

FDSA - Web Hooks

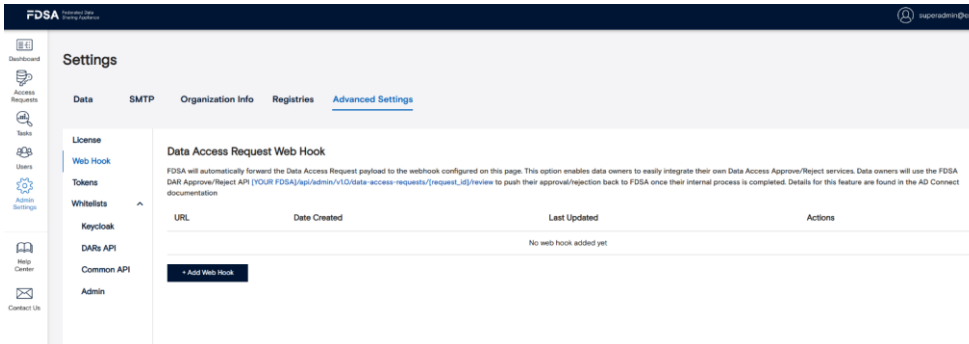
What does the Web Hook do?

- In FDSA you can add an API URL in the Web Hook settings.
- If an API URL is added, FDSA will automatically forward Data Access Requests to that URL.
- This option enables data owners to easily integrate their own data access Approve/Reject services.
- Data owners can use the FDSA DAR Review API to push their approval/rejection back to FDSA once their internal process is completed.

How to add a Web Hook in FDSA

Commented [LS1]: Update

- Go to the Settings tab in your FDSA.
- Click on 'Add Web Hook' on the 'Data Access Requests Web Hook Configuration'. See image below.



- Once you add it, you will see something like in the image below.



- Under 'Actions' you can also 'Edit' or 'Delete' this URL.
- Note that the endpoint **must be a POST API** so FDSA can send the payload there.

DAR Payload sent to the Web Hook endpoint.

These are the payloads (json format) that the FDSA will send to the POST Web Hook API endpoint. The only difference between the two DAR payloads is in the 'scopes' key-value.

Level 1 DAR Example:

```
{
  "id": "XXX-XXX",
  "requester_name": "level2",
  "requester_email": "level2@email.com",
  "start_date": "2022-04-27T20:59:16.210505",
  "end_date": "2024-11-27T20:59:16.210505",
  "status": "RECEIVED",
  "project_description": "We want to start a scientific study...",
  "purpose": "Scientific research",
  "project_code": "project_level2",
  "user_origin_ip": "<IP address the user will access the API from (optional)>",
  "delivery_url": "<URL to deliver access parameters back to (optional)>",
  "user_source_url": "<URL to page detailing the request (optional)>",
  "dataset_id": 1,
  "dataset_code": "sample1",
  "scopes": "can_validate,can_select,can_beacon",
  "reference": "project_level2",
  "request_url": "<URL to page detailing the request (optional)>",
  "form": {
    "form": [
      {
        "name": "name_1",
        "value": 1
      },
      {
        "name": "name_2",
        "value": "text"
      }
    ]
  },
  "created_at": "2023-02-07T16:22:29.568937",
  "updated_at": null
}
```

