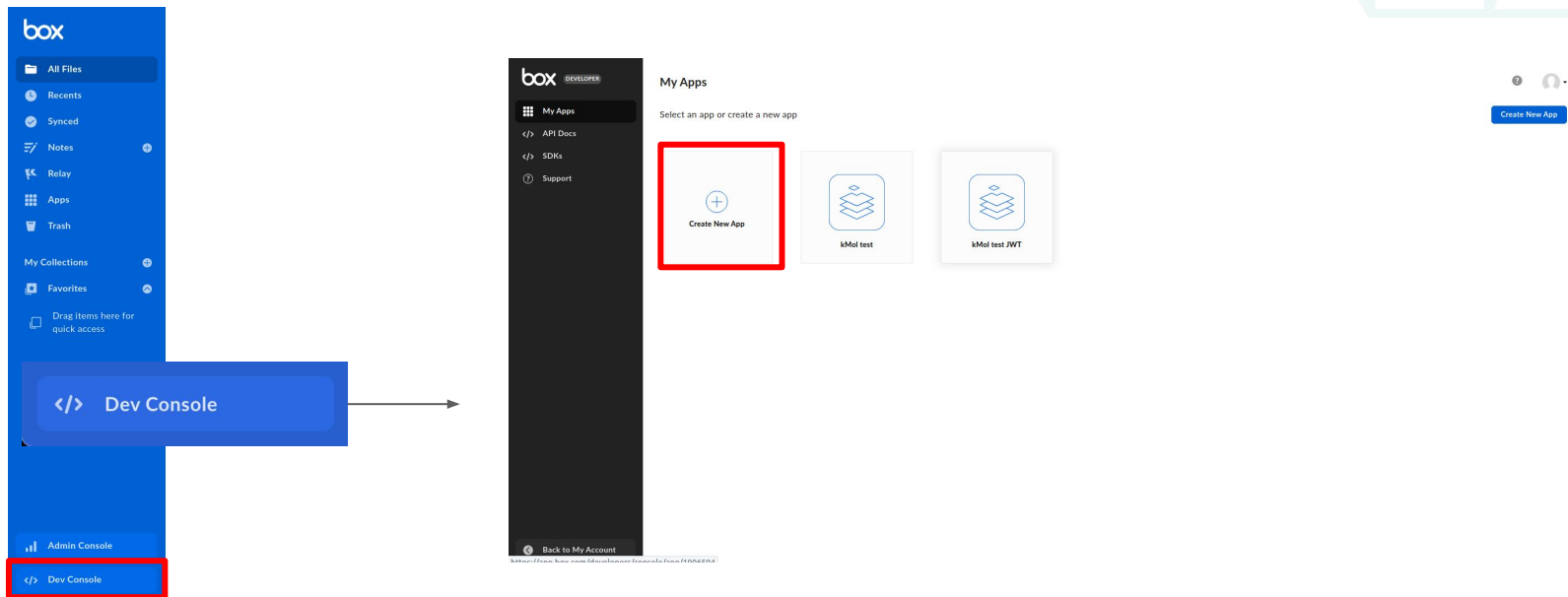


Box Documentation: Administrator side

Administration set up

- Please set up a JWT app the following:
 - From the future box admin account:
 - Go into the dev console (bottom left on box) then **create a new app** with JWT identification.



Administration set up

Select Custom App

box

DEVELOPER

My Apps

API Docs


SDKs

Support

Back to My Account

My Apps > Create New App


Select an app type to get started



Custom App
Best for the majority of apps and integrations. Supports both server-server and end-user apps

- ✓ Select desired authentication method (user or enterprise)
- ✓ Select application's access (i.e. scopes)
- ✓ Create webhooks
- ✓ Publish apps to the App Center or as integrations
- ✓ Request access to all files and users in an enterprise


[View Documentation](#)



Limited Access App
Best for Box View apps or previewing Box content in another application

