Introduction to sequence databases and data sharing - practical exercises

Intended learning outcomes:

- Become familiar with the ENA submission portal.
- Be able to upload data to the ENA TEST database
- Know how to search for and download data from ENA

The exercise consists of two parts:
1. uploading data to ENA
2. retrieving data from ENA.

Optional exercises are available at the end of the document, if you have extra time (or at a later time). A section with links to further documentation and reading is also included at the end of the document.

Pre-course activities:

72 hours prior to the exercise, each participant should have created its own submission account by following this link: https://www.ebi.ac.uk/ena/submit/webin/accountInfo

Data used during the course:

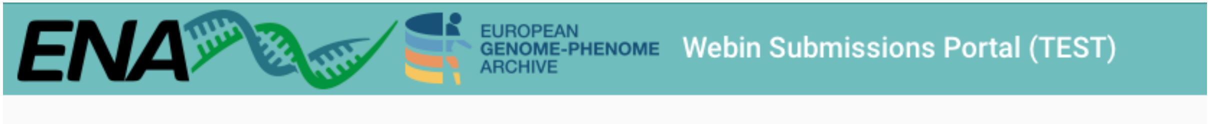
The data used in the exercise is located in the following location:
`BTG/SequenceData/`

There are both illumina fastq files (Ec0xx.illumina_Rx.fastq.gz) and a metadata file (Ecoli_metadata.tsv)

PART 1: Uploading raw reads to ENA

All the exercises will be performed on the **TEST submission server.** Test submissions are automatically removed after 24hrs, while submissions to the production server are *permanent*. Please inform the trainers if data has been erroneously sent to the production server.
To log on to the TEST Webin portal.
https://wwwdev.ebi.ac.uk/ena/submit/webin/login

It should show (TEST) on the top ribbon of the page:

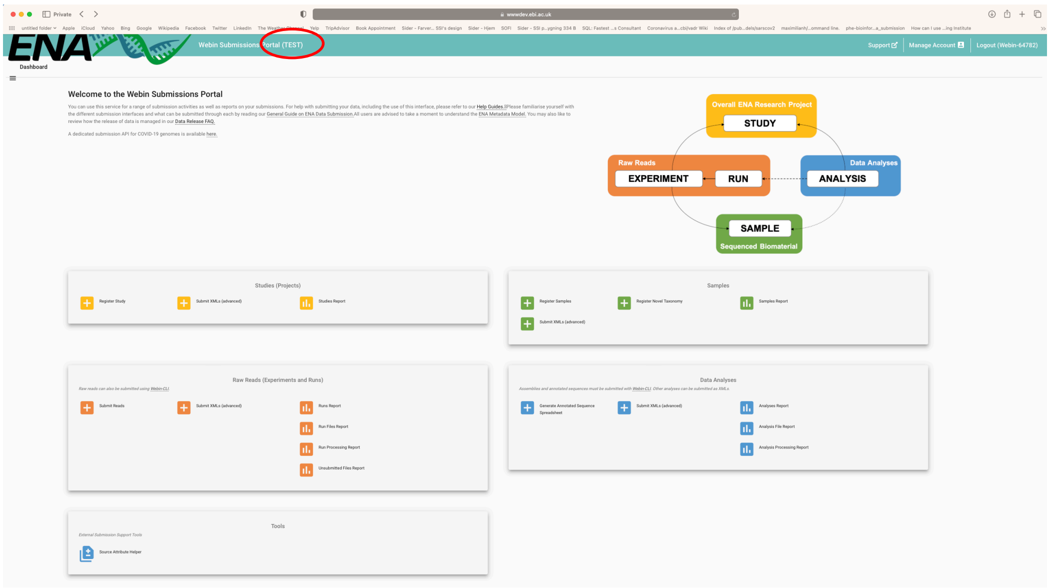


Exercise 1: Submitting raw reads through the interactive route

Step 1: Create a new project

Make sure that you are logged on the TEST server

https://wwwdev.ebi.ac.uk/ena/submit/webin/login



Welcome to the Webin Submissions Portal

You can use this service for a range of submission activities as well as reports on your submissions. For help with submitting your data, including the use of this interface, please refer to our Help Guides. Please familiarise yourself with the different submission interfaces and what can be submitted through each by reading our General Guide on ENA Data Submission. All users are advised to take a moment to understand the ENA Metadata Model. You may also like to review how the release of data is managed in our Data Release FAQ.

