Overview of progress in the NIVA Project

Ioannis Andreou, Kostas Kountouris

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 842009

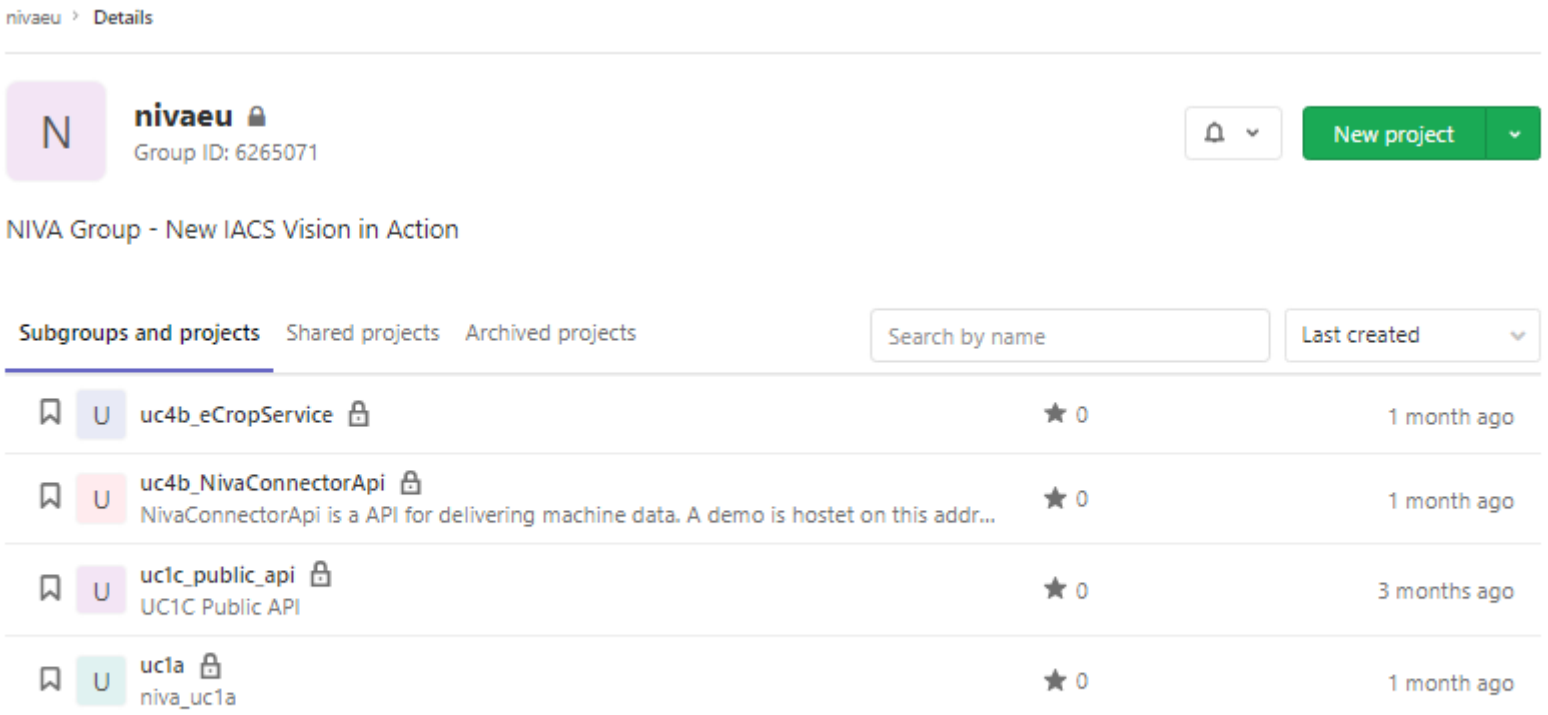
Overview

- Resources
- Current Status
- Instructions
- Discussion

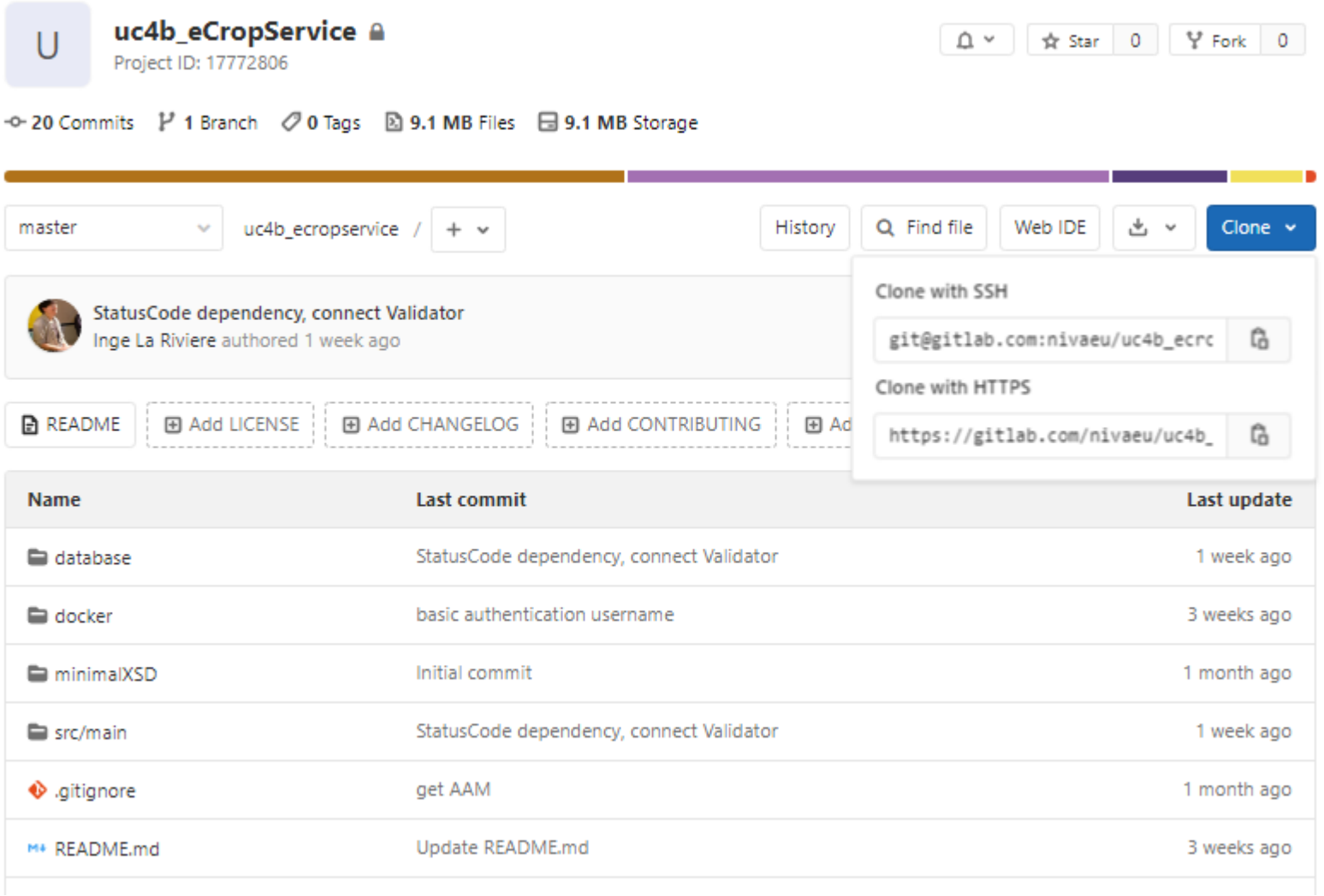
Resources

- All related resources will be in: WP4\Task 4.1
- Contains:
  - Users Table
  - Gitlab Presentation
  - Instructions Document

Current Status



Current Status



Current Status

README.md

this repository holds the in progress demo-1 PA webservice for uc4b

- JAVA servlet war with secured endpoints:
  - nivaecropmessage
  - rest/asappliedmap

nivaecropmessage:

- check input empty
- checks valid JSON
- checks ECROPReportMessage (in object) by parsing into JAVA classes generated from sample message
- stub validator call
- checks validator ECROPReturnMessage (in object) by parsing into JAVA classes
- stub backend call
- returns ECROPReturnMessage (in object)
- saves in single row in database table CropReport
  - ECROPReportMessage as JSONB
  - ECROPReturnMessage as JSONB
  - ID as serial
  - issuer and
  - TZ-datetime
- run in wildfly/JBoss docker (Dockerfile included)
- security constraints and roles for
  - read-only (asappliedmap) and
  - read-write (nivaecropmessage)

rest/asappliedmap{/id}:

Current Status

- No technical issues so far, some network issues were resolved
- Instructions were followed
- Some additional instructions might be needed
- Users from other UCs have also requested and got access, tested the system
- Two Use Cases have created projects and pushed files
- Other Issues?

Instructions

When to use Git

- Always, when you work on the project, not only to ‘upload’.
  - Will be more efficient and safer
  - Versioning, commit comments and other data will be available
- If another platform is used for Version Control, consider changing.
  - If that is not possible, a lot more work is required
  - History information might not be complete
- A Version Control System is assumed to be in place.