- ✓ Server side authentication with an app token
- ✓ Request access to all files and users in an enterprise
- ✗ Cannot access all API endpoints
- ✗ Cannot edit scopes, default scopes are set
- ✗ Cannot store data in the Enterprise
- ✗ Cannot publish apps to App Center

[View Documentation](#)



Box Custom Skill
Best for apps that want to use an external service to enrich content with metadata

- ✓ Send content to a third party service based on updates to a file in Box
- ✓ Utilize a Box provided secure token for easy, secure data transfer between services
- ✓ Create metadata on a Box file based on information from a third party service
- ✗ Cannot publish Skill to App Center

[View Documentation](#)



Cancel

Administration set up

Select a Server authentication with JWT.

Note that the App name will be needed later.

Custom App

×

Select an authentication method and enter a name for your app. You can change the name later in General Settings.

Authentication Method

☒ **Server Authentication (with JWT)**

- Best for building integrations or apps with external collaborators - [Learn More](#)
- The app authenticates using a JSON Web Token. Requires a public/private key-pair for added security.

☐ **User Authentication (OAuth 2.0)**

- Best for building mobile or web apps - [Learn More](#)
- Users must authenticate with their Box login.

☐ **Server Authentication (Client Credentials Grant)**

- Best for scripting and back office integrations - [Learn More](#)
- The app authenticates using Client ID and Client Secret.

App Name

test-JWT

By clicking 'Create App', you agree to the terms of the [Box Developer Agreement](#) and the [Box Privacy Policy](#).

Cancel

Create App

Administration set up

Please check all boxes in **application scope** and **advanced features** and save the changes

test-JWT



General Settings

Configuration

Webhooks

Authorization

App Diagnostics

Manage authentication methods and app permissions

Save Changes

Application Scopes

The app scopes determine which endpoints and resources your app can successfully call. [Learn more about all of our scopes.](#)

Content Actions

☒ Read all files and folders stored in Box

Access to content is further restricted by the users' permission and Access Token used.

☒ Write all files and folders stored in Box

Necessary to download files and folders. Access to content is further restricted by the users' permission and Access Token used. Read access is required when Write access is selected.

Administrative Actions

☒ Manage users

☒ Manage groups

☒ Manage retention policies

For use with the [Governance add-on](#).

☒ Manage enterprise properties

For use with the event stream, enterprise's attributes, and device pins. App + Enterprise Access is required to use this scope.

Developer Actions

☒ Manage webhooks

☒ Enable integrations

☒ Manage Box Relay

Interact with Box Relay endpoints. [Learn more about Box Relay APIs.](#)

Administration set up

In order to connect to the api with the new created app, we need a couple more actions.

First in **Configuration** go to Add and Manage Public Keys and click on **Generate a public / private key**. To do so you will need the Double Authentication to be enable.

Add and Manage Public Keys

Generate an RSA keypair to sign and authenticate the JWT request made by your app or upload your own public key. [Learn to generate your own RSA keypair](#).

Note: Box does not store your private key, so make sure you save the downloaded file if you are using our generate button.

Add a Public Key

Generate a Public/Private Keypair

2-Step Verification

Email

Enter the 6-digit code from the email sent to your email address.

Authentication Code

Enter 6-digit code

Submit

[Resend Code](#)

[Help](#)

[Log Out and Start Over](#)

Administration set up

The previous step will download a json files containing all the information necessary to connect to the api with our App. It's possible to create multiple key pair and to revoke them if necessary.

The **file download is very important** and is needed in order to run the program with box.

Add and Manage Public Keys

Generate an RSA keypair to sign and authenticate the JWT request made by your app or upload your own public key. [Learn to generate your own RSA keypair.](#)

Note: Box does not store your private key, so make sure you save the downloaded file if you are using our generate button.

