

---

# **Sigmify REST API**

## **(Raising Tickets)**

*(A Step by Step Guide on how to raise tickets in Sigmify from your Application/Website)*

---

## Table of Contents

<b>1. Document Version History</b>	3
<b>2. About the Sigmify REST APIs</b>	4
<b>3. Why use the Sigmify REST APIs</b>	4
<b>4. What this guide covers</b>	4
4.1 Collaboration Entities	4
4.2 Technical Components	5
4.3 Steps	6
4.3.1 Configuration Steps	6
4.3.2 Integration Steps	9

## 1. Document Version History

<b>Date</b>	<b>Description</b>	<b>Version</b>
23/05/2017	First version to document APIs for creation of tickets	0.1
25/05/2017	High level diagram to describe flow of Ticket related APIs included	0.2
26/06/2017	Documented updated to include assignment of project rights to the user who is making the REST API call	0.3
31/07/2017	Major update to the document. A lot more details to various components and various points that were brought up during the first integration have been added	0.4

## 2. About the Sigmify REST APIs

Sigmify is moving towards becoming an application framework and to achieve this we have been working on developing various easy-to-use and easy-to-understand REST APIs. These APIs will be made available to the users of registered tenants to seamlessly perform various operations like raising Tickets, creating a Conversation, creating a Contact, etc. These APIs will enable verified tenants to integrate these capabilities into their own application/websites.

## 3. Why use the Sigmify REST APIs

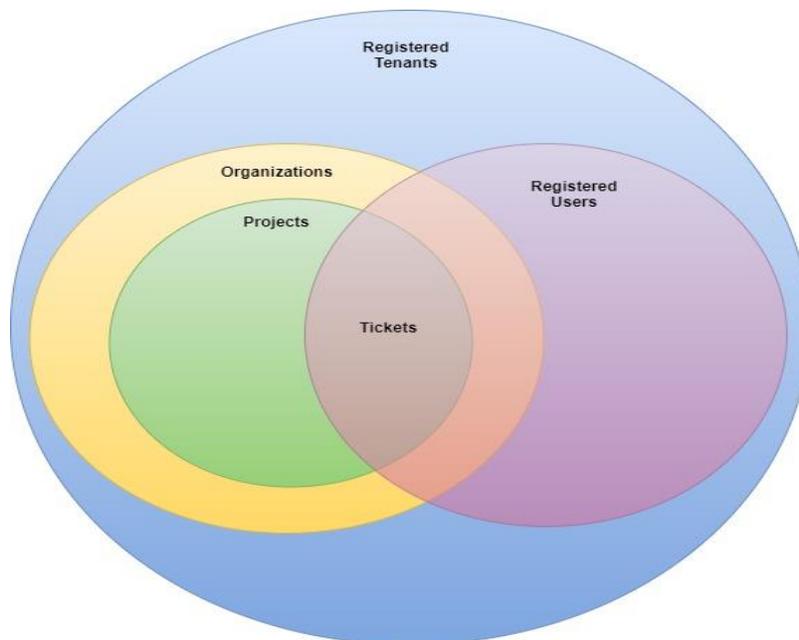
The Sigmify REST APIs can seamlessly integrate with any external application or website. This capability extends the collaboration capabilities provided by Sigmify to all its registered tenants, who can use them to easily implement a collaborative component on an application/website of their choice.

## 4. What this guide covers

This guide specifically explains the collaboration entities, technical components and steps that are used for raising tickets inside of Sigmify.

### 4.1 Collaboration Entities

The following diagram gives a high level view of the various entities that are involved in the “Raise a Ticket” functionality.



**Registered Tenant:** This is any company that subscribes to Sigmify services. Every registered tenant is identified by a unique Tenant ID field that is assigned to it at the time of registration.