Gaining Access

1. Fill in information in Users Table
2. Request access via e-mail => Developer access
3. Create Project
4. Request Maintainer access for at least one person/project, if needed
5. Profit (Work)
6. Members of the project group should have guest access to all projects

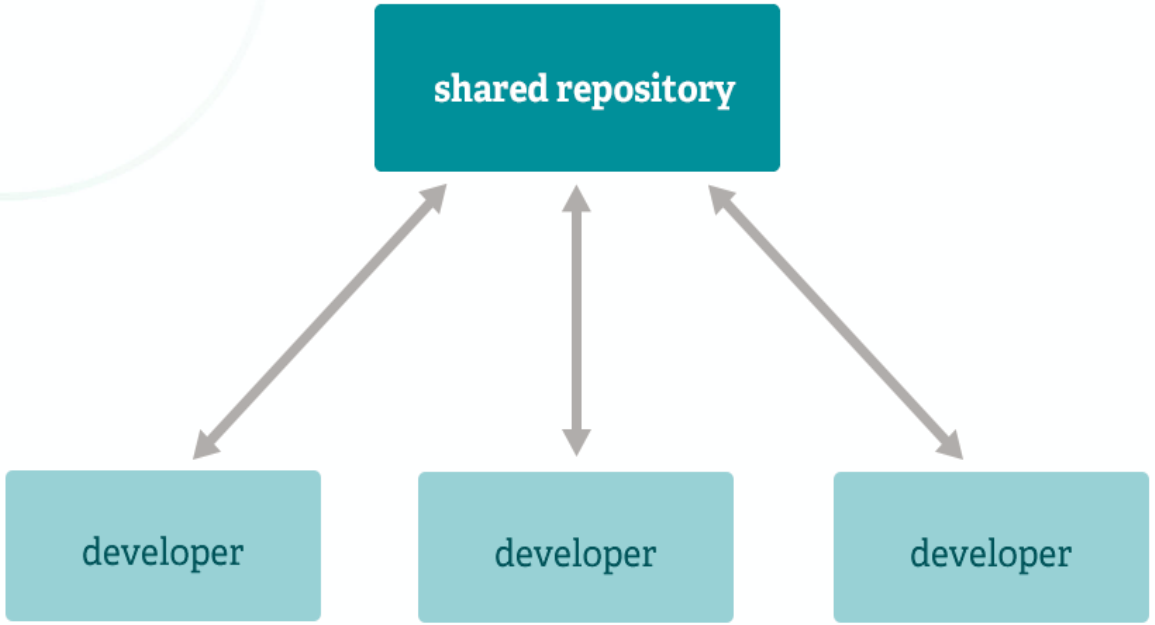
Proposed Methodology

- For each working group a maintainer must be chosen and be communicated as such to the administration.
- The maintainer will create the projects, similarly to the ones existing in the Mock group.
- Maintainer of each project may control every aspect of the project. “Internal” visibility is suggested due to the nature of NIVA and ease of management.
- Every user of the NIVA group will have access to view all the projects

Repository Management

- Don’t use capital letters in repository names.
- Start the name of the repository with the package that it belongs to e.g. uc1a_docs may be an extra documentation repository for package uc1a. Use underscore (_) to separate words.
- Coding conventions: We don’t provide coding conventions or platform suggestions as these are not requested or required by the agreement. Partners probably need this level of freedom anyway.
- **Do not rely only on the Gitlab service as a backup** for your work. Git is a powerful tool that gives more options. Please maintain your own common repository in parallel, as a local Gitlab installation or simply as git folder(s) in your file server.
- Feel free to add other resources that are needed by your project to work, such as images, libraries etc. It is assumed that the submitted components can be installed and work as they are, where applicable.

Expected model for each of the Niva projects (repositories)



- Each group is encouraged to also maintain their own local (group) repository. Draw a line from each developer to a secondary local repository

Additional Recommendations

- More information about projects might be needed, such as:
  - Short description, if it is not obvious by the project name
  - Version number, if needed
  - Dependencies: additional libraries and resources,
    - e.g. uses X library version 1.2, which can be either in a separate project
    - or just mentioned/configured in the project
  - Installation instructions and development/execution environment
    - e.g. requires Apache >=version X
    - or requires Sen4Cap version X1, followed by more installation instructions
- README.md file
  - In the project folder, unless there is already an appropriate structure in your project
  - Use that file to point to relevant information if needed, such as “Documentation is maintained separately in project ‘uc1a_docs’”

Discussion

Thank you

ioannis.andreou@opekepe.gr
kostas.kountouris@opekepe.gr

This project has received funding from the european union's horizon 2020 research and innovation programme under grant agreement no. 842009