Public Key 1 ID: zf7hvyfa

Remove

Add a Public Key

Generate a Public/Private Keypair

```
Open 969235509_zf7hvyfa_config.json
~/Downloads
1 {
2   "boxAppSettings": {
3     "clientId": "lbwtgppcay3bfhaqo4kvizlk94160cqn",
4     "clientSecret": "jEUJUwEA48mqXKvkVfctotDjvaWtgd",
5     "appAuth": {
6       "publicKeyID": "zf7hvyfa",
7       "privateKey": "-----BEGIN ENCRYPTED PRIVATE KEY-----
      \nMIIFDjBAGqhkG9w0BBQ0wMzAbBgkqhkiG9w0BBQ0wDgQI4zNT/K4MbIACAggA\
      z20sbFQSCBMjK+CV+0Tws7nFgMLFSAIcDSuUTPHx9\nShfME+5uPYZDS9W417bf+PP
      x\nUB6SRNDX42MBC+yobJj+z3d5t6znt64vtTodpsKM/-
      KMURQJQn8dzXa4Kv5lYz89l\n40pwSoT4V0gOPTu+TvMiN0MCu2UqX8G0KEV/eiYJ5+
      yxv93lEdDetJEdXoohOaxblcmkz20Q0Aix2ugH3YctIF/MABK9g\nnuEPZrpnxa/17C
      6JldF07JevbW3xpa032gu1\ncRy0f8jnejiLJBvAHhsf3RQgHNPR6aQ5KFTT8zrWM
      DTKNpz9HTkt0JL9WV8p16rNZKwsmjgq/-
      qaQRw3FAaYu7EjPIFal8ht\nI9ayRmp0cZkL2AIuthMVM9p8hXgZnc4Sd0Q6ySSJv
      kaS6LR9yoo0Dw8F4q8ggfGVwIFGLC0sCLNbbPF0c6LEmVv9\nnWYV6051mqPmqkewar
      CF3ux2gdLb7LAU+zeegMmHDMicJ3JLe8np\nnQ8/uzApWwuE090zGkenEb2bGN2ELSGU+
      l11mR\nnmn0LwLMgzSHBKvICQnzUmv1mL7gtCHSIXz07bRMB00eFtnLwrK8X4k0yk
      z3aGQlH8kNobe\nsELCyKmXsf6mXJNeFknYQ4l/-
      2psTuKPN7d4cFYq+Wg5WaqWlf5y5ni0Xp3trgyi8\nnuGEDQTE01KFVcfjLCHCR0k932
      09doMYd51/0JVBaT4D0qef+cRWQjkzYvFETReZKIa9pGDeI8EhG850E\nn4/-
      wVJ3HuJrCEGLKHSjGAgBxajnfBDD2YgRec5CItqX1bW3dxd0/+MPFLaukNB5BR\nneXj
      +80P2k3bXRkawlG54xe7Lpiv5InopExvd+10+14dw18uqil\nnTstN3LncP0kTB05FG
      eZr\nnQrLoarBkLWcetthN70d93aXfk0m3kp8EGiuj9FdmrvXBRYLgRp7lgcVlpoCF5t
      END ENCRYPTED PRIVATE KEY-----\n",
8     "passphrase": "66e8c28a9c3e012cf518cedc39856e85"
9   },
10 },
11 "enterpriseID": "969235509"
12 }
```

Administration set up

Now change tab to the **Authorization** tab and Review and Submit you Application. This step needs to be done after updating the rights of the application.

test-JWT

General Settings

Configuration

Webhooks

Authorization

App Diagnostics

Submit and manage your app's admin authorization

Submit app for authorization for access to the Enterprise

Submit your app for approval based on the access requested. Your admin will receive an email of the request. You will receive an email once the request is approved or denied.

[Learn more about app authorization](#)

Review and Submit

You must re-submit your app if any changes are made after submission

Review App Authorization Submission

App Info

App Name
test-JWT

Client ID
628j046b7tg1k5rnw9qkgzrb6eplr5j2

App Description

Enter a brief description about this application

Cancel

Submit

Administration set up

Now Change to the **Admin** console and go to **Apps** and to the **Custom Apps Manager** tab. We can see our new created app is waiting for authorization.

