EU-FarmBook API manual

A comprehensive guide to making contributions to the EU-FarmBook platform using the API

Funded by
the European Union

Table of Contents

1. Introduction

Welcome to the guide for the EU-FarmBook platform's public API (Application Programming Interface).

This guide will walk you through the steps to authenticate, view projects you can contribute, and upload your Knowledge Objects and metadata.

You can also use the API to query information for the Knowledge Objects and metadata you uploaded to the EU-FarmBook.

The EU-FarmBook API enables the contribution of Knowledge Objects programmatically, for example, using open-source frameworks including Python, JavaScript, C#, etc.

You can also directly use the API documentation interface (see Chapter 4); however, this will be a more manual process, and in such cases, we recommend the upload form.

2. Prerequisites

To provide Knowledge Objects and their metadata via the EU-FarmBook platform API, please first ensure the following:

- You have created an account with the EU-FarmBook.

- You have a coordinator or contributor access for a given project. Click here for more information.

- You have one or more knowledge objects (e.g. digital files including .pdf, .mp4, .docx, etc.)

- For each knowledge object, you have metadata following the EU-FarmBook metadata guide (see the EU-FarmBook Metadata guide in the "Manuals and Documents" section in the Support page here).

- You have some background knowledge about APIs and how to interact with them.

3. High-Level Overview

As illustrated in Figure 1 below, uploading a Knowledge Object and its metadata via the API is a four-step process. Each step involves a contributor providing a set of inputs to a specific API endpoint, producing a specific output. Uploading multiple knowledge objects is possible by repeating steps 3 and 4.
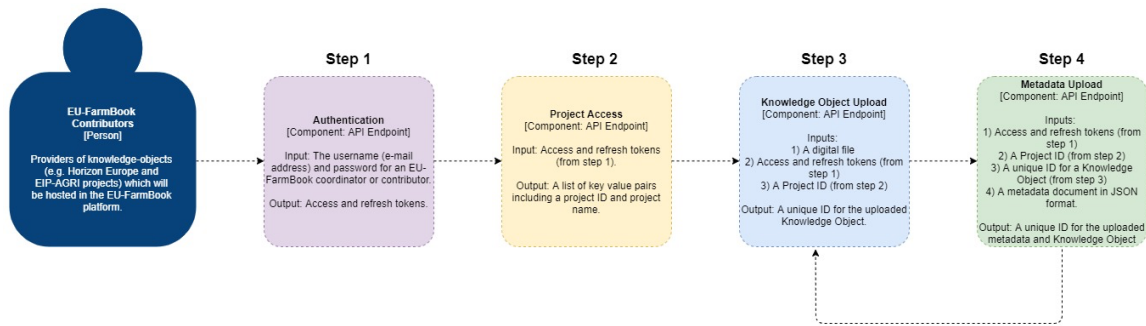
Figure 1: A high-level representation of the four-step process enabling contributors to upload Knowledge Objects and metadata via EU-FarmBook API.

4. Endpoints and methods

The URL prefix for the EU-FarmBook public endpoint is:

https://api-public.eufarmbook.eu

The specific endpoints and methods for the four steps detailed in the above section can be found in Table 1 below.

Step	Endpoint	Method	'accept' Header	Content-Type Header
1 – Authentication	/api/authentication/token/	POST	application/json	application/json'
2 – Project Access	/api/authentication/projects/	POST	application/json	application/json
3 – Knowledge Object upload	/api/upload/knowledge_object_file/	POST	application/json	multipart/form-data'
4 – Metadata upload	/api/upload/knowledge_object_metadata/	POST	application/json	application/json

Table 1: API Endpoints and Methods

You can also test each endpoint and get further information by visiting the online documentation here: https://api-public.eufarmbook.eu/docs.

5. Detailed guidelines for uploading

Step 1: Authentication

Whenever you use the EU-FarmBook API, you must generate access and refresh tokens. These tokens expire periodically and thus should not be hardcoded into any platform or application you create which uses the EU-FarmBook API.

