



ESCAPE

European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

Open-source Software and Service Repository

Thomas Vuillaume*, for the OSSR working group

**LAPP, CNRS, thomas.Vuillaume@lapp.in2p3.fr*

Open Science Days, Berlin, 29/01/2024





FAIR

CERN

cta
Cherenkov
telescope
array

EGO
EUROPEAN
GRAVITATIONAL
OBSERVATORY

SKAO

ESCAPE

European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

ES+O

ET
EINSTEIN
TELESCOPE

EST
european solar telescope

KM3NeT

JIVE
Joint Institute for VLBI
ERIC

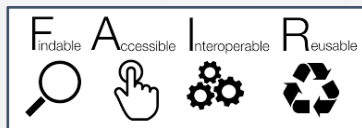


Rationale for ESCAPE

The H2020 cluster concept introduced by the European Commission, in 2018 was aimed at supporting:

- **“Open-science data-intensive research”** in order to **“rise productivity of researchers and to lead to new insights and innovation”**
- Commit in **Open Science** that means implement the **FAIRness** of scientific data
- Connecting ESFRI and other world-class RIs to **EOSC – European Open Science Cloud**
- **ESCAPE** is one of the five Science-Cluster projects that resulted from the H2020 topic call INFRAEOSC-04-2018
Other Science Clusters: **ENVRI-FAIR** (Environment and Earth Sciences), **EOSC-LIFE** (Biomedical Science), **PANOSC** (Neutron and light sources facilities) and **SSHOC** (Social Science and Humanities).

ESCAPE, and all clusters, are now Open Collaborations, with MoUs signed by all partners RIs → long-term structures





ESCAPE Open Collaboration Work Programme

Entities

ESCAPE CC
 Operating the community-based “Competence Center” for EOSC-alignment, train and support, extended outreach, financial model for services and networking with other SCL-CCs

ESCAPE EVSI
 R&I for an “European Virtual Institute for Research Software” for advanced technologies

ESCAPE VRE Virtual Research Environment
 A web-based highly-composable research-enabling interoperable data platform

Working Groups

ESCAPE DIOS Data Infrastructure for Open Science
 Access physical & e-infrastructures
 Processing & Analysis
 Security & Operations



ESCAPE OSSR Open-source Scientific Software and Service Repository
 Aggregator & Integrators
 Sharing and Discover
 Training & Support

ESCAPE ESAP ESFRI Science Analysis Platform
 Processing & Analysis
 Sharing and Discovery
 Training & Support

ESCAPE CS Citizen Science
 Sharing and Discovery

ESCAPE VO Virtual Observatory
 Processing & Analysis
 Sharing and Discovery
 Training & Support

Programmes

ESCAPE COSO
 Challenging “Open Science Objectives” by RI commitments in Open Science Projects (OSP) as well as Cross-Cluster Open Science Projects (COSP)

ESCAPE TECH
 Bring the FAIRness within technology, R&D and innovation projects as well as explore new “close-to-sensors” low-latency open-data science

ESCAPE CARS
 Career development and rewarding for researcher committing in Open Science. Planning, tracking, and assessing scientific knowledge production

ESCAPE SDSS
 Building synergies on “Sector Data Spaces” for Society: Green deal, Health, Manufacturing, Education and Skills



OSSR main goals:

- Provide and archive good quality software in the astronomy and particle physics communities
- Provide a forum for ESFRIs to exchange and discuss about the software they produce
- Train researchers and developers to good code development and sharing practices
- Level-up the software quality in our communities

Open-source and FAIR software

Importance of Open and FAIR software

- Reproducibility of results
- Increased trust in the results
- Increased impact
- Recognition
- Re-use

FAIR principles for research software

Barker, M., Chue Hong, N.P., Katz, D.S. *et al.* Introducing the FAIR Principles for research software. *Sci Data* 9, 622 (2022). <https://doi.org/10.1038/s41597-022-01710-x>



Scriberia 

The Turing Way Community, & Scriberia. (2023). Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. Zenodo. <https://doi.org/10.5281/zenodo.8169292>



Community software

- Communities need a place to share software in the FAIR context
 - Findable
 - Accessible
 - Interoperable
 - Reusable
- A “simple” catalog is not enough
 - FAIR?
 - long-term accessibility of the software?
 - software quality?
 - interoperability and (re)usability?



How to

create a trusted software repository
and globally improve software quality

in a community?

[Records](#) [Requests](#) [Members](#) [Settings](#) [Curation policy](#) [About](#)

48 results found

Sort by [December 11, 2023 \(0.11.3\)](#) [Software](#) [Open](#)**MOC Lib Rust, MOCCLi, MOCWasm and MOCSet**Pineau, Francois-Xavier  Baumann, Matthieu

Rust implementation of the IVOA MOC standard (MOC Lib Rust); associated command line tool (MOCCLi) and Javascript/WebAssembly wrapper to manipulate MOCs in Web Browsers (MOCWasm).

Uploaded on December 20, 2023

6 more versions exist for this record

 184  29[December 4, 2023 \(v0.13.1\)](#) [Software](#) [Open](#)**cds-astro/mocpy: Release v0.13.1**

Matthieu Baumann; Manon Marchand; Francois-Xavier Pineau; and 6 others

What's Changed Mostly maintenance to support astropy 6.0 and python 3.12 while maintaining support for python 3.8 These points have changed internal behaviour, or documentation: Add missing return statement in private abstract class AbstractMOC in <https://github.com/cds-astro/mocpy/pull/112> The deprecated method write now calls save intern...