The screenshot displays the Box Admin console interface. On the left, a blue sidebar contains navigation options: All Files, Recents, Synced, Notes, Relay, Apps, Trash, My Collections, Favorites, and a section for dragging items. At the bottom of this sidebar, 'Admin Console' and 'Dev Console' are highlighted with red boxes. A secondary dark sidebar on the right lists 'Insights', 'Users & Groups', 'Content', 'Reports', 'Governance', 'Relay', 'Apps' (highlighted with a red box), 'Account & Billing', and 'Enterprise Settings'. The main content area is titled 'Box Apps & Integrations' and features a 'Custom Apps Manager' tab highlighted with a red box. Below this, a section titled 'Custom Apps Manager' provides instructions on granting access to custom apps. A toggle switch allows switching between 'Server Authentication Apps' and 'User Authentication Apps'. A dropdown menu is set to 'Any Authorization'. A table lists the installed apps with columns for Name, Last Activity, Developer Email, Authorization Status, and Enablement Status. The first app, 'test-JWT', is in a 'Pending Authorization' state and is highlighted with a green border, with its 'View' button also highlighted with a red box. The other two apps, 'kMol test JWT' and 'kMol test', are 'Authorized' and 'Enabled'.

| Name | Last Activity | Developer Email | Authorization Status | Enablement Status |
|--|------------------|------------------------------|-----------------------|-------------------|
| test-JWT <small>fbwfgppcay2bfhaqo4kvi2k94160cqn</small> | January 16, 2023 | vincent.richard@elix-inc.com | Pending Authorization | Disabled |
| kMol test JWT | December 5, 2022 | vincent.richard@elix-inc.com | Authorized | Enabled |
| kMol test | December 4, 2022 | vincent.richard@elix-inc.com | Authorized | Enabled |

Administration set up

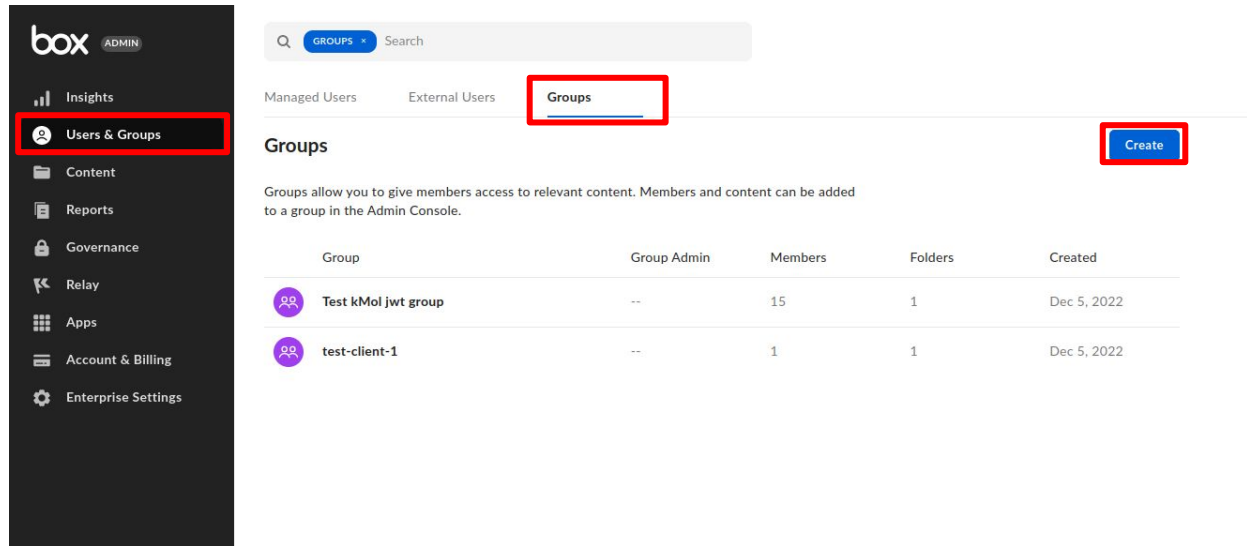
Now you can Authorize the application. This will enable us to access to box. But for now no directory can be access outside the root dir.