A dedicated submission API for COVID-19 genomes is available here.

Raw reads can also be submitted using Webin-CLI.

Assemblies and annotated sequences must be submitted with Webin-CLI. Other analyses can be submitted as XMLs.

1- Click on 'Register Study' and fill in the information. The creation requires you to set a 'Release date'.
Pick one in the calendar that is different from today's date (the project will not persist more than 24 hours on the test server but let us set a date knowingly. Every data submitter should be aware that he has the option to control when the data shall be publicly released). Add a short title and description and fill in the abstract field.

2- Submit the project. If you have created your submission account more than 72 hours prior to the exercise, you will receive a successful project creation pop-up message. In the project creation message, there will be a PREJBxxxx accession ID,

Write down the PRJxxxx down BioProject accession ID it will be needed for the submission exercise.

In case the project had been erroneously created in the production server, you will have to send a service ticket to ENA with the study accession ID to request its removal. To do this, click on “Support” (top right of the window) and follow instructions. Specify that you are the “account holder” and “I have a query/issue” related to “submission”.

Step 2: Add new samples to the newly created project.

Samples can be added to a project programmatically or interactively. This exercise will show you the process using the interactive route.

Make sure that you are logged on the TEST server

1- Go to the Webin portal and click on 'Register Samples'

2- Download the spreadsheet corresponding to the standard checklist for microbial samples ERC000029 (**ENA Global Microbial Identifier reporting standard checklist GMI_MDM:1.1**), under the “Pathogens checklist” tab. .
For information, all checklists are described in the following webpage:
https://www.ebi.ac.uk/ena/browser/checklists

3- Fill in the sample spreadsheet for using the provided sample metadata.
The required fields are shown in green in the checklist page, look for ERC000029

Please check if the following long column names are formatted correctly. If these names are spelt with underscores in the template, you have just downloaded, please use the correct spelling shown below (transient issue with the template):

Is the sequenced pathogen host associated?
isolation source host-associated

isolation source non-host-associated

geographic location (country and/or sea)

geographic location (longitude)

geographic location (latitude)

For the values not provided by the metadata file use the following:

Use the **taxon id 562** for Escherichia coli.
Set the column **Environmental sample** to **No**

Set host health state to **not provided** (this field is **case-sensitive**)
Set ‘Is the sequenced pathogen host associated?’ to **Yes**
Set isolation source host associated to **Gastric biopsies**
Set isolate to Ec0xx_illumina (xx is the sample id)

Mandatory fields with no data cannot be left blank (remove columns if there is no available information)

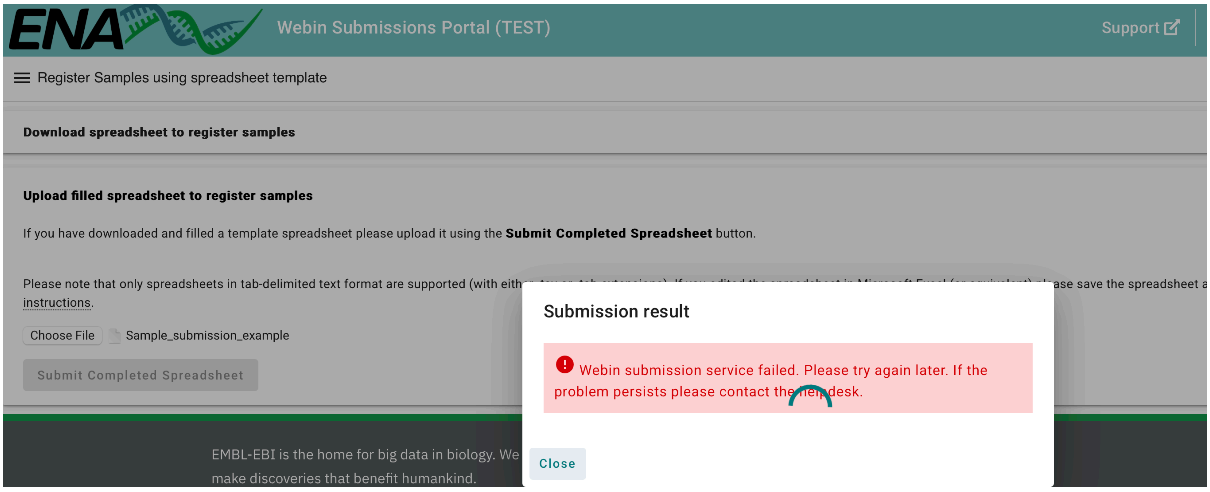
Include only a few samples (e.g. Ec001 to Ec004) so that you can try the other submission methods if you have time. Samples can only be submitted once.