Uploaded on December 4, 2023


5 more versions exist for this record

 158  17[December 4, 2023 \(v2.0.0\)](#) [Software](#) [Open](#)**eOSSR**

Enrique Garcia; Thomas Vuillaume

The ESCAPE OSSR library The eOSSR is the Python library to programmatically manage the ESCAPE OSSR. In particular, it includes: an API to access the Zenodo and the OSSR, retrieve records and publish content functions to

Zenodo as 

- FAIR centered
- long-term archive
- software citability (DOI)
- widely accepted and used
- don't reinvent the 
- integrates with other services
- community management

→ *escape2020* community

Software metadata

Software metadata are the implementation of FAIR principles

- Findable, Interoperable
- They should be part of the software and not defined or retained by an external service



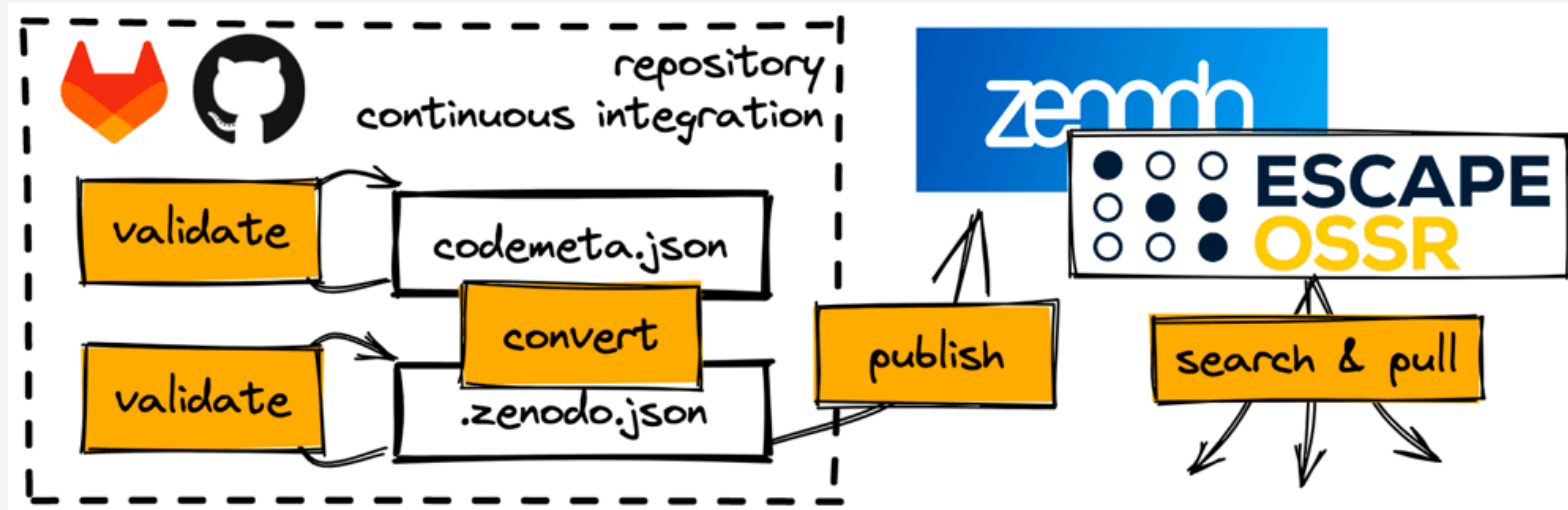
OSSR uses [CodeMeta](#)

- Universal metadata schema to describe software
- Not limited or linked to a specific service
- Increasing adoption
- Integration with other services

→ A `codemeta.json` file with a number of required keys is mandatory to submit software to the OSSR. The file comes with the source code, at the root of the repository.

● The eOSSR is the OSSR Python library

- Connects to Zenodo API to handle:
 - records: search, download, upload, publish, submit...
 - communities: list records, list and handle submissions
- Handles OSSR metadata:
 - Defines required one
 - Converts from CodeMeta to Zenodo schema
 - Validates `codemeta.json` file



Online tools: metadata generator, converter & validator

Validate and convert your metadata



This notebook will help you validate your metadata for an upload to the ESCAPE OSSR.

To do so, upload your codemeta metadata, either using an URL pointing to the `codemeta.json` file, uploading a `codemeta.json` file or copying the metadata in the text box below.

Note that you can generate your ESCAPE codemeta file using the online generator: <https://escape2020.pages.in2p3.fr/wp3/codemeta-generator/>

Load codemeta from a `json` file

Upload (0)

Load codemeta from a `Zenodo record ID`

Record ID:

Load

Load codemeta from an `URL`

URL:

Load

codemeta:

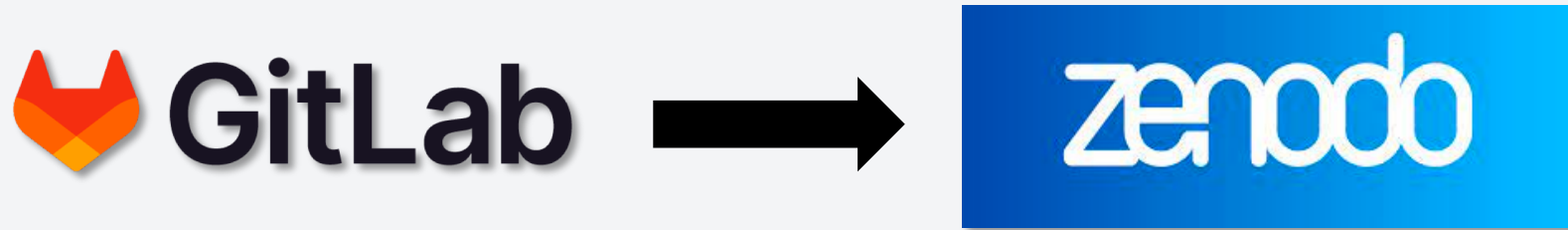
ESCAPE OSSR CodeMeta generator

This tool helps you create a CodeMeta.json file for your software. Note however that it is not exhaustive and other fields can be manually added in your file following the [CodeMeta schema](#). Most fields are optional. Mandatory fields will be highlighted when generating CodeMeta.