The screenshot displays the Box Custom Apps Manager interface. At the top, there is a search bar and navigation icons including a help icon, a notification bell with a red badge, and a user profile icon labeled 'LB'. Below the navigation bar, the 'Custom Apps Manager' tab is selected. The main content area shows the 'test-JWT' application, which is represented by a blue icon with white stacked layers. To the right of the icon, the text 'test-JWT' is displayed, followed by 'App Details'. Below this, two rows of information are shown: 'Last Activity' with a blue dot icon and the date 'December 19, 2022', and 'Developer Email' with the address 'loic.beheshti@elix-inc.com'. On the right side of the application card, there are two buttons: 'Decline' and 'Authorize'. The 'Authorize' button is blue with white text and is highlighted with a red rectangular border.



Administration set up

We will now shared a global directory with our application to use it for saving the models.

In the **Admin Console** go to **Users & Groups** and to the tab **Groups** and Create a new Group.



The screenshot displays the Box Admin Console interface. On the left, a dark sidebar contains navigation links: Insights, Users & Groups (highlighted with a red box), Content, Reports, Governance, Relay, Apps, Account & Billing, and Enterprise Settings. The main content area has a search bar at the top with 'GROUPS' selected. Below the search bar, there are tabs for 'Managed Users', 'External Users', and 'Groups' (highlighted with a red box). A 'Create' button (highlighted with a red box) is located in the top right corner of the Groups section. The 'Groups' section includes a descriptive text: 'Groups allow you to give members access to relevant content. Members and content can be added to a group in the Admin Console.' Below this text is a table listing existing groups.

| Group | Group Admin | Members | Folders | Created |
|---|-------------|---------|---------|-------------|
|  Test kMol jwt group | -- | 15 | 1 | Dec 5, 2022 |
|  test-client-1 | -- | 1 | 1 | Dec 5, 2022 |

Administration set up

The first step provide basic information. Note the group name will be needed inside the configuration provided to the application.

Managed Users External Users **Groups**

Groups > **Create Group** Step 1 of 3 Cancel Next

Group Details
Enter a unique group name that has not already been used in your account.

Group Name

test-JWT

Description (optional)

Enter a description or details

Permission Setting
Specify which users in your account can invite this group to share content and view the members of this group.