Save the file in csv (or tsv) format.

4- Upload spreadsheet to register samples by clicking on “Submit Completed Spreadsheet”

A message will pop-up if the submission is successful. In case of a submission error, please check the formatting of your input.
You may check the newly registered samples by clicking on “Sample Report”

Note:
The spreadsheet will be checked by the submission system for valid data formats, e.g. correct column names, key words, date format and is case sensitive for mandatory fields. An invalid input will return the error screen shown below:



Download spreadsheet to register samples

Upload filled spreadsheet to register samples

If you have downloaded and filled a template spreadsheet please upload it using the **Submit Completed Spreadsheet** button.

Please note that only spreadsheets in tab-delimited text format are supported (with eith
ase save the spreadsheet a
instructions.

Submission result

Webin submission service failed. Please try again later. If the problem persists please contact the helpdesk.

EMBL-EBI is the home for big data in biology. We make discoveries that benefit humankind.

Step 3: Submit raw reads to an existing project and samples in the interactive way.

Detailed instructions for uploading files in the interactive way is available on the ENA webpage https://ena-docs.readthedocs.io/en/latest/submit/reads/interactive.html

In this exercise, we will add raw reads to samples that have just been created in step 2. Submission of fastq-files is done in two steps:

1. Fastq-files are transferred to the ENA server
2. A samplesheet is created, which connects fastq filenames (and their md5sums) to the project. Upload is complete when this samplesheet is successfully submitted

Make sure that you are logged on the TEST server

1- Click on “Submit reads”.

The next page will suggest you download a template for the metadata to include in the submission. Click “Download spreadsheet for Read submission”.

You will be submitting paired-end Illumina raw reads using fastq files, find the correct tsv template and download it with the recommended fields.

2- Check the “Sample report” and write down the sample accession ids SAMEAxxxxxx needed for the submission metadata sheet.

3- Fill in the downloaded template using the provided PRJEBxxxxx project identifier and metadata available metadata. Provide the file names for each fastq file and their matching md5sums. Save as CSV or TSV format.

Here are the values for some of the required fields (in a real case scenario please change to what applies to your run)

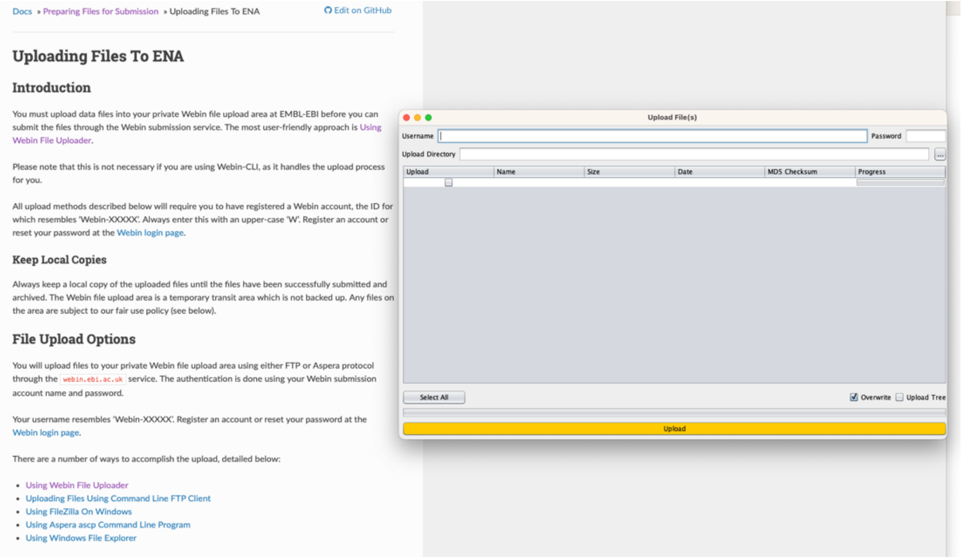
INSTRUMENT_MODEL is: Illumina MiSeq
LIBRARY_STRATEGY: WGS
LIBRARY_SOURCE: GENOMIC
LIBRARY_SELECTION: RANDOM
LIBRARY_NAME: unspecified

4- Now upload the raw reads fastq.gz

For this step, we will need to **use the wireless hotspot (follow the instructions provided by the training facilitator)**.

