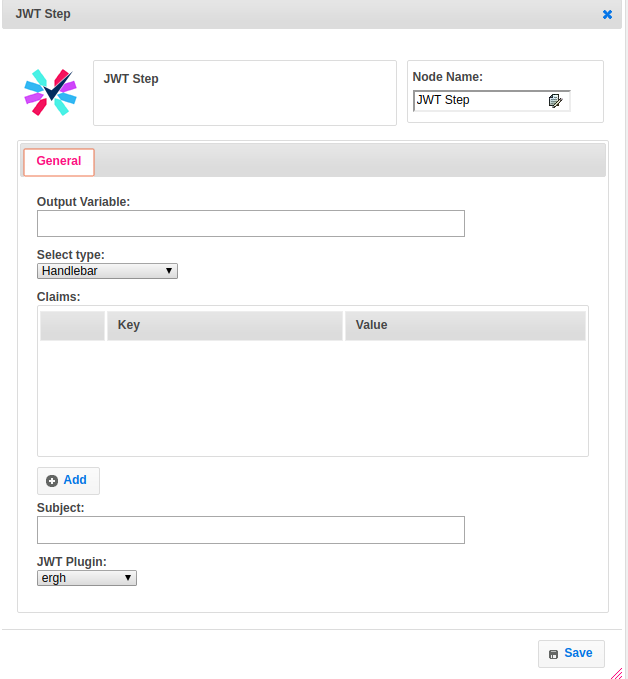
JWT Step

**Objective:**

JSON Web Token (JWT) is an open standard ([RFC 7519](https://tools.ietf.org/html/rfc7519)) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA. The JWT step will be used for creating such a signed JWT token.

## UI

****

## 

## Attributes:

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| Output Variable | Used to store the response of the step |
| Select type | To select type as free marker template or handlebar |
| Claims | Claims are statements about an entity (typically, the user) and additional data  Notice that the claim names are only three characters long as JWT is meant to be compact. |
| Subject | There are three types of claims: registered, public, and private claims.  **Registered claims**: These are a set of predefined claims which are not mandatory but recommended, to provide a set of useful, interoperable claims. Some of them are: iss (issuer), exp (expiration time), sub (subject), aud (audience), and others.  **Public claims**: These can be defined at will by those using JWTs. But to avoid collisions they should be defined in the IANA JSON Web Token Registry or be defined as a URI that contains a collision resistant namespace.  **Private claims**: These are the custom claims created to share information between parties that agree on using them and are neither registered or public claims.  Here subject refers to ‘sub’ in registered claims. |
| JWT Plugin | The name of the plugin which we have configured earlier with signing key. |

## 

## 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.

## Dependent Plugins:

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| SIGN KEY | Signed and encrypted JWTs carry a header known as the JOSE header (JSON Object Signing and Encryption). This header describes what algorithm (signing or encryption) is used to process the data contained in the JWT. |

## Resource:

For more information please refer the below link

<https://jwt.io/introduction/>