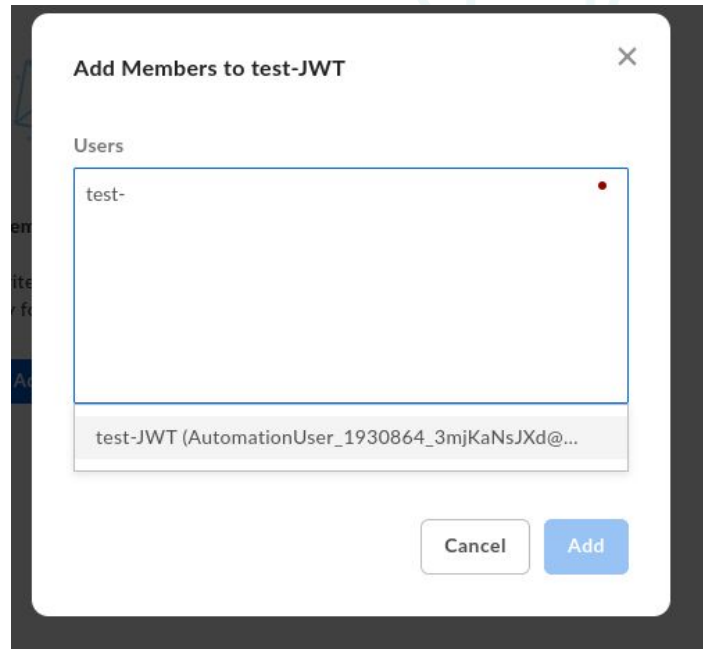
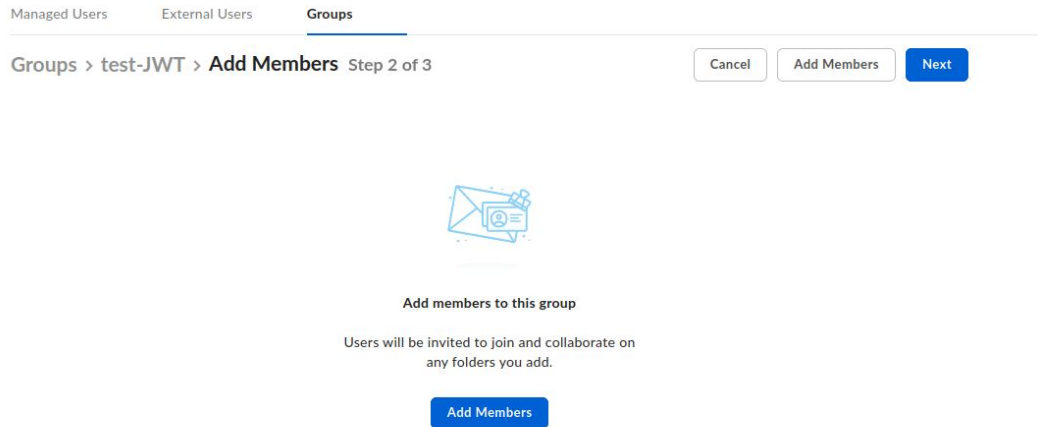
☐ Company
Anyone in your company can provide this group access to folders, view members in this group, and assign a task to members in this group.

☐ Group Members
Only group members can provide this group access to folders, view other members, and assign a task to members in this group.

☒ Admins Only
Only Admins can invite this group to folders, view members in this group in the Admin Console, and assign a task to members in this group.

Administration set up

The search of user name should enable you to find the application name. Please add it as a group member.



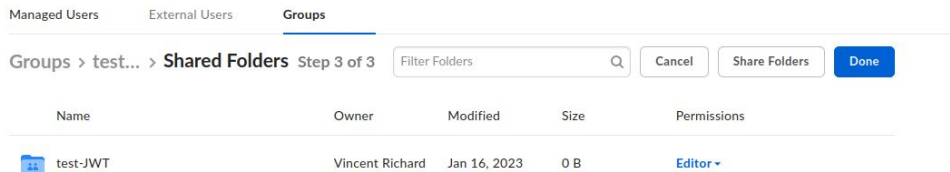
Administration set up

You can now add the folder you want to use for this application. **The folder name will be needed in the parameter of the application.**



Share Folders with this Group

Invite this group to collaborate in any folder owned by your organization so members can gain access to content immediately. Note that this action will not generate a collaboration invitation email.



Administration set up

The application should now be ready to use with the code provided. As the recap you will need those information to make the application work:

- `box_configuration_path`: The file generated with containing Private / Public key pair information.
- `group_name`: The name of the group created for the BoxApplication.
- `shared_dir_name`: The folder shared with the BoxApplication created.

Documentation

Config Documentation

In added several parameters to make the application work with both boxApi and grpc. First we added a server_type, servicer_type and a client_type to know which type of connection should be used.