As shown in Figure 2 below, to retrieve these tokens, you send a POST request to /api/authentication/token/ with your EU-FarmBook user account email and password credentials included in the request body.

```
curl -X 'POST' \
  'https://api-public.eufarmbook.eu/api/authentication/token/' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "email": "user@example.com",
  "password": "AnExample123!"
}'
```

Figure 2: POST request to /api/authentication/token/ with the EU-FarmBook user account email and password credentials included in the request body, to retrieve tokens.

You will receive a 200-response with the following key/value pairs if you enter your details correctly (see Figure 3 below).

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzA0MzkzMDY2LCJpYXQiOjE3MDQzODk0NjYsImp0aSI6IjdkNjI2ZWIzOTBjOTQ5ZTdhM2M3NDhhN2NiY2FkNzk4IiwidXNlcl9pZCI6MywidG9rZW5fdmVyc2lvbiI6NX0.2IS12bhJZUUvQnN9a-KLxwEExzFAblzgr-GlolMlgwc",
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTcwNDQ3NTg2NiwiaWF0IjoxNzA0Mzg5NDY2LCJqdGkiOiJlYjQ4OTc0ZjBlNTc0ZWZhYTUzYTE2NTk1ZmJkNjQ4YyIsInVzZXJfaWQiOjMsInRva2VuX3ZlcnNpb24iOjV9.4V-PlOkvq623BM8qtH3vPLsldj7NBu3ty2FV5GkzKQM",
  "uuid": "123-456-789",
}
```

Figure 3: Access and refresh tokens.

As shown in Figure 4 below, if you enter your email incorrectly, you will receive a 404 response with an error message, as shown below.

```
{
  "detail": {
    "message": "The specified user account does not exist in our records. Please check your details and try again.",
    "status": "error"
  }
}
```

Figure 4: Message received in case of an incorrect email address.

If you enter your password incorrectly, you will receive a 401 response with an error message, as shown in Figure 5 below.

```
{
  "detail": {
```

```
    "status": "error",
    "message": "The login credentials provided are incorrect. Please check your crede
ntials and try again."
  }
}
```
Figure 5: Message received in case of an incorrect password.

Step 2: Project Access

Provided you have generated your tokens in step 1, you can view the projects for which you are authorised to upload Knowledge Objects.

It is important to remember that every time you upload a Knowledge Object and its metadata, you must include the unique ID EU-FarmBook has created for the project for which you are authorised to upload material. This is a different ID to your project's Digital Object Identifier (DOI) to avoid doubt.

If you do not already know the unique ID EU-FarmBook has created for your project, you can send a POST request with the response from step 1 in the request body (see Figure 6 below).

```
curl -X 'POST' \
  'https://api-public.eufarmbook.eu/api/authentication/projects/' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzI
iwiZXhwIjoxNzA0MzkzMDY2LCJpYXQiOjE3MDQzODk0NjYsImp0aSI6IjdkNjI2ZWIzOTBjOTQ5ZTdhM2M3ND
hhN2NiY2FkNzk4IiwidXNlcl9pZCI6MywidG9rZW5fdmVyc2lvbiI6NX0.2IS12bhJZUUvQnN9a-KLxwEExzF
Ablzgr-GlolMlgwc",
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVz
aCIsImV4cCI6MTcwNDQ3NTg2NiwiaWF0IjoxNzA0Mzg5NDY2LCJqdGkiOiJlYjQ4OTc0ZjBlNTc0ZWZhYTUzY
TE2NTk1ZmJkNjQ4YyIsInVzZXJfaWQiOjMsInRva2VuX3ZlcnNpb24iOjV9.4V-PlOkvq623BM8qtH3vPLsld
j7NBu3ty2FV5GkzKQM",
  "uuid": "123-456-789",
}
'
```
Figure 6: Post request to retrieve the unique ID of the project to upload Knowledge Objects.

Provided you enter your details correctly, you will receive a 200-response with a list of one or more Project IDs (project_id) and corresponding Project Names (project_name) for which you have contributor access. This is shown in Figure 7 below.