<p>The software itself</p> <p>Name <input type="text" value="My Software"/> <small>the software title</small></p> <p>Description <input type="text" value="My Software computes ephemerides and orbit propagation. It has been developed from early '80."/></p> <p>Documentation or readme <input type="text" value="https://online-documentation.org"/></p> <p>Creation date <input type="text" value="YYYY-MM-DD"/></p> <p>First release date <input type="text" value="YYYY-MM-DD"/></p> <p>License <input type="text"/> <small>from SPDX license list</small></p>		<p>Discoverability and citation</p> <p>Unique identifier <input type="text" value="10.151.xxxxx"/> <small>such as ISBNs, GTIN codes, UUIDs etc. http://schema.org/identifier</small></p> <p>Application category <input type="text" value="Astronomy"/></p> <p>Keywords <input type="text" value="Projects: CTA, EGO-Virgo, ELT, EST, FAIR, HL-LHC, KM3Net, LSST, LOFAR, SKA; Content: Astronomy, Astroparticle physics, Particle physics"/></p> <p>Keywords <input type="text"/></p> <p>Funding <input type="text" value="ESCAPE 824064"/> <small>grant funding software development</small></p> <p>Funder <input type="text" value="European Union's Horizon 2020 research and innovation programme"/> <small>organization funding software development</small></p> <p><small>Authors and contributors can be added below</small></p>	
<p>Development community / tools</p> <p>Code repository <input type="text" value="git+https://github.com/You/RepoName.git"/></p> <p>Continuous integration <input type="text" value="https://travis-ci.org/You/RepoName"/></p> <p>Issue tracker <input type="text" value="https://github.com/You/RepoName/issues"/></p> <p>Related links <input type="text"/></p>		<p>Run-time environment</p> <p>Programming Language <input type="text" value="C#, Java, Python 3"/></p> <p>Runtime Platform <input type="text" value=".NET, JVM"/></p> <p>Operating System <input type="text" value="Android 1.6, Linux, Windows, macOS"/></p> <p>Other software requirements <input type="text" value="Python 3.4"/> <small>https://github.com/psf/requests</small></p>	
		<p>Current version of the software</p> <p>Version number <input type="text" value="1.0.0"/></p> <p>Release date <input type="text" value="YYYY-MM-DD"/></p> <p>Download URL <input type="text" value="https://example.org/MySoftware.tar.gz"/></p> <p>Release notes <input type="text" value="Change log: this and that; Bugfixes: that and this."/></p>	


- Help software developers to provide valid and complete metadata
- Get that first working version of `codemeta.json`
- Test things out

Gitlab to Zenodo



- Zenodo has an efficient GitHub integration, but no gitlab integration
- Many ESFRIs use their own Gitlab instance
- ➔ We provide a simple gitlab-ci snippet
 - to publish your software to Zenodo / OSSR, e.g. when making a release in gitlab
 - using metadata provided in `codemeta.json`

- The OSSR is a curated software repository
 - implementation of the FAIR principles
 - good code practices
 - software quality
 - do not review scientific results → science paper
- Curation happens in a dedicated gitlab repository
 - completely open
 - automated checks
 - discussion between reviewers and providers
- Curation provides
 - Trust in the repository and provided content
 - Recognition for software providers



ossr / OSSR Curation / Merge requests / 148

[CURATE] HiPeRTA

Open Thomas Vuillaume requested to merge 7378249 into master 1 year ago

Overview 1 Commits 2 Pipelines 2 Changes 1 Add a to do

=== Record #7378250 ===
 Title: HiPeRTA
 DOI: 10.5281/zenodo.7378250
 URL: <https://zenodo.org/record/7378250>
 HiPeRTA is a C++ library providing High Performance computing algorithms which provide full C++ programs from HiPeCTA C++ sources for the Cherenkov Telescope Array (CTA) low-level data analysis real time reconstruction. It takes advantage of the latest SIMD (Single input multiple data) operations included in modern processors, for native vectorized optimization of analytical data processing. It is developed to be updated by respect to the ctape framework.

Check the software checklist for the entry

- Contains valid codemeta.json (see validator output)
- Documentation is provided in the Zenodo entry (at least through codemeta)
- a stable versioned release of the project
- It is under an open-source license (see SPDX [<https://spdx.org/licenses/>])
- Follows a reasonable set of software development / software engineering practices (rough by-eye quality estimate)

Complete onboarding issue

Related onboarding issue: XXX (to be entered by onboarding manager)

- Make sure all boxes of the checklist up to "Uploaded to Zenodo" are ticked
- Tick "software checklist completed" when done with the above
- When cleared for merging, tick "Added to Zenodo community/published" and change issue status to "closed"

There are 8 warnings:

- memoryRequirements not provided in the codemeta schema but is recommended
- processorRequirements not provided in the codemeta schema but is recommended