**Registered Users:** These are users who get on-boarded on to Sigmify under a given Registered Tenant. A registered user is identified by his/her unique Email ID that was used at the time of on-boarding. The relationship between the registered user and the registered tenant is maintained in Sigmify in a mapping table where the email ID of the registered user gets linked to the Tenant ID (described above).

**Organization:** Organization is the name of the entity (any establishment) that is the tenant's customer. The tenant is supplying its products/providing its services to this entity. An Organization is always associated with a registered tenant (defined above)

**Project:** Project is unique name for the service or product that the tenant is supplying to its customer. A Project is always associated with an Organization (defined above).

**Tickets:** Tickets are tasks/requirements/issues raised by customers with respect to the products/services that the tenant is providing to its customers. A Ticket is always associated with a Project (defined above).

## 4.2 Technical Components

- **REST API:** The REST API is based on the RESTful web services which is a way of providing interoperability between computer systems on the internet. Any further details about this component is beyond the scope of this guide. Please refer to the following links for further details:
  - [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)
  - <http://www.restapitutorial.com/>

*Sigmify uses REST to expose its Raise Ticket function via HTTPS. Tenants, who wish to integrate the Raise Ticket functionality into their application/website can refer to the above links to learn about how the REST API works. The steps and the endpoints for making the REST web service call are described in the sections below.*

- **JSON Format:** JSON (JavaScript Object Notation) is a lightweight data-interchange format. JSON is built on two structures
  - A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
  - An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

Any further details about this component is beyond the scope of this guide. Please refer to the following links for further details:

- <http://www.json.org/>
- [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)

*Sigmify uses JSON for receiving data through the REST API and also to send back responses to the calls. For calling the Raise Ticket REST API, developers need to convert the user- entered data items captured through the UI and some fixed tenant related data items to the JSON format and then feed it to the REST API endpoint. The format and the details of the JSON string is described in the sections below.*

- **Shiro Token:** Apache Shiro is an application security framework that provides application developers very clean and simple ways of supporting four cornerstones of security in their applications: authentication, authorization, enterprise session management and cryptography.

Any further details about this component is beyond the scope of this guide. Please refer to the following link for further details: <https://shiro.apache.org/>.

*Sigmify uses Apache Shiro to associate a Registered User with a REST API call. Before making a REST API call to Raise a Ticket, tenant developers need to design their system to make a “Get Token” call, which basically returns a Shiro Token. This token is then used in the “Raise a Ticket” call to verify the source of the call.*

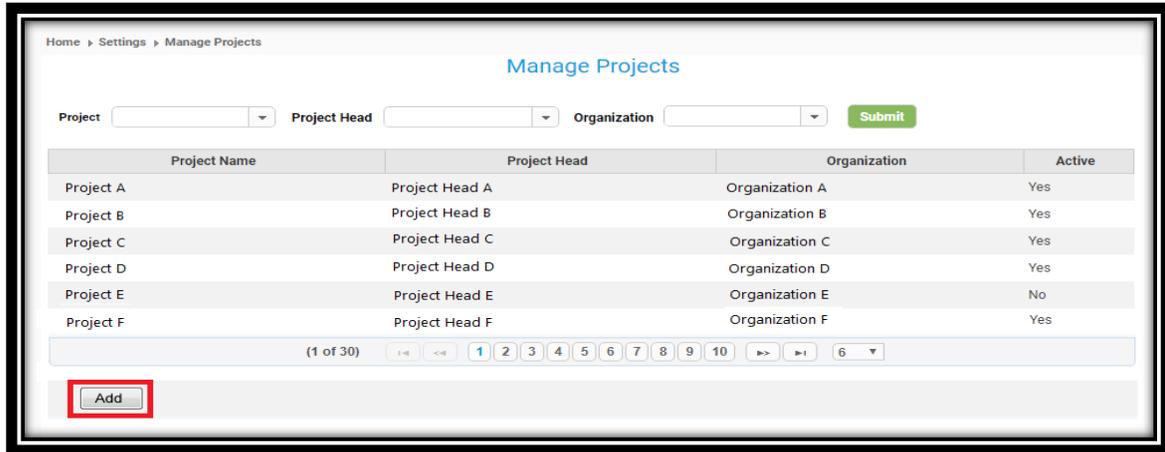
## 4.3 Steps

The following sections cover the configuration and the integration steps that need to be completed before making the REST API call. **Please note that the following steps are applicable only for registered tenants.**