```
[
  {
    "project_id": "6787a952b6ad26cff14650c4492e",
    "project_name": "Example Project A"
  },
  {
    "project_id": "89498f9b756fgg466edcg4fc0c68",
    "project_name": "Example Project B"
  }
]
```
Figure 7: Retrieval of project IDs.

If you entered an incorrect or expired token, you will receive a 401 error message, as shown in Figure 8 below, with the following response.

```
{
```

EU-FarmBook

```
    "detail": "The tokens are invalid or expired. Please login again."
}
```
Figure 8: Message received in case of providing a wrong token.

Step 3: Knowledge Object Upload

Now that you have access tokens and project ID, you can contribute Knowledge Objects and metadata.

The first stage is to upload the digital file you want to contribute to the EU-FarmBook platform and make it publicly available.

To do so, send a POST request including your access tokens from Step 1, the specific project ID from Step 2, and the file itself. This is shown in Figure 9 below.

```
curl -X 'POST' \
  'https://api-
public.eufarmbook.eu/api/upload/knowledge_object_file?user_tokens=%7B%20%20%20%22acce
ss_token%22%3A%20%22eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNz
IiwiZXhwIjoxNzA0MzkzMDY2LCJpYXQiOjE3MDQzODk0NjYsImp0aSI6IjdkNjI2ZWIzOTBjOTQ5ZTdhM2M3N
DhhN2NiY2FkNzk4IiwidXNlcl9pZCI6MywidG9rZW5fdmVyc2lvbiI6NX0.2IS12bhJZUUvQnN9a-
KLxwEExzFAblzgr-
GlolMlgwc%22%2C%20%20%20%22refresh_token%22%3A%20%22eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXV
CJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTcxNDA1MzQ5OSwiaWF0IjoxNzEzOTY3MDk5LCJqd
GkiOiIzMzc1MzEwOGY0Mzg0OTk4Y0tk1YmU5NWUwYWFlNWRiZjcyZSIsInVzZXJfaWQiOjY0MzY2LCJ0b2tlbl92ZX
JzaW9uIjoxfQ.IwrsgJDHyF_j1brtP7_9YwQtxMSkS8haCQU-
qK4HpWM%22%2C%20%20%20%22uuid%22%3A%20%22123-456-
78922%20%7D&project_id=6787a952b6ad26cff14650c4492e ' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'ufile=@images.jpeg;type=image/jpeg'
```
Figure 9: Knowledge Object file upload.

Important: As you are now uploading a digital file and not simple JSON content, the Content-Type in your request is multipart/form-data. As a result, tokens cannot be included in the request body and are instead provided similarly to the project ID. You can do this by transforming the JSON response from Step 1 into a JSON-formatted string. This is possible, for example, by using the json.dumps() function in Python (see here).

Step 4: Metadata Validation and Upload

The final step to make your Knowledge Object available in the EU-FarmBook platform is uploading the metadata.

Please first review the EU-FarmBook metadata guide available from the Support page of the EU-FarmBook platform here to read more about the needed metadata properties and values, and ensure you understand which properties are mandatory.

Metadata is provided in the format of a JSON document. By accessing the online documentation page for the EU-FarmBook API endpoints (https://api-public.eufarmbook.eu/docs) and looking at the example value in the endpoint **api/upload/validate_knowledge_object_metadata**; you can see the correct structure to follow in line with the metadata guide.

It is encouraged that before using the **api/upload/knowledge_object_metadata** endpoint, you first validate that your metadata is in the correct format and structure using the allowed values without any formatting errors.

There is an endpoint **api/upload/validate_knowledge_object_metadata,** which you can use by posting the same content as you would for your actual metadata upload.

If you choose not to do this, you may find some of your uploads are rejected, and you will need to parse the error messages to check which to re-upload and which have already been accepted.

By validating metadata for your entire batch and fixing any issues in advance, you can be confident that the actual upload will work for all objects first and save manual effort.

When you are ready to validate your metadata, you send this in another POST request, again including your access tokens from step 1 in the request body and the specific project ID as a URL parameter. The example in the request shown in Figure 10 below is purely for demonstration purposes.