Assignee Edit
Thomas Vuillaume

Reviewer Edit
Kay Graf

Labels Edit
Ready for curation

Milestone Edit
None

Time tracking +
No estimate or time spent

3 Participants
Kay Graf, Thomas Vuillaume, [Avatar]

Integration with other services

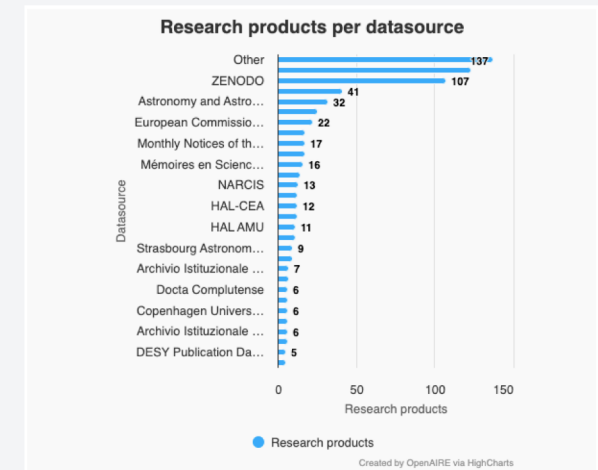
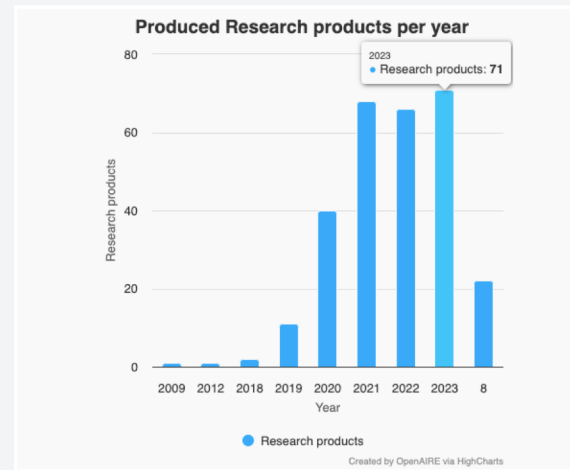


- connects to other services
- analyze data
- search and pull software from the OSSR

- EOSC integration
- Provides integrated statistics
- Connects with other data sources

The screenshot displays a grid of service cards for various astronomical observatories and data centers:

- WSRF-Apertif**: Data from the Apertif survey include imaging and time-domain data.
- ASTRON VO**: The Virtual Observatory defines a set of standards that can be used to download astronomical data.
- Zooniverse**: The Zooniverse is the world's largest and most popular platform for people-powered research.
- Virtual Observatory (VO)**: The Virtual Observatory offers a set of standards that can be used to download astronomical data.
- CTAO**: Cherenkov Telescope Array Observatory (CTAO) will be the world's first ground-based gamma-ray observatory.
- RUCIO**: Built on more than a decade of experience, RUCIO serves the data needs of modern scientific experiments.
- Science Data Centre (SDC)**: Science Data Centre (SDC) is a service of Leibniz-Institute for Astrophysics IPANGU.
- ZENODO**: Zenodo built and developed by researchers, to ensure that everyone can join in Open Science.





A single entry point

<http://purl.org/escape/ossr>

- Find all the information
- Search the OSSR
- Onboarding instructions



OUR VISION ▾

CONTRIBUTE ▾

INFRASTRUCTURE ▾

TOOLS ▾

ABOUT ▾



Search software and services in the ESCAPE repository

Welcome to the ESCAPE OSSR!

[Browse the OSSR content.](#)

What is it?

The ESCAPE Open-source Scientific Software and Service Repository (OSSR) is a sustainable open-access repository to share scientific software, services and datasets to the astro-particle-physics-related communities and enable open science. It is built as a curated [Zenodo community](#) integrated with several tools to enable a complete software life-cycle. The ESCAPE Zenodo community welcomes entries that support the software and service projects in the OSSR such as user-support documentation, tutorials, presentations and training activities. It also encourages the archival of documents and material that disseminate and support the goals of ESCAPE.

How to contribute to the ESCAPE OSSR?

You can onboard your project right now - [see here](#) how.

Learn more about our projects in this website or [contact us!](#)



The OSSR galaxy

The collage features several key components of the OSSR galaxy:

- ESCAPE OSSR Home Page:** A screenshot of the main website with the text "Welcome to the ESCAPE OSSR!" and "What is it?".
- ESCAPE OSSR CodeMeta generator:** A screenshot of a web form used for generating CodeMeta metadata.
- CodeMeta Icon:** A logo consisting of a code symbol (</>) and the text "CodeMeta".
- ESCAPE ESAP and OpenAIRE EXPLORE:** Logos for "ESCAPE ESAP" and "OpenAIRE EXPLORE".
- zenodo Project Page:** A screenshot of the Zenodo project page for ESCAPE OSSR, showing the URL "https://bzurl.org/escape/ossr" and a "New upload" button.
- GitHub and GitLab Logos:** The logos for GitHub and GitLab, indicating their integration with the OSSR workflow.
- Curation Screenshot:** A screenshot of a curation interface for a software entry, with the word "Curation" written above it.
- Workflow Diagram:** A diagram illustrating the workflow: "validate" (twice) -> "convert" (from "codemeta.json" to "zenodo.json") -> "publish" -> "search & pull" (in a "repository" context). It also includes "continuous integration" and "zenodo" logos.



The OSSR galaxy



Built by,
for
and with
all the
partnering
communities



Curation

```

graph TD
    subgraph "continuous integration"
        V1[validate] --> C[codemeta.json]
        V2[validate] --> Z[zenodo.json]
    end
    C --> P[publish]
    Z --> P
    P --> R[repository]
    R --> S[search & pull]
  