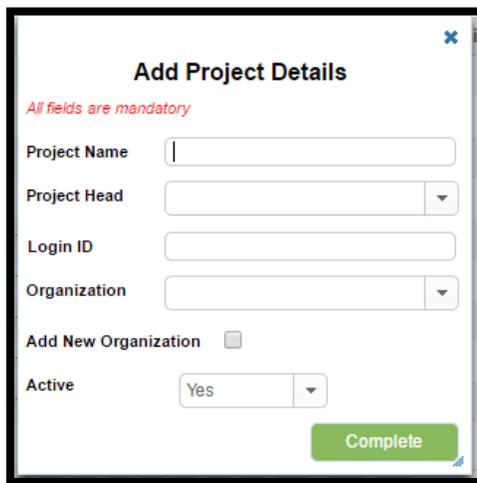
### 4.3.1 Configuration Steps

These steps are for configuring the mandatory fields that will be used during the REST API call. Without configuring these components the REST API call cannot be done for raising tickets.

- **Who?** : The **Tenant Admin User** is responsible for doing these steps. Normal registered users might not have enough rights to perform the steps mentioned below.
- **What?** : The following is the list of components that need to be configured:
  - Organization
  - Project
  - Ticket Priorities
  - Submission User details
- **When?** : These steps should be completed before starting the integration steps as described in section 4.3.2.
- **How?** The following section describes how the configuration steps need to done.
  - ✓ **Configuring the Project and the Organization:**
    - Login into Sigmify.
    - Go to the Settings screen by clicking on the Settings option that appears under the gear icon  on the top right hand side corner of the Sigmify landing page.
    - Click on the Manage Projects link that appears under the “Setup Tickets” section.
    - The following is a screenshot of the Manage Projects screen (please scroll to the next page)



- Click on the Add button highlighted in red above.
- This will bring up the Add Project Details screen.



- This screen can be used for creating a new project and also to add a new Organization to a given tenant. The fields shown in this form are self-explanatory. The Login ID field here gets populated automatically once the Project Head is chosen from the Project head drop down. The login id here is the registered email of the Project Head. The following two values from this screen will be used in the REST API call for raising tickets:
  - Project Name
  - Organization

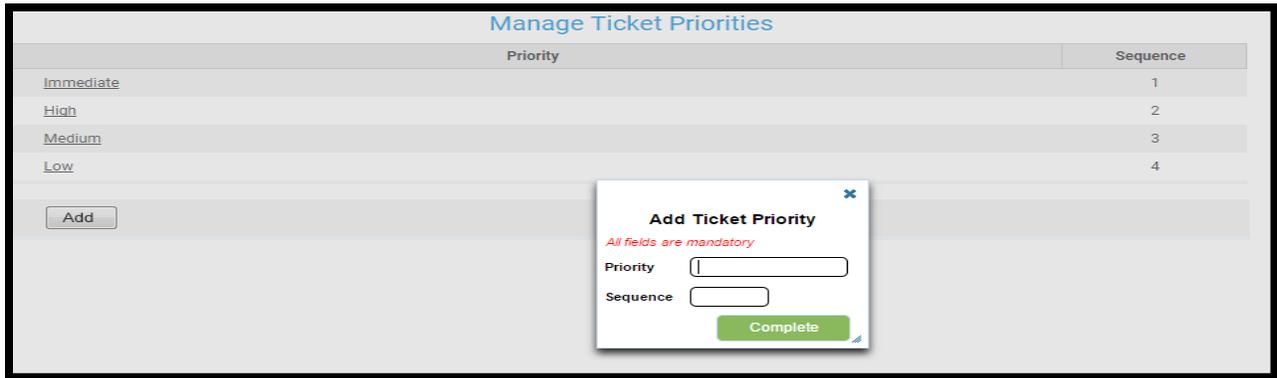
The values from these fields should **exactly match (including case)** the corresponding fields that are used in the REST API call for raising tickets.