```
curl -X 'POST' \
  'https://api-
public.eufarmbook.eu/api/upload/validate_knowledge_object_metadata?project_id=6787a95
2b6ad26cff14650c4492e \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "user_tokens": {
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzI
iwiZXhwIjoxNzA0MzkzMDY2LCJpYXQiOjE3MDQzODk0NjYsImp0aSI6IjdkNjI2ZWIzOTBjOTQ5ZTdhM2M3ND
hhN2NiY2FkNzk4IiwidXNlcl9pZCI6MywidG9rZW5fdmVyc2lvbiI6NX0.2IS12bhJZUUvQnN9a-KLxwEExzF
Ablzgr-GlolMlgwc",
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVz
aCIsImV4cCI6MTcwNDQ3NTg2NiwiaWF0IjoxNzA0Mzg5NDY2LCJqdGkiOiJlYjQ4OTc0ZjBlNTc0ZWZhYTUzY
TE2NTk1ZmJkNjQ4YyIsInVzZXJfaWQiOjMsInRva2VuX3ZlcnNpb24iOjV9.4V-PlOkvq623BM8qtH3vPLsld
j7NBu3ty2FV5GkzKQM",
  "uuid": "123-456-789",
},
  "metadata": {
    "knowledge_objects": [
      {
        "database_id": "A database ID from the response in Step 3",
        "language": "English"
      },
      {
        "database_id": "Another database ID from the response in Step 3",
        "language": "Dutch"
      }
    ],
    "title": "Your Knowledge Object Title",
    "doi": "Your Knowledge Object Digital Object ID",
    "description": "Your Knowledge Object Description",
    "keywords": [
      "Keyword 1",
      "Keyword 2",
      "Keyword 3"
    ],
    "creators": [
      {
        "name": "Creator 1",
```

```
      "email": "creator1@test.com"
    },
    {
      "name": "Creator 2",
      "email": "creator2@test.com"
    }
  ],
  "date_of_completion": "01-01-2023",
  "intended_purpose": [
    "Dissemination"
  ],
  "geographic_locations": [
    "Netherlands"
  ],
  "category": "Factsheet",
  "type": [
    "Document"
  ],
  "topics": [
    "Crop farming"
  ],
  "subtopics": [
    "Agroecology",
    "Biodiversity and nature management",
    "Organic farming"
  ],
  "license": "CC BY",
  "contributor_custom_metadata": {
    "An example custom property": [
      "An example value"
    ]
  }
 }
}'
```

Figure 10: Metadata validation.

If you have uploaded invalid metadata, you will receive specific JSON response messages that will clarify the issue and how to fix it.

Three common scenarios may result in your receiving such an **error messag**e:

- Using an unaccepted value for a metadata property based on a fixed list of allowed inputs (e.g. topics, subtopics, licences).

- Failing to include a mandatory metadata property (e.g., missing the subtopic property (and value(s)).

- Using the incorrect format for a metadata property's value (e.g., date or email address formatting).

In the scenario in which you have chosen a metadata property value not mentioned in the EU-FarmBook Metadata guide (for example, you chose 'Farm' as a topic rather than one of the expected topics), you will receive a 422 error with the response shown in Figure 11 below.

```
{
  "detail": [
    {
      "loc": [
        "body",
```

```
        "metadata",
        "topics",
        0
    ],
    "msg": "Farm is not an allowed topic. Topic must be from the following list: ['
Forestry', 'Livestock', 'Crop farming', 'Economics', 'Environment', 'Society']",
    "type": "value_error"
    }
  ]
}
```

Figure 11: Message received in case of an incorrect metadata property value (incorrect value to the "topic" metadata property).

Now, take the scenario in which you have also forgotten to include the subtopics property in your metadata. You will then receive the same 422 error with an additional message similar to the one shown in Figure 12 below.