```




The OSSR galaxy



OSSR workshops,
collaboration meetings
and Software Schools
and community events

CodeMeta

zenodo

GitHub **GitLab**

eOSSR

continuous integration repository

validate → codemeta.json → publish → search & pull

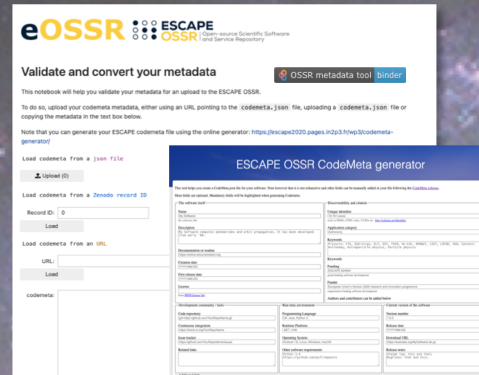
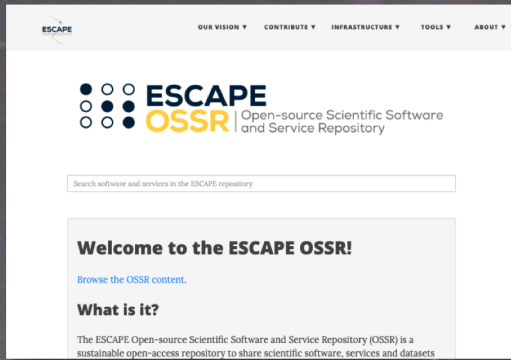
validate → convert → zenodo.json → publish → search & pull

Curation

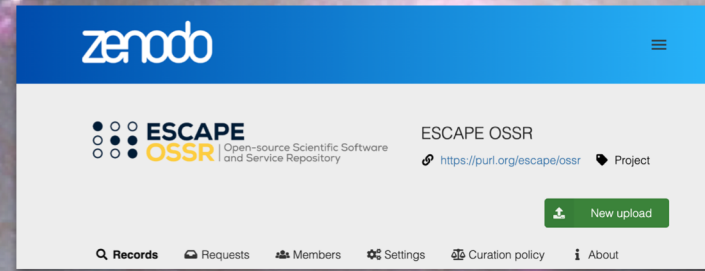
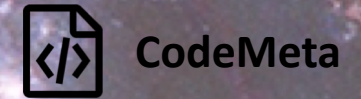


Final question...

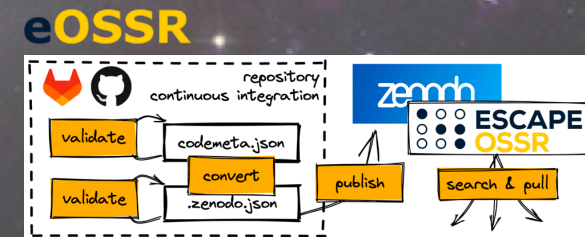
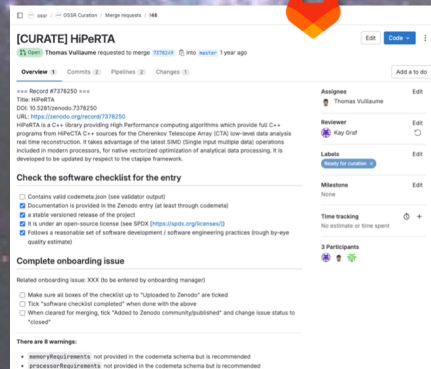
How to add **your software** to the OSSR?

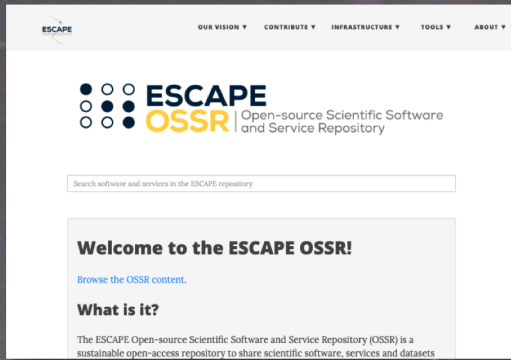


OSSR entry point
See our requirements
Contact us if needed

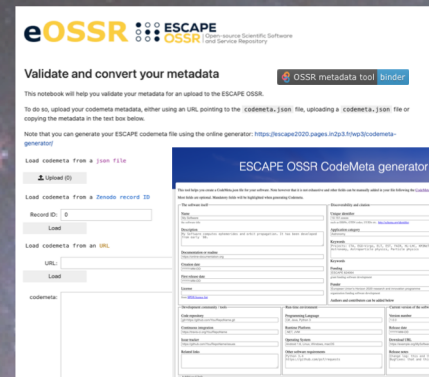


Curation





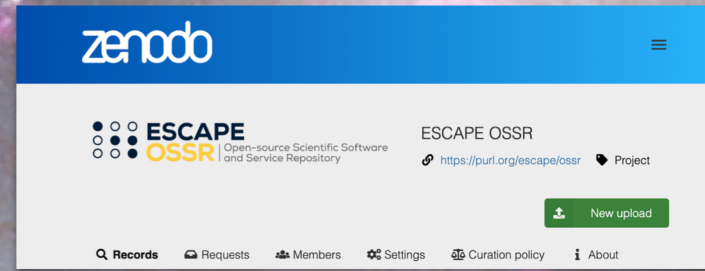
OSSR entry point
See our requirements
Contact us if needed



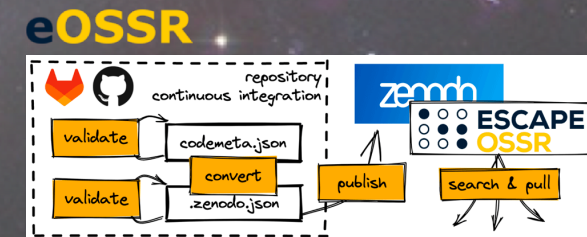
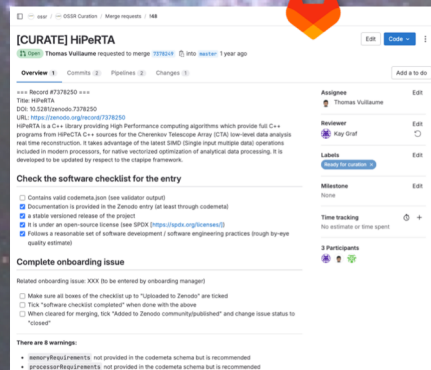
Produce codemeta.json.
Check its validity.