Files with names matching the samples you included in the metadata sheet need to be uploaded to the test submission server using the **ena file uploader**. Find the icon launcher on the desktop.
Use the Webin uploader with your ENA login information

More information here https://ena-docs.readthedocs.io/en/latest/submit/fileprep/upload.html



Docs » Preparing Files for Submission » Uploading Files To ENA

Edit on GitHub

Uploading Files To ENA

Introduction

You must upload data files into your private Webin file upload area at EMBL-EBI before you can submit the files through the Webin submission service. The most user-friendly approach is Using Webin File Uploader.

Please note that this is not necessary if you are using Webin-CLI, as it handles the upload process for you.

All upload methods described below will require you to have registered a Webin account, the ID for which resembles 'Webin-XXXXX'. Always enter this with an upper-case 'W'. Register an account or reset your password at the Webin login page.

Keep Local Copies

Always keep a local copy of the uploaded files until the files have been successfully submitted and archived. The Webin file upload area is a temporary transit area which is not backed up. Any files on the area are subject to our fair use policy (see below).

File Upload Options

You will upload files to your private Webin file upload area using either FTP or Aspera protocol through the `webin.ebi.ac.uk` service. The authentication is done using your Webin submission account name and password.

Your username resembles 'Webin-XXXXX'. Register an account or reset your password at the Webin login page.

There are a number of ways to accomplish the upload, detailed below:

- Using Webin File Uploader
- Uploading Files Using Command Line FTP Client
- Using FileZilla On Windows
- Using Aspera ascp Command Line Program
- Using Windows File Explorer

If the ena file uploader does not work, one can alternatively upload the raw read files using Aspera.

activate the conda environment BTG_aspera

Then the command below, using your ENA account credentials

```
ascp –QT –l300M –L– <raw read file name> Webin-XXXXX@webin.ebi.ac.uk:.
```

6- Go back to the TEST submission portal and upload the filled spreadsheet. The file will be validated, and you will receive a pop-up message to confirm whether your raw read submission was successful or not.

7- Check your submission. Click on “Runs report” and check one of the records that you have just uploaded to ENA.

If you have time and interest, you may try the programmatic submission methods in the optional exercises (Part 3). Otherwise please move to the data download exercises and come back later.

PART 2: Querying and retrieving data from ENA databases

Exercise 1: Use the ENA browser to search and retrieve data

The ENA browser is a free access resource for read sequence data. Datasets are organized as projects which could be searched using keywords.

Go to the ENA browser page https://www.ebi.ac.uk/ena/browser/home

a) Let us first try to find a record using its accession id.

1-Search the BioProject PRJEB25979. How many samples can you see? What type of data has been uploaded (e.g. reads, assemblies)?
Download the TSV summary of records and click on the button to generate a script to fetch the first 3 records of the list.

2-Download the files of the first record using the browser.

3-Use the terminal to download the data of the first 3 records using the script you just generated.

b) Now let us try the advanced search mode.

We want to search all SARS-CoV-2 (NCBI taxa 2697049) sequences from genomes collected in Denmark between August 1st, 2022 and August 5, 2022.

1- Go to Advanced search and create a query using the filters corresponding to the criteria. Select all fields. How many records did you find?

2- Download the full tsv report.

You can also find step-by-step advanced search examples on the ENA documentation webpage here
https://ena-docs.readthedocs.io/en/latest/retrieval/advanced-search.html
You may try these if you have time.