```
{
  "detail": [
    {
      "loc": [
        "body",
        "metadata",
        "topics",
        0
      ],
      "msg": "Farm is not an allowed topic. Topic must be from the following list: ['
Forestry', 'Livestock', 'Crop farming', 'Economics', 'Environment', 'Society']",
      "type": "value_error"
    },
    {
      "loc": [
        "body",
        "metadata",
        "subtopics"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

Figure 12: Message received in case of an incorrect metadata property value (incorrect value to the "subtopic" metadata property).

Finally, take the scenario that you have used the incorrect date format for the "date_of_completion" property and an incorrectly formatted email address for a creator. In such a case, you will receive a 422 error with the response shown in Figure 13 below.

```
{
  "detail": [
    {
      "loc": [
        "body",
        "metadata",
        "creators",
        0,
        "email"
      ],
      "msg": "value is not a valid email address",
      "type": "value_error.email"
    },
```

```
{
    "loc": [
        "body",
        "metadata",
        "date_of_completion"
    ],
    "msg": "Invalid date format, should be DD-MM-YYYY",
    "type": "value_error"
    }
  ]
}
```

Figure 13: Message received in case of incorrect metadata property values (incorrect values to the "email address" and "date of completion" metadata properties).

Assuming you are using the "api/upload/validate_knowledge_object_metadata" endpoint and have fixed these errors, you will receive a 200-response code confirming that the metadata structure and values are correct. The message will read, "Metadata validated successfully".

You can then use the **api/upload/knowledge_object_metadata** endpoint to upload your metadata to the EU-FarmBook.

Please note if you have included an incorrect or invalid document_id, you will receive a 404 error with the response shown in Figure 14 below.

```
{
    "detail": {
        "detail": "Documents not found"
    }
}
```

Figure 14: Message received in case of an incorrect or invalid document_id.

Provided you have used a correct database_id, taken from the response in step 3, you will receive a 200-reponse with a new unique ID of the knowledge object and metadata you have just uploaded (see Figure 15 below).

```
{
    "_id": "your_id_here"
}
```

Figure 15: Knowledge Object ID returned after the successful upload of a Knowledge Object.

6. Accessing your Knowledge Object

Accessing via the EU-Farmbook platform

Once you have completed uploading your Knowledge Object(s) and metadata, you can immediately find it via the https://eufarmbook.eu platform. You can do this via the search or your project page or use the ID you received in the response to take you directly to the specific contribution page, e.g.

https://eufarmbook.eu/en/contributions/your_id_here

EU-FarmBook

You can also log into your account via the EU-Farmbook website here and visit the "Uploaded Contributions" page by clicking on your initials in the top right-hand corner of any page and selecting "Uploaded Contributions" from the dropdown menu.

Accessing via the EU-FarmBook API

Your project may wish to integrate Knowledge Objects and metadata into your platform or website or simply all the metadata you have provided for another purpose.

You can send a GET request to an API endpoint using your access tokens and project ID. If you have the correct access, you will receive a response of all Knowledge Objects and metadata in JSON-LD format (see Figure 16 below).

```
curl -X 'POST' \
  'https://api-public.eufarmbook.eu/api/project/knowledge_object_metadata?project_id=
6787a952b6ad26cff14650c4492e' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzA0Mzk
zMDY2LCJpYXQiOjE3MDQzODk0NjYsImp0aSI6IjdkNjI2ZWIzOTBjOTQ5ZTdhM2M3NDhhN2NiY2FkNzk4Iiwi
dXNlcl9pZCI6MywidG9rZW5fdmVyc2lvbiI6NX0.2IS12bhJZUUvQnN9a-KLxwEExzFAblzgr-GlolMlgwc",
  "refresh_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTcwNDQ
3NTg2NiwiaWF0IjoxNzA0Mzg5NDY2LCJqdGkiOiJlYjQ4OTc0ZjBlNTc0ZWZhYTUzYTE2NTk1ZmJkNjQ4YyIs
InVzZXJfaWQiOjMsInRva2VuX3ZlcnNpb24iOjV9.4V-
PlOkvq623BM8qtH3vPLsldj7NBu3ty2FV5GkzKQM",
  "uuid": "123-456-789",
}'
```

Figure 16: Request to API endpoint using access tokens and project ID to retrieve Knowledge Objects and their associated metadata.