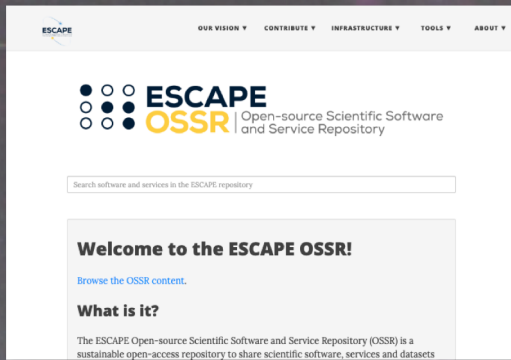


CodeMeta

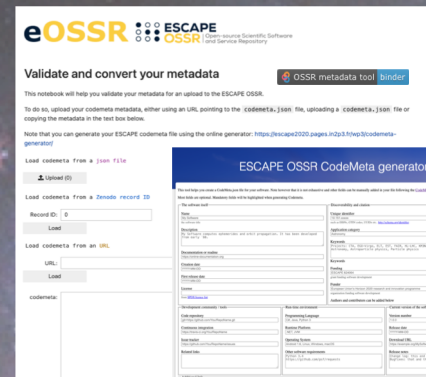


Curation





OSSR entry point
See our requirements
Contact us if needed

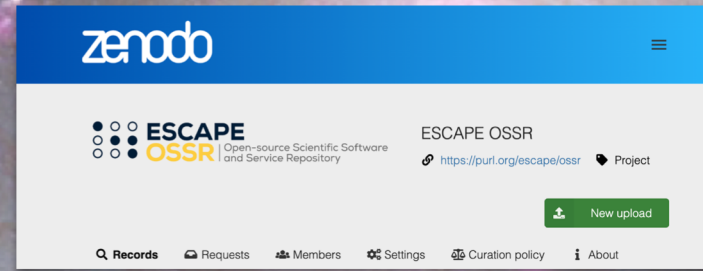


Produce codemeta.json.
Check its validity.

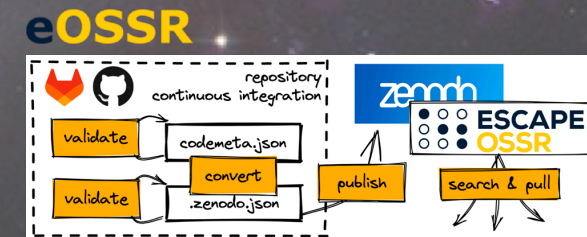
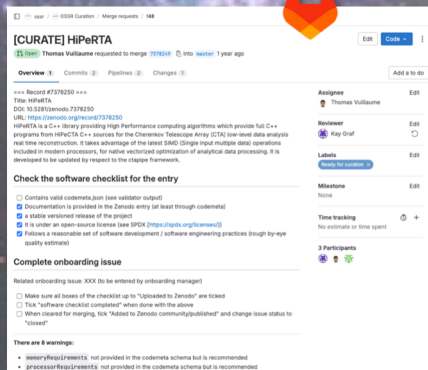


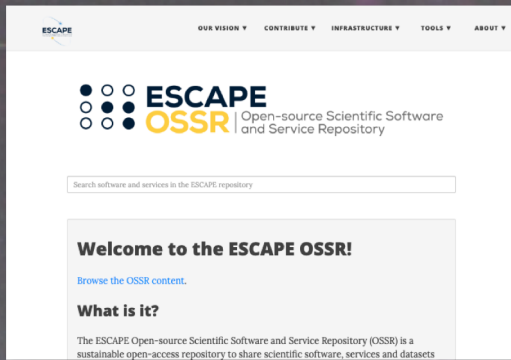
CodeMeta

Add codemeta.json
to your repo.

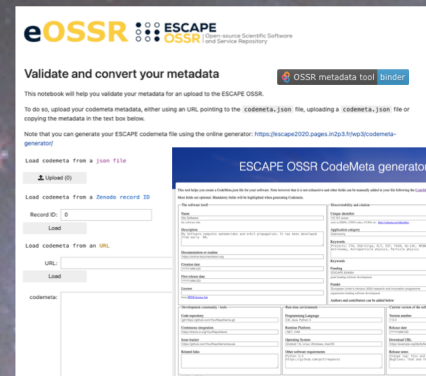


Curation





OSSR entry point
See our requirements
Contact us if needed

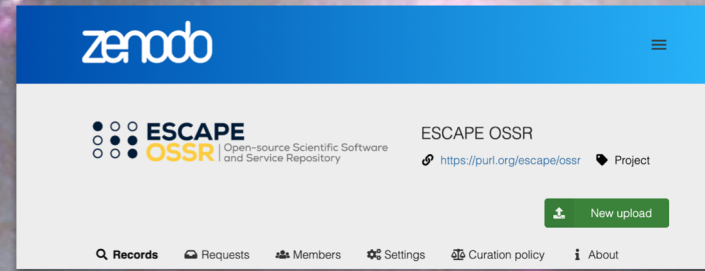


Produce codemeta.json.
Check its validity.