Exercise 2: Build a command line search query for the search API

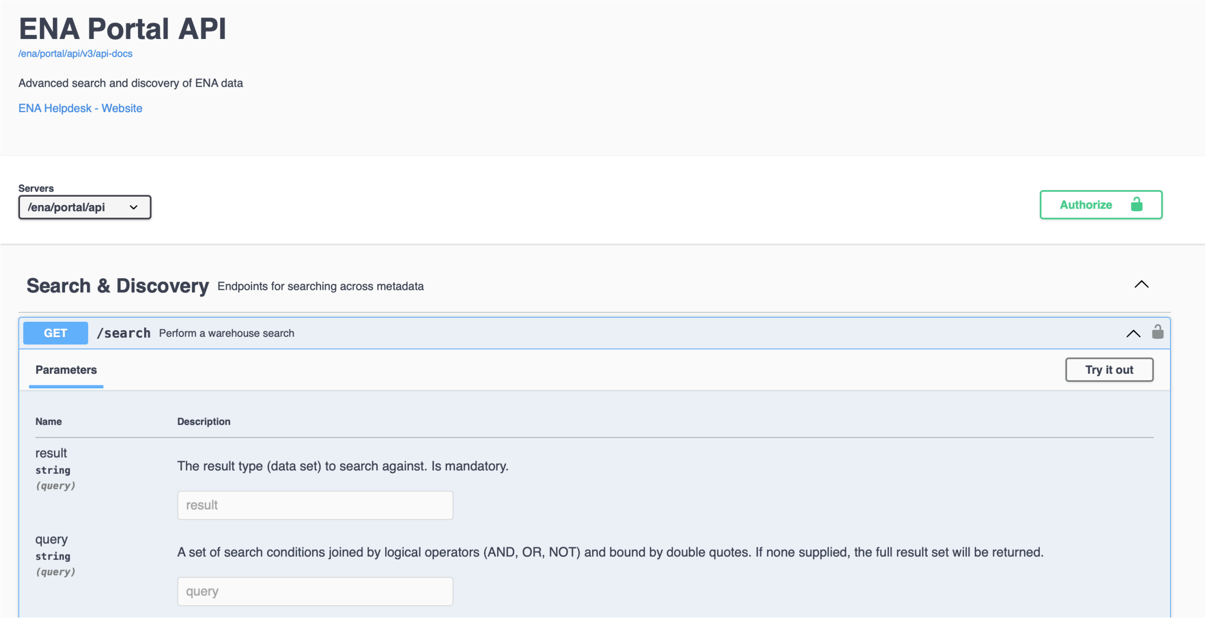
Data queries can be done programmatically using the API. The full documentation is available here
https://docs.google.com/document/d/1CwoY84MuZ3SdKYocqssumghBF88PWxUZ/edit
And on ENA documentation webpage
https://ena-docs.readthedocs.io/en/latest/retrieval/programmatic-access/advanced-search.html

Write the query for the same search query as for the previous exercise (SARS-CoV-2 (NCBI taxa 2697049) sequences from genomes collected in Denmark between August 1st, 2022 and August 5, 2022)

Look at the available keywords for each criterion in the documentation.
We will use the search keywords **tax_id** (taxon id), **collection_date**, **country** and **sequence**.
Submit the query using the Webin API.

https://www.ebi.ac.uk/ena/portal/api/swagger-ui/index.html#/Search%20%26%20Discovery/search

For search queries, authentication is not required (no need to click on “authorize”)



Advanced search and discovery of ENA data

Endpoints for searching across metadata

Perform a warehouse search

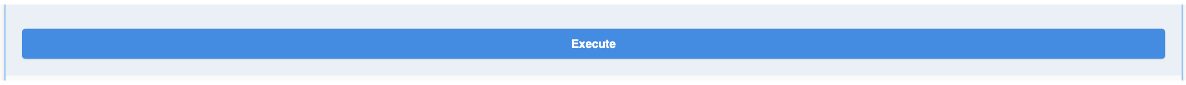
The result type (data set) to search against. Is mandatory.

A set of search conditions joined by logical operators (AND, OR, NOT) and bound by double quotes. If none supplied, the full result set will be returned.

Click on ‘try it out’

Use **sequence** as result type and write the search conditions in the query box

Click execute, the result will show below



The search query can also be submitted directly to the terminal without the interface. The search command is shown on the screen below the button ‘execute’ after submitting the query.

You may also try the example shown in the ENA documentation for a more complex query https://ena-docs.readthedocs.io/en/latest/retrieval/programmatic-access/advanced-search.html#retrieve-raw-read-and-primary-metagenome-datasets-for-cow-rumen-samples-collected-in-the-uk

Exercise 3a: Retrieving data files from ENA resources using the ena-file-downloader

Data files can be retrieved in different ways on ENA. The first exercised showed the ENA browser route.

This exercise will show you how to download data from the terminal using the java-based tool ena-file-downloader. The full documentation is shown here:
https://github.com/enasequence/ena-ftp-downloader/

Open a terminal and test the interactive mode of the java tool by just typing

```
ena–file–downloader
```

Follow instructions and download files linked to the accession number: SAMEA1116772

Now try to run the tool in full command-line mode, the syntax is shown in the documentation

Exercise 3b Retrieving data files from ENA resources using the ena-browser-tools

Optionally, you may try another command-line tool to retrieve data files from ENA. This tool is called `enaDataGet`, see documentation below.
https://ena-docs.readthedocs.io/en/latest/retrieval/programmatic-access/browser-tools.html

Use enaDataGet to download fastq files from this sample SAMEA1116773. An example of the command to run is shown in the documentation.