As shown in Figure 17 below, you will receive a 404 error with the response below if you have not yet uploaded a Knowledge Object.

```
{
  "detail": "Documents not found"
}
```

Figure 17: Error message received in the case of a GET request if there is no Knowledge Object available.

You will receive a 200 response with the JSON-LD document in the response body if you have already uploaded one or more Knowledge Objects. In addition to be able to view all the metadata you provided, the value attached to the key @id property gives the URL of the physical location of the Knowledge Object.

7. Questions and Issues

If you have any questions or experience issues using the EU-FarmBook API or have any requests for future features, please use one of the following options.

- Visit our support pages <u>here</u> and check the FAQs section.

- Use the contact form at the bottom of our support pages <u>here</u>.

- Contact your EU-FarmBook Ambassador.

8. API Case Studies

Before the pre-launch event of the EU-FarmBook platform in February 2024, several Horizon 2020, Horizon Europe, and Operational Group projects began working with the EU-FarmBook to integrate their Knowledge Objects via the API. These projects all had different use cases for the EU-FarmBook platform, and they approached collecting and providing the metadata in a way that best suited their requirements.

These projects kindly agreed to work with the EU-FarmBook to build case studies based on their approach to making this public and the code used to integrate their Knowledge Objects into the EU-FarmBook platform via the API.

Some of these case studies are ongoing, and we will thus continually update this documentation to include them.

Case Study 1: ResAlliance

Project background

ResAlliance: Growing solutions to achieve resilient Mediterranean landscapes

An estimated 40 % of Europe's land surface is forested. Also, farms in the EU covered more than a third of the total EU land area in 2020. No doubt, farmers and foresters are key to landscape resilience. The EU-funded ResAlliance project will facilitate information and knowledge flow and increase the awareness, understanding and capacity of farmers and foresters on landscape resilience in the countries of the Mediterranean basin. More specifically, researchers will assess knowledge and practice needs with special emphasis on measures against hazards caused by climate change. The project will also create a Mediterranean thematic network on landscape resilience for forestry and agriculture: LandNet. Its members will exchange technological, managerial, financial and governance solutions through different channels and events while growing new cooperations and networks.

See CORDIS for more information here.

Requirements and use case

From March 2024, ResAlliance intends to produce Knowledge Objects at regular intervals and make them available via the EU-FarmBook in different languages.

Rather than manually contributing each Knowledge Object via the EU-FarmBook upload form, ResAlliance plans to build an Excel file that consortium partners and Knowledge Object creators can populate periodically as they are produced.

ResAlliance is also building its project website but will not store its Knowledge Objects on its servers. Instead, they plan to use the EU-FarmBook API as the effective 'back-end' of their website, whereby the EU-FarmBook will host their content and allow them to access it directly. This gives the added benefit of not needing to build their own object and metadata databases and ensuring their outputs stay available after the ResAlliance project and its website are closed.

Finally, ResAlliance plans to collect its project metadata outside the mandatory and optional properties of EU-FarmBook metadata. ResAlliance will use the EU-FarmBook's API feature to include custom metadata, which it can access and make available via its project website.

Approach

ResAlliance created an Excel template where each row corresponds to a Knowledge Object or collection of Knowledge Objects in different languages that share the same metadata.

Every time a Knowledge Object is produced, generally in the format of a PDF document, the responsible project partner creates a new row in the Excel template and completes all the metadata required by the EU-FarmBook, as well as the custom metadata properties ResAlliance wish to store for their purposes.

Whenever a metadata property had multiple values, for example, keywords or subtopics, ResAlliance used a semi-colon (";") to delimit each value, for example, "keyword1; keyword2; keyword3".

By April 2024, ResAlliance had already created their first batch of 6 Knowledge Objects and completed their metadata Excel file accordingly. In collaboration with EU-FarmBook, Python scripts were built and tested.