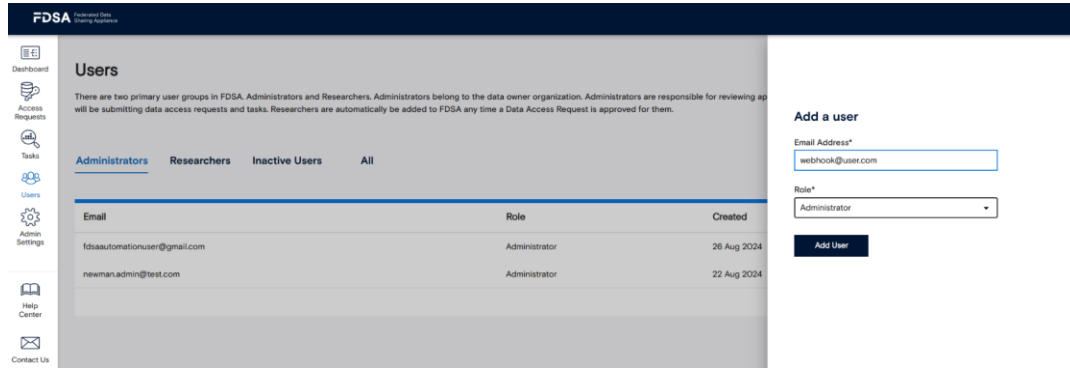
Level 2 DAR Example:

```
{
  "id": "XXX-XX",
  "requester_name": "level2",
  "requester_email": "level2@email.com",
  "start_date": "2022-04-27T20:59:16.210505",
  "end_date": "2024-11-27T20:59:16.210505",
  "status": "RECEIVED",
  "project_description": "We want to start a scientific study...",
  "purpose": "Scientific research",
  "project_code": "project_level2",
  "user_origin_ip": "<IP address the user will access the API from (optional)>",
  "delivery_url": "<URL to deliver access parameters back to (optional)>",
  "user_source_url": "<URL to page detailing the request (optional)>",
  "dataset_id": 1,
  "dataset_code": "sample1",
  "scopes": "can_compute",
  "reference": "project_level2",
  "request_url": "<URL to page detailing the request (optional)>",
  "form": {
    "form": [
      {
        "name": "name_1",
        "value": 1
      },
      {
        "name": "name_2",
        "value": "text"
      }
    ]
  },
  "created_at": "2023-02-07T16:22:29.568937",
  "updated_at": null
}
```

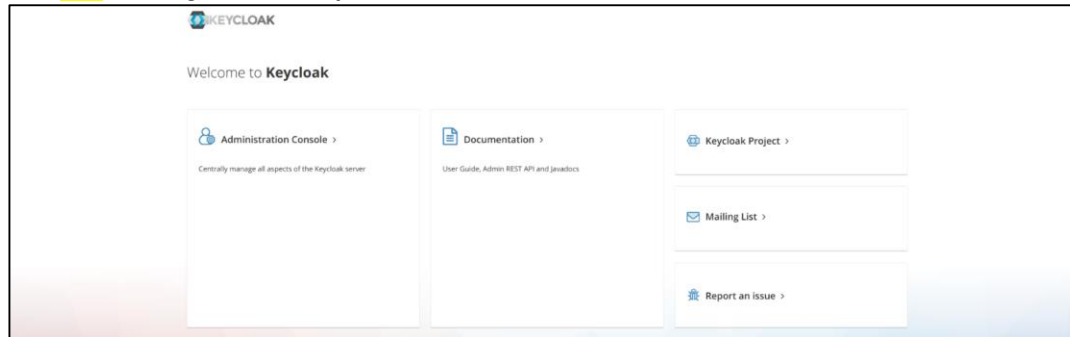
How to use the DAR Review API to Approve/Reject a DAR