Exercise 4: Explore the ENA Pathogens portal

This exercise will allow you to explore the ENA Pathogen portal.
Go to https://www.pathogensportal.org

All data deposited on ENA associated to pathogens of interest can be searched using different entry points.

a) Let us first try “Sequences”. This search mode allows to find all sequences and raw reads for different pathogens.

Use the filters on the left columns to find all raw read datasets from Denmark for Listeria monocytogenes sequenced using Illumina.
How many samples do you see?

b) Go to “Outbreak”. Let us look at the data for Monkeypox virus

1-Use the filters in the left column to search all Monkeypox sequences.
Download the metadata for host human collected in the UK between 2021 and 2024

2-Look at the Nextstrain phylogenetic tree using the Nextstrain reports tab.

c) Go to Cohorts.

Click on one cohort and go to the dashboard. The dashboard provides a description and a link to the cohort webpage. The accession number links to the samples that have been collected.

PART 3: Optional exercises

Optional exercises: programmatic raw read submission to ENA

This section provides you with some information to get started with doing larger submissions using the command line.
There are two command-line routes to submit data:

- The XML route
- The json-route

Programmatic submission using the json route can be done using the Webin REST API V1
https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit
The documentation is available here
https://ena-docs.readthedocs.io/en/latest/submit/general-guide/webin-v1.html

Json (and XML) submissions can be done through the new Webin REST V2 API.
https://wwwdev.ebi.ac.uk/ena/submit/webin-v2/
The documentation is available here:
https://ena-docs.readthedocs.io/en/latest/submit/general-guide/programmatic.html

Optional exercise 1: XML programmatic submission

Step 1: Add new samples to an existing project.

1- A sample submission XML must be prepared and submitted to the portal using the command-line.

An example of samplelist.xml file for the food pathogen test sample used in this exercise (checklist ERC00029) could be:

```
<?xml version="1.0" encoding="UTF-8"?>
<SAMPLE_SET>
 <SAMPLE alias="Ec005_testsample_5">
  <TITLE>Test sample Ec005</TITLE>
  <SAMPLE_NAME>
   <TAXON_ID>562</TAXON_ID>
   <SCIENTIFIC_NAME>Escherichia coli</SCIENTIFIC_NAME>
  </SAMPLE_NAME>
  <DESCRIPTION>Original</DESCRIPTION>
  <SAMPLE_ATTRIBUTES>
   <SAMPLE_ATTRIBUTE>
    <TAG>collection date</TAG>
    <VALUE>2018</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>geographic location (country and/or sea)</TAG>
    <VALUE>Denmark</VALUE>
```

```
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>geographic location (longitude)</TAG>
    <VALUE>10.3333283</VALUE>
    <UNITS>DD</UNITS>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>geographic location (latitude)</TAG>
       <VALUE>55.670249</VALUE>
       <UNITS>DD</UNITS>
     </SAMPLE_ATTRIBUTE>
  <SAMPLE_ATTRIBUTE>
    <TAG>host scientific name</TAG>
    <VALUE>Homo sapiens</VALUE>
   </SAMPLE_ATTRIBUTE>
 <SAMPLE_ATTRIBUTE>
    <TAG>host health state</TAG>
    <VALUE>not provided</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>isolate</TAG>
    <VALUE> Ec005_illumina</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
     <TAG>collected_by</TAG>
     <VALUE>SSI</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
     <TAG>environmental_sample</TAG>
     <VALUE>No</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
     <TAG>serovar</TAG>
     <VALUE/>
   </SAMPLE_ATTRIBUTE>
<SAMPLE_ATTRIBUTE>
     <TAG>isolate</TAG>
     <VALUE/>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
     <TAG>serovar</TAG>
     <VALUE/>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
       <TAG>Is the sequenced pathogen host associated?</TAG>
       <VALUE>Yes</VALUE>
     </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
     <TAG>isolation source host-associated</TAG>
     <VALUE>gastric biopsies</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
     <TAG>isolation source non-host-associated</TAG>
     <VALUE/>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>ENA-CHECKLIST</TAG>
    <VALUE>ERC000029</VALUE>
   </SAMPLE_ATTRIBUTE>
  </SAMPLE_ATTRIBUTES>
 </SAMPLE>
</SAMPLE_SET>
```

