# JWTStep

**Objective:**

It is used to create a token for every user.

**Attributes:**

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| output\_variable | Used to store the response of the request |
| value\_type | To select type as free marker template or handlebar |
| claims | Used to enter the keys and values. It is list of Key Value pair |
| subject | To enter the subject |
| plugin\_name | To connect with jwt based on signKey |

**Sample config.json:**

{

"workflows": [

{

"name": "jwt",

"id": 200,

"trigger": "rest",

"expression": "jwt",

"method": "GET",

"steps": [

{

"type": "start",

"id": 1,

"next": {

"start": "start"

}

},

{

"subject": "claims",

"claims": [

{

"key": "header1",

"value": "handler"

},

{

"key": "header2",

"value": "value"

}

],

"output\_variable": "jwt-Output",

"plugin\_name": "jwt",

"value\_type": "hb",

"type": "jwt",

"id": "start",

"next": {

"success": "jwt-success",

"failure": "hangup"

}

},

{

"variable\_name": "jwt-Output",

"http\_response\_code": "200",

"response\_content\_type": "text/html",

"value\_type": "hb",

"type": "send",

"id": "jwt-success",

"next": {

"success": "hangup",

"failure": "hangup"

}

}

]

}

],

"filters": [],

"server": {},

"plugins": [

{

"id": 25,

"name": "jwt",

"type": "jwt-plugin",

"version": "1.0",

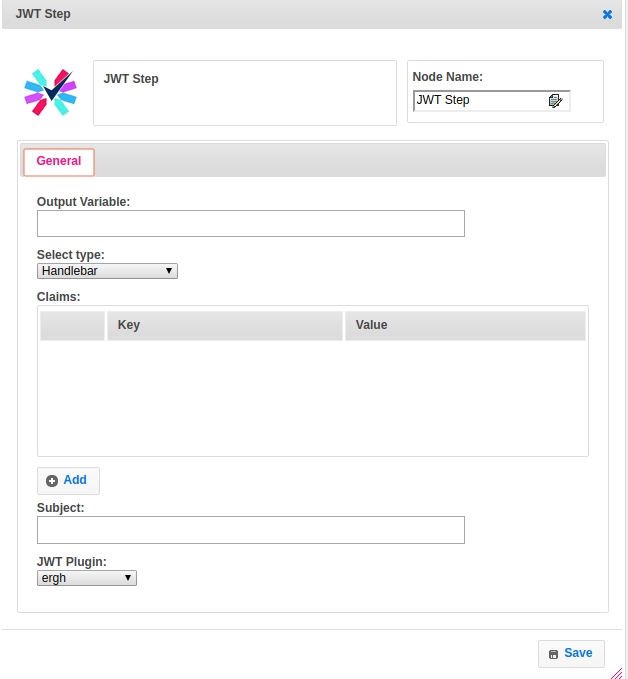
"category": "jwt",

"signKey": "z0NN9huvOKW8Me8QSrREaG9BP4\_7oG1PaKbs6wYlysY"

}

]

}

****

**Executor Description:**

* In executor class first will get jwt\_sign\_key from jwt plugin handler by passing signKey
* Claim values will add to jwtbuilder object.
* Jwtbuilder object will generate jwt token by using signature algorithm and jwt\_sign\_key and then finally will put the jwt token into output variable.