These scripts had two primary purposes:

- Using the Python data analysis library Pandas (https://pandas.pydata.org/) to extract the metadata from the Excel file and transform it into the correct JSON structure required by the EU-FarmBook.

- Using the Python library Requests (https://pypi.org/project/requests/) to make the necessary API calls to authenticate, find projects and upload Knowledge Objects and Metadata.

The scripts were setup in a way in which all that would be needed to reuse them in the future for new Knowledge Objects would be to save down the new Knowledge Objects and metadata Excel file in a specific location and update the script to point to the name of the excel file to be processed.

Integrating ResAlliance website with the EU-FarmBook

As mentioned, ResAlliance is using the EU-FarmBook as its website's 'back-end' database. The technical partner responsible for building the website has integrated the code introduced in section 6. This now means that as soon as new Knowledge Objects are uploaded to the EU-FarmBook via the API or upload form, they immediately become available via ResAlliance's website. You can see that integration here: https://www.resalliance.eu/collection-of-knowledge/

Access to scripts

ResAlliance has kindly allowed the EU-FarmBook to publish the scripts for their integration in GitHub. You can get access to these scripts here and see the readme.md file inside the project to get more information and learn how to test the scripts.

Case Study 2: Grazing4AgroEcology

Project background

Grazing4AgroEcology: The importance of grazing-based farms

Farmers, animals, and society benefit from grazing-based production systems. However, grazing, which helps yield high-quality food and contributes to enhancing other ecosystem services, is declining in Europe. To hinder this, the EU-funded Grazing4AgroEcology (G4AE) project collaborates with grazing farmers and other stakeholders to find solutions for sustainable, integrated grazing-based animal production systems. Part of this includes sharing knowledge and capturing, introducing, and enhancing best practices and innovations to promote grazing for agroecology. The work will support farmers by fuelling their understanding of their agroecological performance through an integrated self-assessment and encouraging them to innovate.

See CORDIS for more information here.

Requirements and use case

From March 2024, Grazing4AgroEcology intends to produce Knowledge Objects regularly throughout their project lifecycle.

Rather than manually contributing each Knowledge Object via the EU-FarmBook upload form, Grazing4AgroEcology plans to build individual Excel files that consortium partners and creators can populate each time a new Knowledge Object is created.

Approach

Grazing4AgroEcology created an Excel template using VBA macros to help users ensure they can only select metadata property values from those stated in the EU-FarmBook metadata guide.

Each time a new Knowledge Object digital file is created, normally in the format of a PDF document or video (.mp4), the responsible project partner makes a copy of the Excel template and gives the copy the same name as the Knowledge Object it corresponds to.

Because of the use of macros to validate the data, users of the template can take advantage of the dropdown lists which appear for the metadata properties based on pre-defined allowed values.

Whenever a metadata property has multiple values, for example, keywords or subtopics, Grazing4AgroEcology uses a semi-colon (";") to delimit each value, for example "keyword1; keyword2; keyword3".

By April 2024, Grazing4AgroEcology had created its first batch of 5 Knowledge Objects with their corresponding five metadata Excel files. In collaboration with EU-FarmBook, a set of Python scripts was programmed.

These scripts had two primary purposes:

- Using the Python data analysis library Pandas (https://pandas.pydata.org/) to extract the metadata from each Excel file and transform it into the correct JSON structure required by the EU-FarmBook.

- Using the standard Python library Requests (https://pypi.org/project/requests/) to make the necessary API calls to authenticate, find projects and upload Knowledge Objects and Metadata from each Excel file.

Because the Excel file names already correspond to the file names of each Knowledge Object, the scripts can be easily reused for new Knowledge Objects once they are saved in the specific location without any manual changes required.

Access to scripts

Grazing4AgroEcology has kindly allowed the EU-FarmBook to publish the scripts for their integration in GitHub. You can get access to these scripts here and see the readme.md file inside the project to get more information and learn how to test the scripts.