Authentication

When using Eva, security is paramount. For each action requested by a user, Eva must ensure:

- Authentication: Is the user who they claim to be. I.e. if a request is issued by Dave, we can verify that it is actually Dave who made the request.
- Authorization: Does the user have permission to perform the requested action. I.e. if Dave tries to delete Jill's user account, Dave needs to have administrator privileges.

We use session tokens to ensure both of these requirements are met, think of them as a passport Eva can issue and check for user authentication and authorization. The session token lets Eva know who the user is and what permissions they have, for example the user is Dave and he is an admin. The user subsequently attaches the token to every request they make to Eva. Only Eva can make these tokens and can also determine if a token is genuine or not.

Authentication

To check: *Is the user who they claim to be*

When using the Eva programmatically over the HTTP API, the user will need to generate new session tokens, Eva will use these tokens as a proof of identity.

Authorization

To check: *Does the user have permission to access this*

There are two levels of permission on Eva: `admin` and `user`. In broad terms, `user` permission allows access to use the robot (run toolpaths, use I/O etc.), while `admin` permission can also access:

- Users: creating, updating and deleting users, changing user permissions.
- Eva Config: Changing Eva's config options, such as time and networking options.

The first user created on Eva has `admin` permissions, all subsequent users are created with `user` permissions but can be upgraded to `admin` permission by another `admin` user.

API Tokens

Using an API token, a user can generate a session token for authentication. A user can generate API tokens using Choreograph's user page (*your email* > *Profile* > *API Tokens*). Please see the API token session endpoint for more information about session generation over the API.

As these API tokens provide access to Eva's API, it is important to keep these secret to ensure only only authorized agents can access the robot.

It is also recomended to use unique API Tokens between different users, computers and apps as much as possible. This keeps isolation in case of the token being misused, tokens can be revoked to invalidate their access.

Session Tokens

Session tokens are used to give API access to Authenticated users for a specified time period. They can be created using a valid API Token and the API token session endpoint. Once you have your session token, include it as the `Authorization` header of each request prefixed with `"Bearer"`:

Example

```
GET /api/v1/users HTTP/1.1
Host: eva.automata
Accept: */*
Authorization: Bearer SESSION_TOKEN
Content-Type: application/json
Content-Length: 0
```

Session tokens have an expiry time: they are only valid for a set time duration. Session tokens are valid for 30 minutes from creation, but can be renewed via the renew endpoint. They can be renewed up to a maximum duration of 48 hours before a new session is required. This means that the user needs to periodically re-authenticate themselves. Session tokens can also be invalidated before their expiry time by the following actions:

- If a user is deleted, all session tokens for that user (API token or user/password generated) are invalidated
- If an API token is deleted on Choreograph, all session tokens generated with that API token are invalidated
- A session token can be invalidated using the session invalidation endpoint
- In a power cycle of Eva, all session tokens are invalidated

Endpoints

Create API Token Session

This endpoint checks that the supplied API token is valid and creates a new session. Session tokens are valid for 30 minutes from creation, but can be renewed via the renew endpoint.

Request

```
POST /api/v1/auth
```

Payload

```
{
  "token":YOUR_API_TOKEN_HERE
}
```

Example reply

```
{
  "token":NEW_SESSION_TOKEN
}
```

or an error.

Invalidate Session

This endpoint invalidates the session token supplied in the auth header.

Request

```
DELETE /api/v1/auth
```

Payload

`NONE`

Response

Status Code `204`

or an error.

Renew Session

This endpoint renews the pre-existing session token supplied in the auth header. Session tokens are valid for 30 minutes from creation or renewal and can be renewed up to a maximum duration of 48 hours before a new session is required.

Request

```
POST /api/v1/auth/renew
```

Payload

`NONE`

Response

Status Code `204`

or an error.