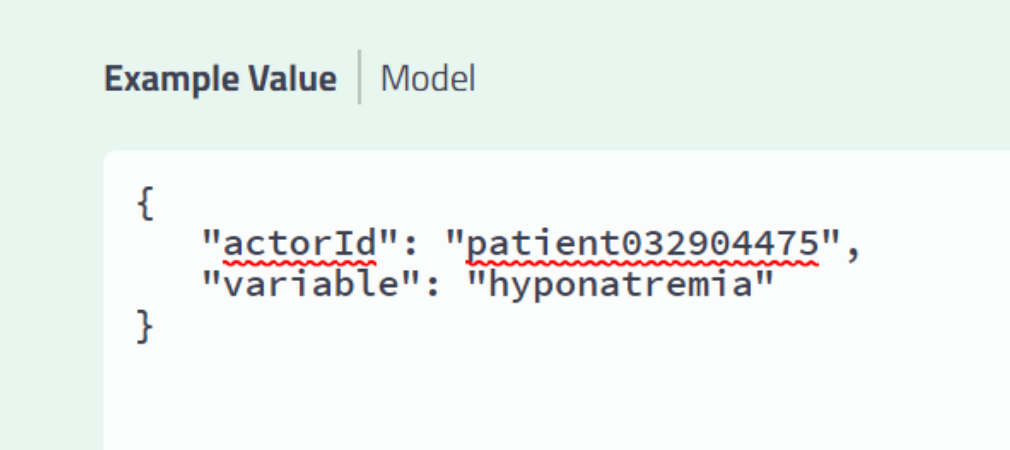
```
string decision = "unknown";
if (bloodSodiumValue < 135) {decision = "true";}
if (bloodSodiumValue >= 135) {decision = "false";}
HttpClient client = new HttpClient();
```

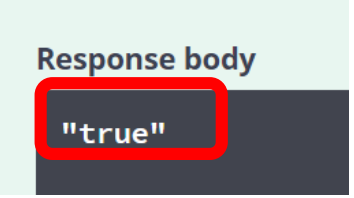
```
string addVariableUri = "http://csmlab7.uconn.edu/addVariable";
var firstRequest = new HttpRequestMessage(HttpMethod.Post, addVariableUri);
dynamic firstRequestBody = new ExpandoObject();
        firstRequestBody.actorId = actorId;
        firstRequestBody.variable = "hyponatremia";
```

```
string setValueUri = "http://csmlab7.uconn.edu/setValue";
var secondRequest = new HttpRequestMessage(HttpMethod.Post, setValueUri);
dynamic secondRequestBody = new ExpandoObject();
```

The HyponatremiaDDx function 1) decides whether it is hyponatremia or not, 2) adds a new variable “hyponatremia” to the patient actor in case the variable is not yet added, and 3) sets the value to true (hyponatremia) or false (not hyponatremia).

Step 8 [getValue]: Get the value of the variable “hyponatremia” which is now “true” (because 109 < 135)





Step 9 [setValue]: Set the value of the variable “bloodSodium” to 235.

Example Value | Model


```
{
   "actorId": "patient032904475",
   "variable": "bloodSodium",
   "value": 235
}
```


Step 10 [getValue]: Get the value of the variable “hyponatremia” which is now “false” (because 235 > 135)

Response body


```
"false"
```

Step 11 [removeVariable]: Remove the variable “bloodSodium”.

POST `/removeVariable` Removes a variable from an actor.

Sample request body (requestBody):

```
{
   "actorId": "patient032904475",
   "variable": "bloodSodium"
}
```

- "actionId": string
- "variable": string

To try this service:

1. Click [Try it out] button (white).
2. Type your request body into “Example Vaule | Model” textbox (white). A sample request body is shown above.
3. Click [Execute] button (blue).
4. Check “Response body” (ignore “Code” for now). If you see "Variable removed", your request is processed successfully. Otherwise, you will get an error messsage.

Response body

```
"Variable removed"
```

For more information about Nanoservice, please visit https://csml.uconn.edu/