Please note that this is an optional step. Tenants can choose to use any existing Organization and/or Project for use in the REST API Call.

✓ **Configuring the Ticket Priorities**

- Login into Sigmify.
- Go to the Settings screen by clicking on the Settings option that appears under the gear icon  on the top right hand side corner of the Sigmify landing page.

- Click on the Manage Ticket priorities link that appears under the “Setup Tickets” section.



- The above is a screen shot of the Manage Ticket Priorities screen. You can add/remove and assign sequences to the priorities are associated with a ticket. This setting will be applicable to all the projects/organizations that are present in the system.

The values set under the priority field can be used for setting the priority of the tickets that are raised through the Sigmify REST API.

The values from the priority field should **exactly match (including case)** the corresponding field that is used in the REST API call for raising tickets.

- ✓ **Submission User details:** This is a registered user that has rights to the Organization and Project configured in the above steps. This login credentials of this user will be used in the REST API call for performing the following steps:
  - Get the Shiro token
  - Make the Raise Issue call.

**Please Note:** The registered user data used for getting the Shiro token should match the registered user data that is being used to make the Raise Issue REST API call.

If you have created Project(s)/Organization(s) and would like to give the Submission User access to these then you can do that from the “Grant Organization and Project Rights to Users” screen. This screen can be accessed by clicking on the “Grant Organization and Project Rights to Users” link which appears under the Setup Tickets on the Sigmify settings screen.

At this point we are done with the configuration related steps and before moving to the Integration steps the Tenant Admin needs to accumulate the following tenant related data and pass it over to the tenant developer

- Organization that needs to be used in the Raise Ticket REST API call.  
Eg: Test Organization
- Project that needs to be used in the Raise Ticket REST API call.  
Eg: Test Project
- List of the Ticket Priorities.  
Eg: Immediate, High, Medium and Low.

d) User name and password of the registered user who will be receiving all the tickets raised by the REST API call on his/her Sigmify Stream.

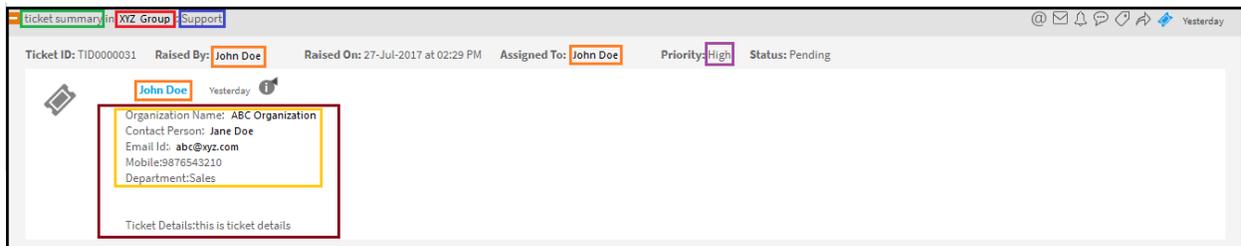
Eg: Username: [abc@xyz.com](mailto:abc@xyz.com) Password: u4An0awa9

### 4.3.2 Integration Steps

The Integration steps described in this section involve the various technical steps that a tenant developer needs to perform on the Tenant Application/Website. Before getting into the details of these steps it's important to understand the basic structure of a Ticket raised inside of Sigmify.

#### 4.3.2.1 Structure of a ticket

The following is the screenshot of a ticket, how it appears on the Sigmify stream. The colored boxes and the color legend are used here only for explaining the ticket components that are affected by the Raise Ticket REST API call.



