Post Resource Status API

Overview

Juvare provides the Post Resource Status application programming interface (API) as a web service that its clients and partners can use to enhance interoperability by updating resource status data in EMResource. Data is communicated using XML for SOAP Web Service or JSON for REST Web Service.

Benefits

The Post Resource Status API offers numerous benefits, including:

- **Security** – Data is communicated securely using the TLS 1.2 protocol.
- **Flexibility** – The API is designed to update virtually any status data for any set of resources in EMResource.
- **Configurability** – The API is configured based on the requirements and information you specify.

Terms

In relation to the Post Resource Status API document and the EMResource solution, there are some common terms that are helpful to keep in mind.

- **Resource** – An entity that reports a status. Resources can be facilities, organizations, and agencies that report status information on capabilities, services, and supplies.
- **Status Type** – A type of status that resources record. Status types can be multiple choice, dates, numbers, scores, or text. For example, Status Types include the Number of Available Beds, ED Status, Contact Name, and NEDOCS Score.
- **Status** – A value or option that resources report for a status type. For example, Statuses include Open, Closed, or Limited.
- **Status Reason** – The reason specified for making a status change. They are applicable to some multiple-choice Status Types. There is a predefined list of valid Status Reasons for each applicable Status. For example, Status Reasons include ED overflow, surgeon not available, and no ICU beds.
- **Comment** – Free text that can be entered to supplement the status value and reason.

Documentation

Juvare recommends using this document in conjunction with the details and procedures in the *Partner Interoperability Development Guide*. The interoperability guide describes web service interfaces and provides an example procedure for connecting to Juvare Web Services using a .NET Web Services' client.

Implementation

Step 1: Pick SOAP or REST

Step 2: Configuration

Working in conjunction with Juvare, define the scope of resources and status types you want to include in the interface. Juvare will configure the interface and provide you with a URL, action name, division name, and the credentials needed to make web service requests.

Step 3: Web Service Code and Connection

Write the code for your side of the interface, referring to the example code and procedures available in the *Partner Interoperability Development Guide*. Make sure the code can connect to and invoke the Post Resource Status API.

Invoking the API requires a single web service call. You develop the code needed to create the XML (SOAP) or JSON (REST) request and process the response that the web service returns to you.

With appropriate web service permissions, EMResource updates the status for all values that changed. If you do not have permission to update a specific status or the value has not changed, EMResource does not update the status.

EMResource processes all authorized and valid updates and skips the ones that are not. The response includes a report of which ones succeeded and an error message for each failure.

Technical Support

Juvare supports your development efforts by providing the *Partner Interoperability Development Guide*, technical documentation on the specific API, the XML schema document, and examples of request and response statements. We are also able to assist you with testing the interface.

Setup

Values used in the API setup are unique to you, the client, and must be obtained from Juvare. These values include:

- Web service parameters (actionName and divisionName)
- Web service user credentials (username and password)

- Test user credentials (username and password)
- Resource IDs (for example, hospital A, B, and C, and clinic 1)
- Status Types
    - Status Values and Status Reasons for each applicable Status Type

SOAP Web Service

Gateway Partner Endpoints

- Demo/Test: https://emresource.demo.juvare.com/partnergateway/PartnerEndpoint
- Production: https://emresource.juvare.com/partnergateway/PartnerEndpoint

Update Status Type

Request Header

```
<requestDefinition actionName='{Action Name}' systemName='EMRESOURCE'
                    divisionName='{Division Name}'
                    username='{Username}' password='{Password}'>
</requestDefinition>
```

Request Body

```
   <resources>
     <resource id='{Resource ID}'>
       <statuses>
         <status>
               <statusName>{Status Name}</statusName>
               <statusValue>{Status Value}</statusValue>
         </status>
         <status>
               <statusName>{Status Name}</statusName>
               <statusValue>{Status Value}</statusValue>
               <comments>{Comment}</comments>
               <reasons>
                     <reason>{Reason}</reason>
               </reasons>
       </status>
       </statuses>
      </resource>
</resources>
```