Note that the fields correspond to those provided using the spreadsheet. Only the formatting is different.
To submit more than one sample one needs to add them to the sample set

```
<SAMPLE_SET>
```

```
  <SAMPLE>
   …
 </SAMPLE>
<SAMPLE>
   …
 </SAMPLE>
  …
</SAMPLE_SET>
```

2- The command-line submission is done using a curl command and requires attaching a submission.xml file containing the lines shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<SUBMISSION>
   <ACTIONS>
      <ACTION>
         <ADD/>
      </ACTION>
   </ACTIONS>
</SUBMISSION>
```

Run the submission command using your account credentials (make sure that the submission drop-box points to 'wwwdev.ebi.ac.uk).

```
curl -u Webin-XXXXX:yourpassword -F "SUBMISSION=@submission.xml" -F
"SAMPLE=@samplelist.xml" https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/ >
sample_receipt.xml
```

Please make sure that you are using the correct environment. The TEST server URL should start with **wwwdev.ebi.ac.uk**

3- After submission of the samples, you will receive an acknowledgement XML.
You will need to parse the sample accessions to submit data to the samples you have just created.

Step 2: Submit raw reads to an existing project and samples in the programmatic way.

If you have already submitted the raw read files in the first exercise using the interactive method, you must make a copy of the fastq.gz files and save the copy using a different name. ENA only allows raw read names with the same names to be submitted once.

Two XML files need to be created: an experiment XML which contains the information related to sequencing and a run XML containing the raw read file information.

An experiment XML (name it for example experimentlist.xml) looks as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXPERIMENT_SET><EXPERIMENT alias="test_experiment_1”>
    <TITLE>test_experiment_1</TITLE>
    <STUDY_REF accession="PRJEBxxxxx" />
    <DESIGN>
     <DESIGN_DESCRIPTION/>
      <SAMPLE_DESCRIPTOR accession="SAMEAxxxxxxxx" />
```

The project ID you created

The sample accession ID you received after step 1

```
      <LIBRARY_DESCRIPTOR>
        <LIBRARY_STRATEGY>WGS</LIBRARY_STRATEGY>
        <LIBRARY_SOURCE>GENOMIC</LIBRARY_SOURCE>
        <LIBRARY_SELECTION>RANDOM</LIBRARY_SELECTION>
      </LIBRARY_DESCRIPTOR>
    </DESIGN>
    <PLATFORM>
      <ILLUMINA>
        <INSTRUMENT_MODEL>Illumina MiSeq</INSTRUMENT_MODEL>
      </ILLUMINA>
    </PLATFORM>
  </EXPERIMENT>
</EXPERIMENT_SET>
```

A run XML (name it for example runlist.xml) looks as fo

Must match the experiment alias

```
<?xml version="1.0" encoding="UTF-8"?>
<RUN_SET><RUN alias="test_1">
        <EXPERIMENT_REF refname="test_experiment_1"/>
        <DATA_BLOCK>
            <FILES>
                <FILE filename="Ec005.illumina_R1.fastq.gz" filetype="fastq"
                    checksum_method="MD5" checksum="6f81bc93b6c10f4f14e8ce4873ab9aa0"/>
                <FILE filename="Ec005.illumina_R2.fastq.gz" filetype="fastq"
                    checksum_method="MD5" checksum="98728144654e021f5bf3f0c30310c856"/>
            </FILES>
        </DATA_BLOCK>
    </RUN>
</RUN_SET>
```

Md5sum of 1st file

Md5sum of 2nd file

IMPORTANT: Before submitting the XMLs, raw read files must be uploaded in advance using for example Aspera

ascp -QT -l300M -L- <raw read file name> Webin-XXXXX@webin.ebi.ac.uk:.

or using the java uploader provided by ENA here:
https://ena-docs.readthedocs.io/en/latest/submit/fileprep/upload.html

After uploading the read files, submit the experiments and runs XMLs as follows:

curl -u Webin-XXXXX:yourpassword -F "SUBMISSION=@submission.xml" -F "EXPERIMENT=@experimentlist.xml" -F "RUN=@runlist.xml" https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/ > exp_run_receipt.xml

Optional exercise 2 : json programmatic upload

This exercise will show you how to make a programmatic submission using json files.