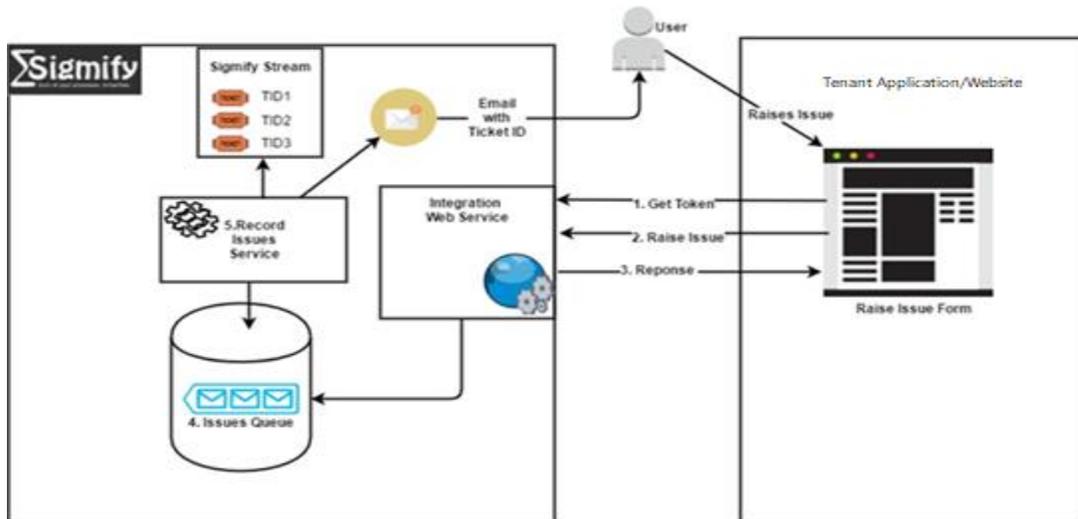
#### 4.3.2.2 Raise ticket REST API mapping with Sigmify Ticket fields.

The following is the list of fields that are used while making the Raise Ticket REST API call. The following table shows the mapping between the ticket components highlighted above and the parameters used in the Raise Ticket REST API call.

#	Ticket Component	Raise Ticket REST API call parameter
1	Ticket Summary	SUMMARY
2	Organization	ORGANIZATION
3	Project	PROJECT
4	Submission User	user
5	Ticket Details	DETAILS
6	Tenant specific custom fields	<p>These fields can be combined together to form a name/value pair strings that are concatenated with each other and having a string newline character (\n) between each set of name/value.</p> <p>Eg: TenantSpecificFields =                      "Name 1:"+"Value1" + "\n"+                      "Name 2:"+"Value2" + "\n"+                      .....</p> <p>The final string then needs to be concatenated to the DETAILS parameter and posted. You can choose to append or prepend the custom field's final string to the DETAILS string based on your requirement. In the example shown in the screenshot above the final string has been prepended to the DETAILS parameter. A couple of newline characters have been added between the final string and the DETAILS string for formatting purposes.</p>
7	Ticket Priority	PRIORITY

### 4.3.2.3 Raise Ticket REST API call - Steps

The following diagram gives a snapshot of the steps that are involved in calling the Sigmify REST API Raise Ticket function.



Now, we will discuss the different integration steps shown in the above diagram.

**Assumptions:** The registered tenant has a “Raise Ticket” form ready on their application/website. The tenant developer is responsible for building this form. There are no restrictions on the contents or the number of fields. Only while submission any extra set of fields that are not a part of the standard Ticket parameter set will have to be submitted in a particular manner. We will discuss these details later in the sections below.

