UN/CEFACT JSON-LD Code Publications

Streamlining UN/CEFACT Code Lists Maintenance with GitHub

Understanding GitHub

- **GitHub** is a web-based platform designed for collaborative software development. It offers features like **version** control, **issue** tracking, and **collaboration** tools, making it ideal for managing code list **maintenance**.

Transparent DMRs tracking using GitHub issues

- With GitHub, data maintenance requests can be raised as **GitHub issues** and tracked easily and transparently.
- A request **progress**, updates and discussions are easy to monitor by all participants of the process.
- Additionally it can be used to collect the **feedback** from the community and **improve** the data quality of the code lists.

Teams and permissions management

- GitHub provides a simple and secure way to engage in the data maintenance process by using teams and specifying fine grained permissions:

- new issues – logged in user

- pull requests – contributors or maintainers team

- pull requests approve – maintainers team

- release management – release manager

Permissions are easy to grant or revoke at any moment.

Efficient Issue Management

- GitHub's **issue tracking** system ensures efficient management of DMRs.
- The issues can be **categorized**, **prioritized** and **assigned** to a person/team, responsible for a particular code list maintenance.
- **Specific templates** are defined for every code list that makes it easy to fill in all required data for DMR.
- This process helps to ensure **timely resolution** of the requests **without overload** for the maintainer team.

Version Control

- **Git**, the underlying version control system used by GitHub, enables **maintainers**, **contributors** and **developers** to track changes.
- It allows to **preview** the changes before applying them and **correct** if needed right away.
- DMR pull request must be **approved** before it can be merged in. The enforced requirement for pull requests to be up to date with the main branch **prevents the data loss or conflicts** when several DMRs are in progress.

Documentation

GitHub provides tools for documenting data standards, schemas, and maintenance processes. This includes **wikis**, **README** files, and markdown **documentation** within repositories.

- Clear and up-to-date documentation helps to understand the **data maintenance process** and to adhere to **established guidelines**.

- **Continuous integration/delivery (CI/CD) pipelines**

- Continuous integration/delivery **(CI/CD)** pipelines, that allows to **automate basic validation**, **transformation** into different formats and outputs **publication**.
- GitHub workflows are easy to implement and support, they don’t require dedicated infrastructure and maintenance.
- Validation, transformation and publication code is easy to invoke and **execution logs** are available real-time.
- DMR PRs that are failing basic validation cannot be merged in.

Community Engagement

- GitHub has a large and active community of developers and data professionals. Leveraging GitHub for UN/CEFACT data maintenance can help **engage** this community, **encourage contributions**, and facilitate knowledge sharing. Open discussions and collaboration on GitHub can lead to improved data standards and **increased adoption**.
- GitHub is already widely used by world standard issuing authorities like the World Wide Web Consortium (**W3C**) and others.

Release management

- A new release can be **drafted** before publishing, code lists output files can be attached as **artifacts**, what makes them accessible to contributors and other users. A **release note** with all the changes can be easily generated and included into release for review. Once the release is approved it is **published** and release **announcement** is made.

Publication process

- Publication process includes preview publication of the changes to be included into release on **test endpoint** and release publication on **production endpoint** when the release is published.
- This allows to **preview** and **verify** the applied DMRs before including them into the release.
- In case if an DMR was included into release by mistake the specific changes always can be **rolled back** before the release is published.

DMR in simple steps

- User creates an issue using template.
- User or maintainer raises a pull request based on the issue to update code list csv file content. Basic QA checks for duplicates or data removal.
- Maintainer reviews the request and approves it, suggests changes or rejects it.
- Release manager includes the request into release.

Conclusion

- Overall, GitHub provides a robust platform for managing UN/CEFACT data maintenance requests for code lists, offering transparency, collaboration, version control, and documentation capabilities that are essential for effective data management processes.

Links

- https://github.com/uncefact/vocab-codes/wiki - wiki for the proposed approach to manage code list recommendations on GitHub platform.
- https://test.uncefact.org/vocabulary/vocab-codes - published code lists on the test endpoint.
- https://vocabulary.uncefact.org/unlocode-about - UN/LOCODEs JSON-LD vocabulary.