Response Body

```
<response>
      <exposition>
         <resource id="{Resource ID}">
```

```
            <details>
               <errors>
                     <error id="externalID">{Error Message}</error>
               </errors>
               <totalSent>0</totalSent>
               <totalUpdated>0</totalUpdated>
            </details>
            <statuses>
               <errors/>
               <totalSent>0</totalSent>
               <totalUpdated>0</totalUpdated>
            </statuses>
         </resource>
      </exposition>
      <summary>
         <resourceFailures>1</resourceFailures>
         <resourceSuccesses>0</resourceSuccesses>
         <statusFailures>0</statusFailures>
         <statusSuccesses>0</statusSuccesses>
      </summary>
</response>
```

Request and Response Examples

You need to develop code to create the XML header and body.

Request Header Example

```
<requestDefinition actionName='postResourceStatus'
systemName='EMRESOURCE' divisionName='DivisionABC'>
</requestDefinition>
```

Request Body Example

```
<resources>
  <resource id='976eb6c3-xxcd-4b55-ae94-4a25bf8d17d6'>
    <statuses>
     <status>
            <statusName>Avail-AdultICU</statusName>
            <statusValue>111</statusValue>
            </status>
     <status>
            <statusName>Avail-NICU</statusName>
            <statusValue>222</statusValue>
            </status>
     <status>
            <statusName>Avail-Pediatric</statusName>
            <statusValue>333</statusValue>
```

```
      </status>
                  <status>
            <statusName>Avail-PICU</statusName>
            <statusValue>444</statusValue>
      </status>
     <status>
            <statusName>Avail-VentsTotal</statusName>
            <statusValue>555</statusValue>
      </status>
</statuses>
  </resource>
  <resource id='abceb6c3-xxcd-4b55-ae94-4a25bf8d17d6'>
    <statuses>
     <status>
            <statusName>Avail-AdultICU</statusName>
            <statusValue>11</statusValue>
      </status>
     <status>
            <statusName>Avail-NICU</statusName>
            <statusValue>22</statusValue>
      </status>
     <status>
            <statusName>Avail-Pediatric</statusName>
            <statusValue>33</statusValue>
      </status>
                  <status>
            <statusName>Avail-PICU</statusName>
            <statusValue>44</statusValue>
      </status>
     <status>
            <statusName>Avail-VentsTotal</statusName>
            <statusValue>55</statusValue>
      </status>
      <status>
            <statusName>EmergencyDepartment</statusName>
            <statusValue>On Divert</statusValue>
            <comments>Updating status and reasons</comments>
            <reasons>
                  <reason>Surge-InPlace</reason>
                  <reason>Surge-AlternateCare</reason>
            </reasons>
      </status>
      </statuses>
  </resource>
```

Response Body Example

The HTTP response contains a summary of how much of your request succeeded, and an error message for each failure. You need to develop code to parse and process the XML.

```
<response>
   <exposition>
         <resource id="4f64566aa3744fc8993c0a7a049bb3e5" name="My
                  Hospital" region="Region1">
               <details>
                     <errors/>
                     <totalSent>0</totalSent>
                     <totalUpdated>0</totalUpdated>
               </details>
               <statuses>
                     <errors>
                           <error id="Invalid-Key">Invalid-Key=-Unknown-; User
                           may not update status</error>
                     </errors>
                     <totalSent>3</totalSent>
                     <totalUpdated>1</totalUpdated>
               </statuses>
         </resource>
   </exposition>
   <summary>
         <resourceFailures>0</resourceFailures>
         <resourceSuccesses>1</resourceSuccesses>
         <statusFailures>2</statusFailures>
         <statusSuccesses>1</statusSuccesses>
   </summary>
</response>
```

REST Web Service

Gateway Partner Endpoints

- Demo/Test: https://emresource.demo.juvare.com/partnergateway/serviceendpoint/{ActionName}/{DivisonName}
- Production: https://emresource.juvare.com/partnergateway/serviceendpoint/{ActionName}/{DivisonName}