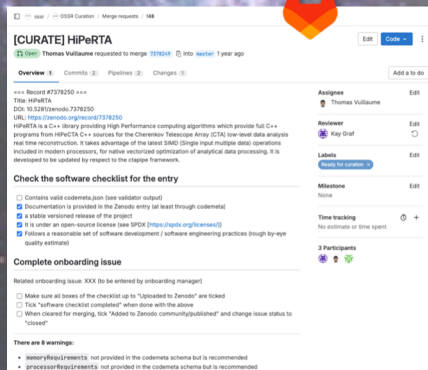


CodeMeta

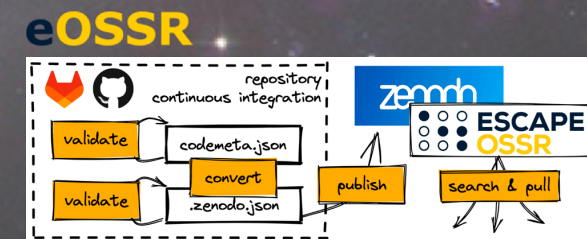
Add codemeta.json to your repo.

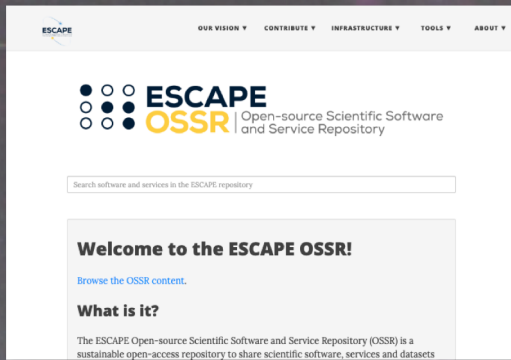


Curation

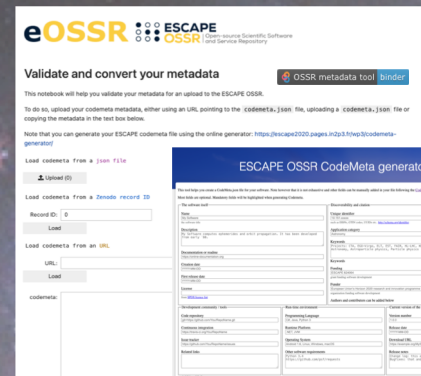


Keep codemeta.json updated
CI/CD validates codemeta.json in time





OSSR entry point
See our requirements
Contact us if needed

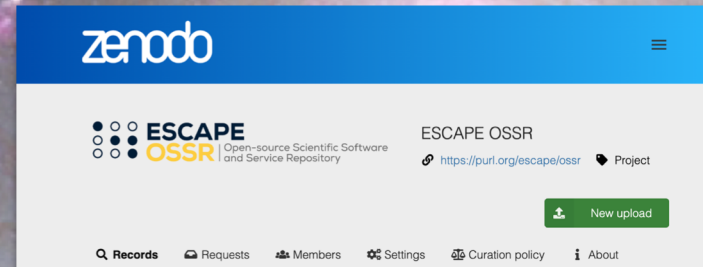


Produce codemeta.json.
Check its validity.



CodeMeta

Add codemeta.json to your repo.

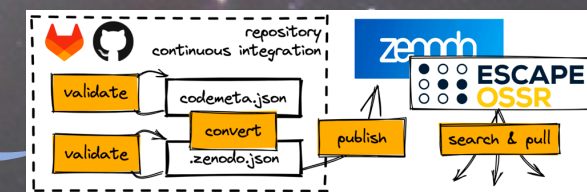


GitHub/Gitlab to Zenodo.
Request (once) to integrate
escape2020 community
Each new release is automatically
added as new version

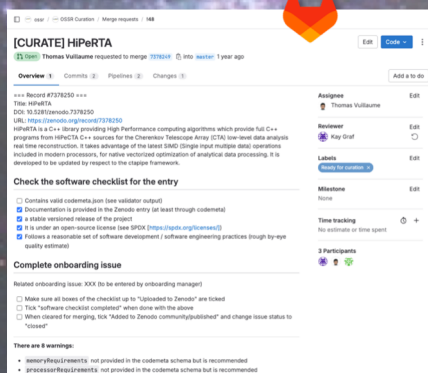


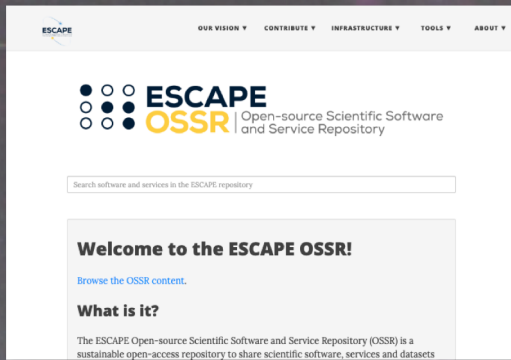
Keep codemeta.json updated
CI/CD validates codemeta.json in time

eOSSR

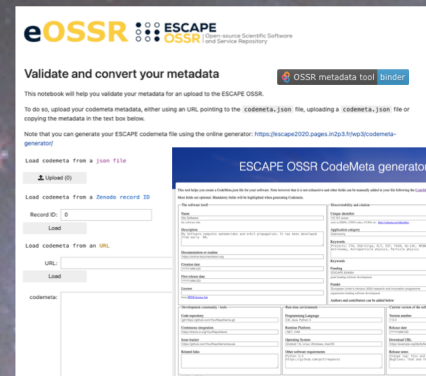


Curation





OSSR entry point
See our requirements
Contact us if needed

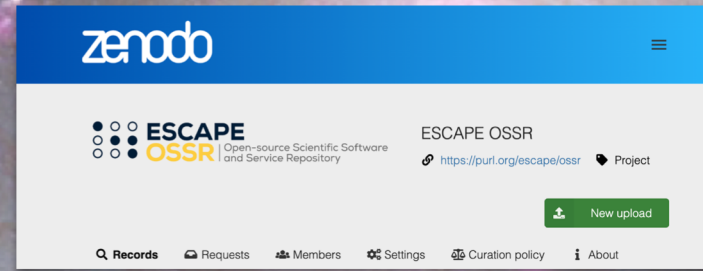


Produce codemeta.json.
Check its validity.