Step1: Submit a sample.json file

A sample submission json must be prepared and submitted to the version 2 REST API.

An example sample.json file for the test data used in the exercise should look like as follows:

```
{
  "submission": {
   "alias": "submission1",
   "accession": "",
   "actions": [
     {
        "type": "ADD"
     },
     {
        "type": "HOLD",
        "holdUntilDate": "2026-01-12"
      }
    ]
  },
  "samples": [
    {
      "alias": "Ec008_test",
      "title": "Ec008",
      "organism": {
        "taxonId": "562"
      },
      "attributes": [


        {
          "tag": "collection date",
          "value": "2015"
        },
        {
          "tag": "isolation source host-associated",
          "value": "gastric biopsies"
        },
        {
          "tag": "environmental_sample",
          "value": "No"
        },
        {
          "tag": "Is the sequenced pathogen host associated?",
          "value": "Yes"
        },
        {
          "tag": "geographic location (longitude)",
          "value": "10.3333283",
          "value": "DD"
        },
        {
          "tag": "geographic location (latitude)",
          "value": "55.670249"",
          "value": "DD"
        },
        {
          "tag": "geographic location (country and/or sea)",
          "value": "Denmark"
        },

        {
          "tag": "host health state",
          "value": "not provided"
        },
        {
          "tag": "host scientific name",
          "value": "Homo sapiens"
        },

        {
          "tag": "isolate",
          "value": "Ec008"
        },
        {
          "tag": "ena-checklist",
```

```
          "value": "ERC000029"
        }
      ]
    }
  ]
}
```

You can submit the sample.json file also using the webin-cli instead of using the terminal and run the command above.

To try it out go to this link,
https://wwwdev.ebi.ac.uk/ena/submit/webin-v2/swagger-ui/index.html
then authenticate using your account credentials. Click on submit, then try it out and attach the file to upload. Press the button 'execute' to run the command.

The windows below show your submission formatted as a command-line and the receipt. Documented errors are also reported.

Check your Sample report on the Webin portal.

Step2: Submit an experiment and run .json file

Before submitting the json files, raw read files must be uploaded in advance using for example Aspera

```
ascp -QT -l300M -L- <raw read file name> Webin-XXXXX@webin.ebi.ac.uk:.
```

make a copy of the files using a different name from the first 2 submissions and use this name when you fill in the file information in the json.

An experiment and run json file for our example should look as follows:

```
{
   "submission":{
      "alias":"subs_test-alias-117",
      "accession":"",
      "actions":[
         {
            "type":"ADD"
         },
         {
            "type":"HOLD",
            "holdUntilDate":"2025-01-01"
         }
      ],
      "attributes":[
         {
            "tag":"test_tag",
            "value":"test_val"
         },
         {
            "tag":"test_tag_1",
            "value":"test_val_1"
         }
```

```
      ]
   },
   "runs":[
      {
         "alias":"run_alias_1",
         "identifiers":null,
         "centerName":"Statens Serum Institut",
         "title":"test run 1",
         "description":"run for experiment alias 1",
         "experiment":{
            "alias":"experiment_alias_1"
         },
         "instrumentPlatform":"ILLUMINA",
         "instrumentModel":"Illumina MiSeq",
         "files":[
            {
                "fileName":"Ec008.illumina_R1.fastq.gz",
                "fileType":"fastq",
                "checksumMethod":"MD5",
                "checksum":"4a094170bcbae43eac16cbbd63dd47ab"
             },
            {
                "fileName":"Ec008.illumina_R2.fastq.gz",
                "fileType":"fastq",
                "checksumMethod":"MD5",
                "checksum":"9f09122dab6872ff23b5d1dfc69ef087"
             }

         ]
      }
   ],
   "experiments":[
      {
         "alias":"experiment_alias_1",
         "identifiers":null,
         "centerName":"Statens Serum Institut",
         "title":"test experiment 1",
         "study":{
            "accession":"PRJEBxxxxx"
         },
         "samples":[
            {
               "accession":"SAMEAxxxxxxxxxx"
            }
         ],
         "designDescription":"",
         "libraryDescriptor":{
            "libraryStrategy":"WGS",
            "librarySource":"GENOMIC",
            "librarySelection":"RANDOM",
            "libraryLayout":"PAIRED"
         },
         "instrumentPlatform":"ILLUMINA",
         "instrumentModel":"Illumina MiSeq"
      }
   ]
}
```