1. Create a new Administrator in FDSA and configure it.

Go to the FDSA Users tab and add the webhook user

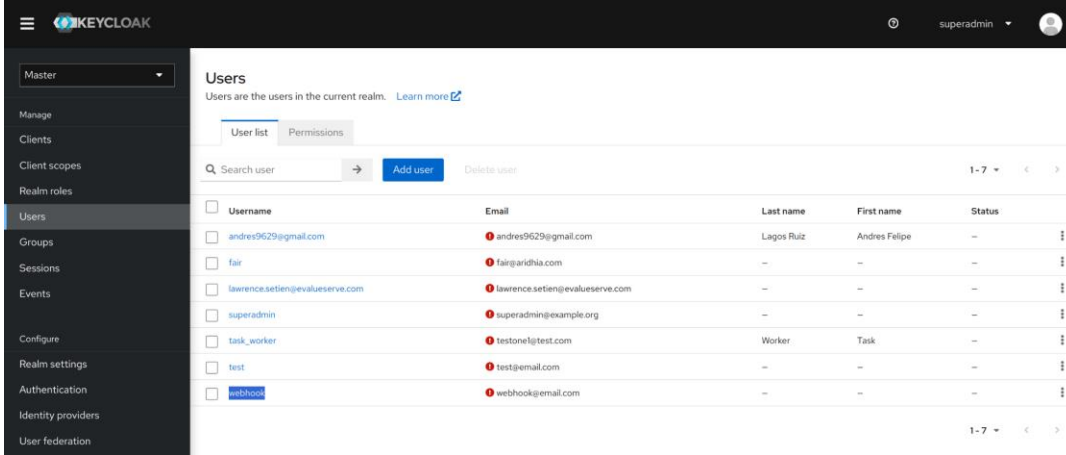


As a **root** user, login into the Keycloak UI and click on Administration Console

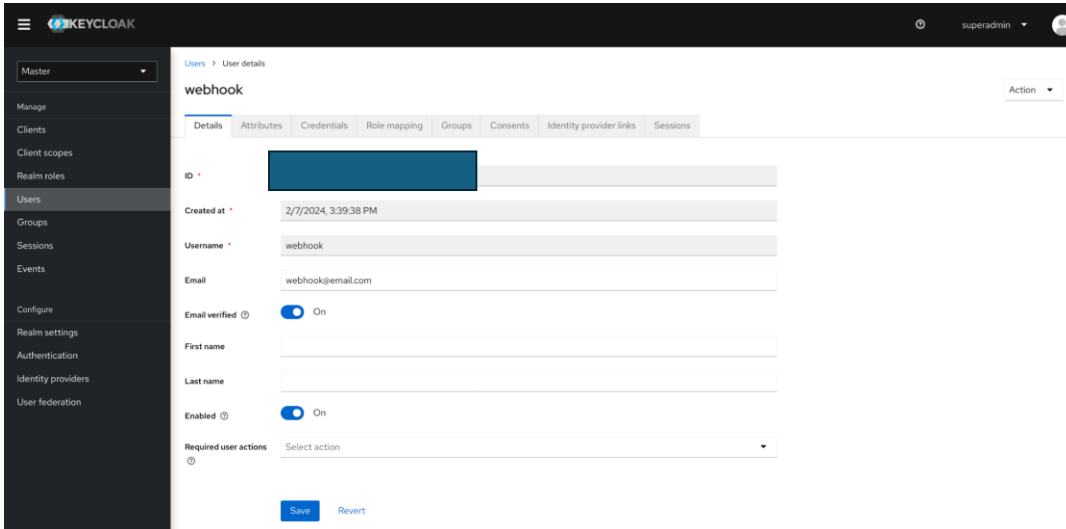


Federated Data Sharing Appliance Full Guide

Now click on the Users Tab, and then in the webhook user just created:

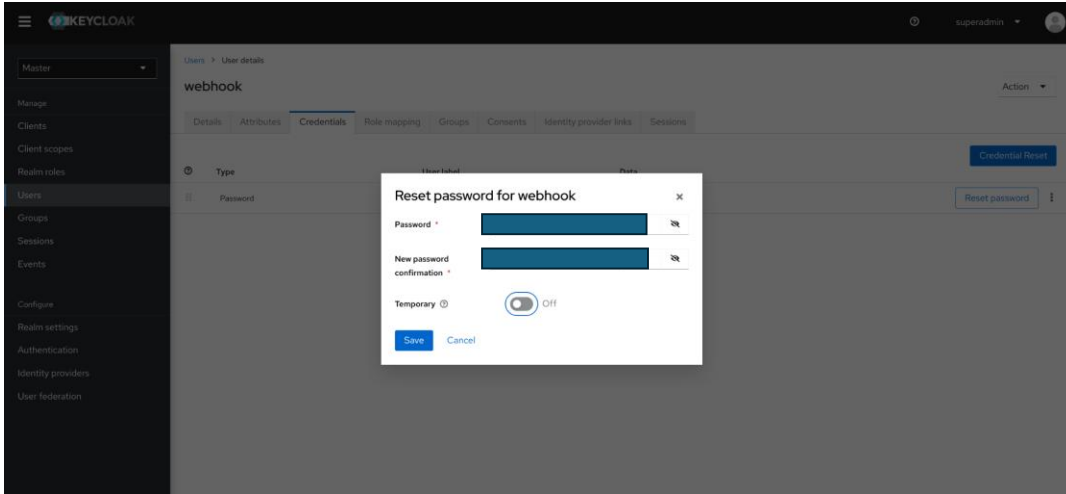


Set the email as verified and remove the **'Required user actions':** 'Configure OTP' and 'Update Password'. The screen should look like this:



Federated Data Sharing Appliance Full Guide

Next go to the **'Credentials'** tab and click on **'Reset password'**, make sure to add a very strong password and unselect the **'Temporary'** option. Save and confirm.

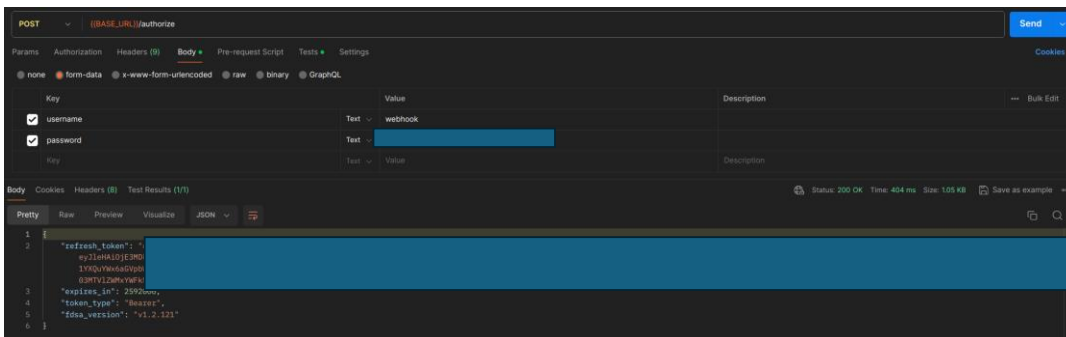


2. Get the new Administrator token

To get an admin token you need to execute the /authorize API. Below you can see an example on how to call it.

Header required: Content-Type: multipart/form-data.

For username and password, use the webhook user credentials.



3. Execute the DAR /review API

In the images below you can find an example on how to use the /review API.

Federated Data Sharing Appliance Full Guide

The screenshot displays a REST client interface for a POST request to the endpoint `{{BASE_URL}}/api/admin/v1.0/data-access-requests/{{request_uuid}}/review`. The 'Authorization' tab is selected, showing a 'Bearer Token' type and a token field containing `{{admin_token}}`. Below this, the 'Body' tab is active, showing a JSON payload:

```
1 {
2   .. "reviewer_uuid": "{{user_uuid}}",
3   .. "decision": "APPROVED",
4   .. "reason": "ok",
5   .. "id": "{{request_uuid}}",
6   .. "trusted": false
7 }
```

Required Headers:

- Authorization: Bearer <admin_token>
- Content-Type: application/json

Required Variables:

- admin_token (string): The refresh_token value returned on the /authorize API
- id/request_uuid (string): The uuid of the DAR, you will receive in the Web Hook API URL
- decision (string): APPROVED / DENIED
- reason (string): The reason why this was APPROVED or DENIED
- reviewer_uuid/user_uuid (string): The Admin UUID that is Approving or Denying the DAR. You can find this UUID by decoding the refresh_token ('sub' value), copying it from the FDSA UI (if it's not the superadmin), or by copying it from the Keycloak UI.
- trusted (boolean): true / false. If it's true, then all tasks generated from this DAR will go straight to COMPLETE, without needing a review first.

Contact Support

For any support requests or inquiries, please do not hesitate to contact us at fdsa.support@alzheimersdata.org.