REST API

* Caucasus Mountains, Russia*
Tango REST API EcoSystem
Tango REST API RC4

There are three parts in this proposal: URL specification; Implementation remarks; Implementation recommendations. The first one names valid URLs that must be handled by the implementation. Each URL is presented following this format:

<table>
<thead>
<tr>
<th>URL</th>
<th>RESPONSE TYPE</th>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHOD</td>
<td>url</td>
<td>JSONArray/JSONObject/NULL</td>
</tr>
</tbody>
</table>

Such table is followed by a JSON response's examples block:

```json
URL: { "JSON response": "example" }
```

In two following sections several implementation guidelines are highlighted.

In general API follows standard CRUD idiom:

<table>
<thead>
<tr>
<th>HTTP verb</th>
<th>CRUD</th>
<th>collection</th>
<th>instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>READ</td>
<td>Read a list. 200 OK</td>
<td>Read the details of one instance. 200 OK</td>
</tr>
<tr>
<td>POST</td>
<td>CREATE</td>
<td>Create a new instance. 201 OK</td>
<td>-</td>
</tr>
<tr>
<td>PUT</td>
<td>UPDATE/CREATE</td>
<td>Full Update. 200 OK</td>
<td>Create an instance. 201 Created</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE</td>
<td>-</td>
<td>Delete instance. 200 OK</td>
</tr>
</tbody>
</table>

POST create an instance of collection by the URI of this collection. POST returns the URI and the id of the newly created instance.

**URL example driven specification:**

All URLs in this section omit protocol//host:port part: http://host:port... An implementation may or
Tango REST API status report

- Migration to GitHub
- Read the Docs
- Updates (RC4)
- Added to the official Tango documentation
- ESRF deployment
- DESY deployment
- JINR deployment
Roadmap (main points)

- **RC5**
  - TANGO_HOST port as a matrix parameter:
    - tango/rest/rc4/hosts/orion/10000/... VS tango/rest/rc5/hosts/orion/...
    - OR tango/rest/rc5/hosts/orion;port=10005/...
  - Content-type VS URL addressing:
    - .../attributes/ushort_image/value/image VS .../value Content-type: image/jpeg
    - .../attributes/ushort_image/value/plain VS .../value Content-type: text/plain

- **RC6**
  - HTTP/2
    - push events!!!
mtangorest.server - Tango REST API implementation
Getting started with mTangoREST.server

Download latest mTangoREST.server zipped war or jar file from downloads

-war

Unzip and place it into your servlet container's webapps folder (/CATALINA_HOME)/webapps. Restart the container. If everything is fine you should notice something like the following output in the log:

```
TangoRESTServer has been initialized;
[2915-03-06 03:54:31.118] Artifact tango:war: Artifact is deployed successfully
[2915-03-06 03:54:31.113] Artifact tango:war: Deploy took 2338 milliseconds
```

To be 100% sure that mTango server is properly deployed open your browser and type in the address bar http://localhost:8080/tango/rest. An authorization popup will appear, to pass the authorization you should define user(s) with role(s) mTango-rest, mTango-admin (*). In the servlet container (see [CATALINA_HOME]/conf/tomcat-users.xml), you should see a list of supported Tango REST API versions list:

```
{
   "rc3": "http://localhost:8088/tango/rest/rc3"
}
```

*) This one is used to access mTango admin panel. If you do not plan to use it (it is not required for this tutorial) just skip this role.

-jar

Download the latest mTango server jar.

Define TangoRESTServer device in the Tango DB:

```
```

```
mtangorest.server

- Latest release rc4-1.4
- RC4
  - Image (2D array) data types
  - Event mechanism tweaks
- Refactoring (JAX-RS)
- Consistent http response codes (305, 500, 503)
- Docker container
- Bug fixes
  - No filter configured with id 'json-response-fields-filter'
  - Events wrong response
General purpose Tango web application

Add topics

144 commits 1 branch 2 releases 1 contributor

Branch: master

Ingvord progress #127

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB-INF</td>
<td>README.md edited online with Bitbucket</td>
<td>9 months ago</td>
</tr>
<tr>
<td>apps</td>
<td>Resolve #34: Remember user defined REST_API and TANGO_HOST</td>
<td>3 months ago</td>
</tr>
<tr>
<td>controllers</td>
<td>Resolve #34: Remember user defined REST_API and TANGO_HOST</td>
<td>3 months ago</td>
</tr>
<tr>
<td>docs/classes</td>
<td>README.md edited online with Bitbucket</td>
<td>9 months ago</td>
</tr>
<tr>
<td>engines</td>
<td>README.md edited online with Bitbucket</td>
<td>9 months ago</td>
</tr>
<tr>
<td>images</td>
<td>README.md edited online with Bitbucket</td>
<td>9 months ago</td>
</tr>
<tr>
<td>jmvc</td>
<td>Progress #25:</td>
<td>5 months ago</td>
</tr>
<tr>
<td>libs/webix</td>
<td>README.md edited online with Bitbucket</td>
<td>9 months ago</td>
</tr>
<tr>
<td>models</td>
<td>progress #127</td>
<td>18 days ago</td>
</tr>
</tbody>
</table>
TangoWebapp status report

- Moved to GitHub
- Updates
  - Multiple TANGO_HOSTs
  - Use cookie to remember TANGO_HOSTs
  - Migration to RC4
- Amazon Cloud deployment
- Deployed at DESY (P03nano)
Other Tango REST API projects
Related projects

- **JYSE studio (the next talk)**
  - In-browser dashboard builder
- **tangojs**
  - A bunch of JS components for data visualization
  - MAX IV prepared a demo project: [https://github.com/Unknown22/TangoJSDemo](https://github.com/Unknown22/TangoJSDemo)
- **mTangoUI**
  - Component based framework for building web/mobile applications
Thank you!

Questions?