Update Status Type

Request Header

```
Content-Type: application/json
Accept: application/json
```

Request Authorization

```
Basic Auth
```

Request Body

```
{
    "resources": [
        {
            "id": "{Resource ID}",
            "statuses": [
                {
                    "statusName": "{Status Name}",
                    "statusValue": "{Status Value}",
                    "comments": "{Comments}"
                },
               {
                    "statusName": "{Status Name}",
                    "statusValue": "{Status Value}",
                    "comments": "{Comment}",
                    "reasons": {
                        "reason": [
                              "{Reason}"
                        ]
                  }
                }
            ]
        }
    ]
}
```

Response

```
{
    "summary": {
        "resourceSuccesses": 1,
        "resourceFailures": 0,
        "statusSuccesses": 1,
        "statusFailures": 0
    },
    "exposition": {
        "resources": [
            {
                "id": "{Resource ID}",
                "name": "{Facility Name}",
                "regionName": "{Region Name}",
                "details": {
                    "totalSent": 0,
                    "totalUpdated": 0,
                    "errors": [
                        {
                              "id": "{externalID}",
                              "errorMessage": "{error message}"
                        }
                  ]
                },
                "statusDetails": {
                    "totalSent": 0,
                    "totalUpdated": 0,
                    "errors": [
                        {
                              "id": "{externalID}",
                              "errorMessage": "{error message}"
                        }

                  ]
                }
            }
        ]
    }
}
```

Request and Response Examples

You need to submit a request to update data in EMResource.

Request URL Example

```
URL:
https://emresource.demo.juvare.com/partnergateway/serviceendpoint/postResourceStatus/DivisionABC
Method: POST
```

Request Header Example

```
Accept: application/json
Content-Type: application/json
Authorization: Basic VVNFUk5BTUU6UEFTU1dPUkQ=
```

The configured web service validates the request, retrieves the data, and delivers it in JSON.

Request Body Example

You need to develop and use a processing code to parse and process the JSON.

```
{
    "resources": [
        {
            "id": "bdba43cf024d41769dfe9ecfb2b290c1",
            "statuses": [
                {
                    "statusName": "EmergencyDepartment",
                    "statusValue": "On Divert",
                    "comments": "Json Update",
                    "reasons": {
                        "reason": [
                            "Surge-InPlace",
                            "Surge-AlternateCare"
                        ]
                    }
                }
            ]
        }
    ]
}
```

Response Body Example

The HTTP response contains a summary of how much of your request succeeded, and an error message for each failure. You need to develop code to parse and process the JSON.

```
   {
       "summary": {
           "resourceSuccesses": 1,
           "resourceFailures": 0,
           "statusSuccesses": 1,
           "statusFailures": 0
       },
       "exposition": {
           "resources": [
               {
                   "id": "bdba43cf024d41769dfe9ecfb2b290c",
                   "name": "IntegrationTestResource",
                   "updateType": "null",
                   "regionName": "IntegrationTest",
                   "details": {
                       "totalSent": 0,
                       "totalUpdated": 0,
                       "errors": [
                           "id": "externalID",
                                 "errorMessage": "Resource id
bdba43cf024d41769dfe9ecfb2b290c is invalid or not configured for this interface."
                     ]
                   },
                   "statusDetails": {
                       "totalSent": 1,
                       "totalUpdated": 1,
                       "errors": [
                     "id": "EmergencyDepartment",
                            "errorMessage": "EmergencyDepartment=Emergency
Department; Invalid status value On Diver"
                     ]
                   }
               }
           ]
       }
   }
```

Values

Attribute	Format	Comment
Action Name	Text	Uniquely identifies the web service and its scope
Division Name	Text	
Username	Text	Username for authentication
Password	Text	Password for authentication
Resource ID	UUID	Unique identifier of each resource (for example, facility)
Status Name	Text	Unique identifier of type of status (for example, licensed beds)
Status Value	Text/Number	Value of the status (for example, “555-555-1234” for ED Phone Number, 13 for licensed beds)
Comment	Text	Freeform comments that supplement the status value (optional)
Reason	Text	Reason for Status Value (where applicable)

Verification

After running the API, log in to EMResource using the test user credentials to verify the status changes.

Need Help?

For more information, contact the Juvare Support Center by sending an email to support@juvare.com or by calling 877-771-0911.