Server specific parameter:

- server_type: `"mila.services.servers.BoxServer"` or `"mila.services.servers.GrpcServer"`
- server_manager_type: `"mila.services.server_manager.BoxServicer"` or `"mila.services.server_manager.GrpcServicer"`

Client specific parameter:

- client_type: `"mila.services.clients.BoxClient"` or `"mila.services.clients.GrpcClient"`

Config Documentation

We also separated the grpc argument and box argument. For box we have:

```
"box_configuration": {  
    "box_configuration_path": The Public / private key pair file downloaded during the admin  
set up,  
    "shared_dir_name": The name of the directory shared with the group of the application,  
    "save_path": The path inside shared_dir_name where the training logs and weights should  
be save. To avoid issue we also add a date_time layer based on the time launched.,  
    "group_name": The name of the group of the application,  
}  
  
"box_configuration": {  
    "box_configuration_path": "kmol_jwt_config.json",  
    "shared_dir_name": "kmol folder jwt",  
    "save_path": "test_run",  
    "group_name": "Test kMol jwt group"  
}
```







- There is no change to the current api, so to launch a training. We use the following command
 - mila server \$path_to_client_config.json
 - mila client \$path_to_client1_config.json
 - mila client \$path_to_client2_config.json
- Note that the server should be started before the client.

Output Documentation

Inside the box directory multiple file are created:

- Heartbeats: File used to ensure the clients are still only. If a client failed to connect for too long he will be drop and the training will continue without him.
- Rounds: All models each in their respective round directory. The model have the following naming syntax: \$client_name.pt the aggregation is named server.agg
- Authentication: used at the start of the training to know the number of client.







 > kmol folder jwt > test_run > 2023-01-16_19-11

| NAME | UPDATED ↓ | SIZE |   |
|---|----------------------------|---------|---|
|  heartbeats | Yesterday by tester1 | 2 Files | |
|  rounds | Yesterday by tester1 | 2 Files | |
|  authentications | Yesterday by kMol test JWT | 2 Files | |
|  cfg.json | Yesterday by kMol test JWT | 2.3 KB | |

Output Documentation

The heartbeat folder will contain text file with the following name: \$client_name_heartbeat.txt. These files contains the time of the last modification of the client. It is not readable by a human, ex: 1673864102.2054172

 > test_run > 2023-01-16_19-11 > heartbeats





| NAME | UPDATED ↓ | SIZE |   |
|---|----------------------|------|---|
|  tester1_heartbeat.txt V4 | Yesterday by tester1 | 18 B | |
|  tester2_heartbeat.txt V3  | Yesterday by tester2 | 18 B |  Share |

Output Documentation

The authentication folder work in a similar way. The input is first empty when the client create the file to register. Then the Server will fill the text base on the result of the registration.

Either: Authenticated - Token 23118091987 or Error - Authentication failed... Registration is closed.








 > test_run > 2023-01-16_19-11 > authentications

| NAME | UPDATED ↓ | SIZE |   |
|--|----------------------------|------|---|
|  tester1_authentication.txt V2 | Yesterday by kMol test JWT | 33 B | |
|  tester2_authentication.txt V2 | Yesterday by kMol test JWT | 33 B | |

Output Documentation

A certain number of file are also generated during the training to provide information about errors or termination.

- `cfg.json`: The configuration file used during the training
- `$client_name.lost_connection`: text file indicating that the client is not part of the training anymore. The text contains in which round the client was disconnected.
- `server_ended.txt`: Text file indicating the server has stop, the reason could be an error or the termination of the training. The text indicates the reasons. Ex: Training finished 1673943293.448432

| | | |
|---|----------------------------|--------|
|  <code>server_ended.txt</code> | Yesterday by kMol test JWT | 35 B |
|  <code>tester9.lost_connection</code> | Yesterday by kMol test JWT | 26 B |
|  <code>tester5.lost_connection</code> | Yesterday by kMol test JWT | 26 B |
|  <code>tester2.lost_connection</code> | Yesterday by kMol test JWT | 26 B |
|  <code>tester10.lost_connection</code> | Yesterday by kMol test JWT | 26 B |
|  <code>tester4.lost_connection</code> | Yesterday by kMol test JWT | 26 B |
|  <code>cfg.json</code> | Yesterday by kMol test JWT | 2.3 KB |