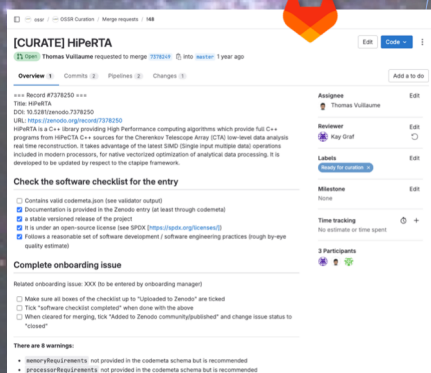
CodeMeta

Add codemeta.json to your repo.



Record gets curated

Curation

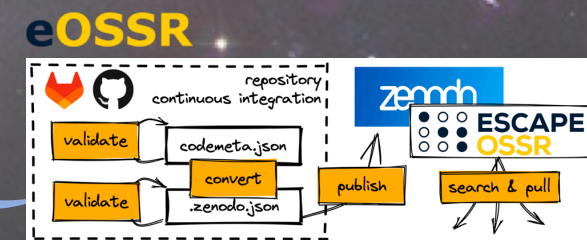


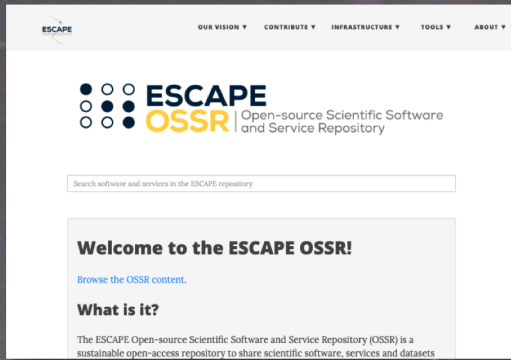
GitHub/Gitlab to Zenodo.

Request (once) to integrate
escape2020 community
Each new release is automatically
added as new version

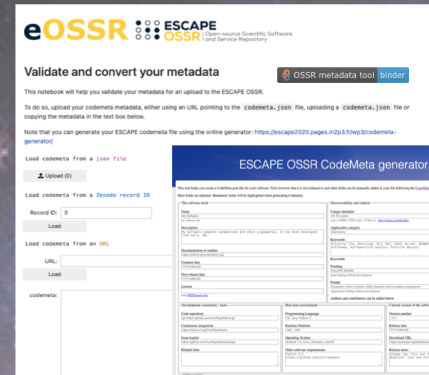


Keep codemeta.json updated
CI/CD validates codemeta.json in time





OSSR entry point
See our requirements
Contact us if needed



Produce codemeta.json.
Check its validity.

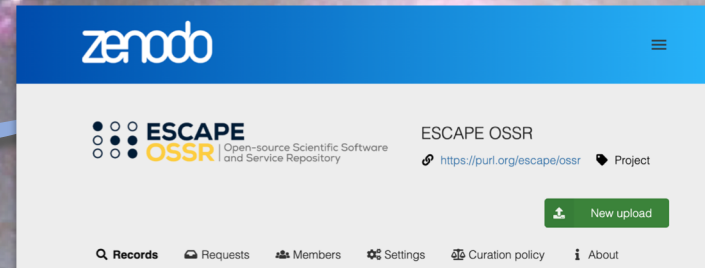


CodeMeta

Add codemeta.json to your repo.



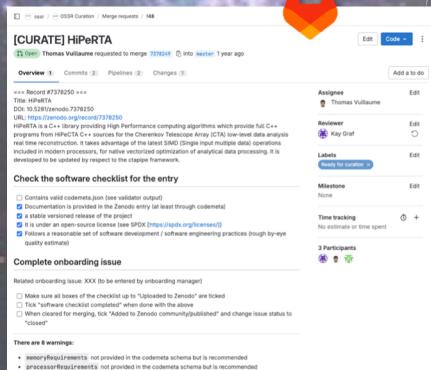
Acceptation in OSSR
Findable and Accessible
by other services



Record gets curated

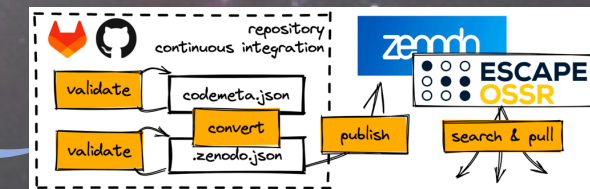
GitHub/Gitlab to Zenodo.
Request (once) to integrate
escape2020 community
Each new release is automatically
added as new version

Curation



Keep codemeta.json updated
CI/CD validates codemeta.json in time

eOSSR



Conclusion and future

- The [OSSR](https://open-research-europe.ec.europa.eu/articles/3-46) is accepting quality software and analysis code from astro & particle physics communities

- Set of tools to help you in the software lifecycle
- Integration with services
- Curation to build trust and recognition
- Find our open letter at <https://open-research-europe.ec.europa.eu/articles/3-46>

<https://purl.org/escape/ossr>

- Community events to come in 2024



- Partnership with <https://research-software-directory.org/> for enhanced capabilities and better findability

- [EVERSE EU project](#) starting in March 2024

- Establish an European Virtual Institute for Research Software
- Keep improving software quality in research
- Include more software quality and FAIRness metrics in the OSSR



- [OSCARS EU project](#) to consolidate all clusters achievements in time





ESCAPE

European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

 projectescape.eu

 [@ESCAPE_EU](https://twitter.com/ESCAPE_EU)

 company/projectescape