**4.3.2.3.1 Get Token:** This call is used for receiving a Shiro token from the Sigmify server. This Shiro token is then used in the Raise Ticket REST API Call. The Sigmify server identifies and validates the incoming Raise Ticket REST API call using this Shiro Token. Basically when a ticket is submitted from the tenant website the first thing it has to do is to verify it’s identify i.e. the submission is coming from a registered tenant’s website. To do this the tenant developer needs to obtain a user name and password from the tenant Admin that can uniquely identify the tenant. Refer to section 4.3.1 to learn how to configure such a user.

Now, let’s look at the Get Token call in detail:

- **Who?** The tenant developer is responsible for calling this function from their tenant website.
- **What?** The tenant developer needs to build the code to make a HTTPS call (preferably from the server side to avoid cross-domain calls related issues) from the Raise Issue web site. The following is the basic structure of the URL that needs to be called:
  - ❖ <https://<INSTANCE>/getAuthToken?user=<registered-sigmify-user-id>&password=<password>>
 Where:
  - **<INSTANCE>**: is the Sigmify domain where the call needs to go. The Sigmify support team will be responsible for supplying this information. **Note:** For testing purposes the first domain value that will be provided will be of the stage environment. Once the tenant developer has tested and integrated successfully with the stage instance only then will the production instance domain details will be provided.
  - **<registered-sigmify-user-id>**: This is the username of a Registered User who was configured in the system as per steps given in section 4.3.1. Note: The value has to be UTF8 encoded before it is added to the URL string given above for making the call.

- **<password>**: This is the password of a Registered user who was configured in the system as per steps given in section 4.3.1. Note: The value has to be UTF8 encoded before it is added to the URL string given above for making the call.

**Note: The <> brackets are only used here for formatting purposes. Please do not use them while building the three parameters discussed above.**

- **When?** The call mentioned above needs to be done exactly after the users of the Tenants application/website press the submit button after entering the issue (from the tenants application/website).  
**Note:** The Shiro token issued by the Sigmify is valid for 8 hours or till the Sigmify application server is refreshed/restarted. The tenant developers can choose to cache the Shiro Token made by the first call and use it for the next 8 hours or they can choose to get a new token for each submission. There is no restriction for this step.
- **How?** Sample codes on how to make this call is available on request. If you need any assistance with making this call. Please get in touch with the Sigmify support group ([support@sigmify.com](mailto:support@sigmify.com))

Once you are able to make this call successfully, you should expect to get back a HTTPS web response in the following JSON format:

Sample Shiro token response

```
{
  "token":"0cbda5ea-e552-4c92-884b-f491831xx3f0",
  "firstName":"<first-name-of-registered-sigmify-user>",
  "lastName":"<last-name-of-registered-sigmify-user>",
  "userLoginId":"<registered-sigmify-user-id>",
  "message":"OK"
}
```

**4.3.2.3.2 Raise Ticket:** This is the core function that is used for submitting the ticket related details (captured from the registered tenant’s website). Unlike the Get Token function call which is a simple server side HTTPS call, the Raise Ticket function call involves a bit more technical specifications to be followed.

In the following section we will look at this function in detail:

- **Who?** The tenant developer is responsible for implementing this function on the tenant website.
- **What?** This is the core function that transfers all the ticket related details from the tenant’s website to the Sigmify website. There are four important components of this call:
  - **Json String payload:** The data that is captured from the registered tenant’s website has to be arranged in a certain format before it can be embedded into the Raise Ticket API call. The following is the format of the string (json structure):
 

```
{
            "ORGANIZATION":"","
            "PROJECT":"","
            "PRIORITY":"","
            "SUMMARY":"","
            "DETAILS":"","
            "user":"user@bisil.com"
          }
```

**Note: These are the same “Raise Issue REST API” parameters we had discussed in section 4.3.2.2.**

The following table is a data dictionary of these fields:

Parameter	Description	Data Type	Min	Max	Mandatory
<b>ORGANIZATION</b>	This is a pre-configured value from Sigmify that needs to be passed while making the REST API Call. Please go through section 4.3.1 for setting this value. For definition please refer to page number # 5.	Alpha Numeric	1	100	YES
<b>PROJECT</b>	This is a pre-configured value from Sigmify that needs to be passed while making the REST API Call. Please go through section 4.3.1 for setting this value. For definition please refer to page # 5.	Alpha Numeric	1	50	YES
<b>PRIORITY</b>	This is a pre-configured list of values from Sigmify that needs to be passed while making the REST API Call. Please go through section 4.3.1 for setting this values. As the name suggests this fields is used for setting the priority of the ticket being raised. Eg: High, Medium, Low, etc	Alpha Numeric	1	30	YES
<b>SUMMARY</b>	This field is used for capturing the summary of the ticket being raised. Please refer to page # 9 where the text passed in the Summary field would appear on the Sigmify Ticket structure.	Alpha Numeric	1	200	YES
<b>DETAILS</b>	This field is used for capturing the details of the ticket being raised. Please refer to page # 9 where the text passed in the Details field would appear on the Sigmify Ticket structure. The details field also acts as a container field to all the custom user related fields that a tenant developer chooses to add to their Raise Issue website page. Please refer to point 6 in the mapping table under section 4.3.2.2 for further details.	Alpha Numeric	1	2000	YES
<b>user</b>	This is the user name (email address) of the Submission user that was set up in section 4.3.1.	Valid Email address	NA	NA	YES

- **HTTPS Multipart request:** A multipart HTTPS request allows one or more different sets of data to be combined in a single body. The Raise Issue function also accepts 1 attachment in .zip format (with size up to 5 MB). To facilitate this transfer along with the json string described above, the multi-part structure is expected by the Raise Ticket function when it receives a call from the tenant website. Please note that even when an attachment is not being sent the call has to be an HTTPS multi-part call.
- **Https Post Call:** The HTTPS Multi-part request is sent to the Sigmify server through a HTTPS Post call that is initiated from the server side. This is recommended. Client side calls might have issues related to cross domain calls. The HTTP Post call need to be sent to an endpoint which has the following format:  
<https://<INSTANCE>/ms/transactUtil/raiseIssue?jsonData=<jsonData>&authToken=<token>>
  - ✓ **<INSTANCE>:** is the Sigmify domain where the call needs to go. The Sigmify support team will be responsible for supplying this information. **Note:** For testing purposes the first domain value that will be provided will be of the stage environment. Once the tenant developer has tested and integrated successfully with the stage instance only then will the production instance domain details will be provided.
  - ✓ **<jsonData>:** This is the json string payload that is explained on page 11. Before embedding it into the above URL it should be encoded in UTF8 format.
  - ✓ **<token>:** This is the same Shiro token explain in section 4.3.2.3.1. Before embedding it into the above URL it should be encoded in UTF8 format.

**Note: The <> brackets are only used here for formatting purposes. Please do not use them while building the three parameters discussed above.**

- **API Response:** The REST API Raise Ticket call sends back a HTTPS response containing a json string. The following is the format of this json string:

```
{
  "flag": "true/false",
  "messages": "[list of messages]"
}
```

  - ✓ The value true for the parameter flag denotes that the ticket has been successfully created in Sigmify.
  - ✓ The value false for the parameter flag denotes that the ticket generation failed in Sigmify.
  - ✓ The list of messages is used for communicating any validation errors that might occur on the Sigmify side when the submission is done.
- **When?** This function needs to be called as soon as we have a Shiro token and the json data payload read to be sent out from the tenant website after their user click on the submit button.
- **How?** Sample codes on how to make this call is available on request. If you need any assistance with making this call. Please get in touch with the Sigmify support group ([support@sigmify.com](mailto:support@sigmify.com))

**Note about HTTPS calls:** The Sigmify server blocks https calls that use any cryptographic protocols that are older than TLS 1.1. So please make sure that the calls originating from your website use higher/latest version of the https cryptographic protocols.