Project Symmetry: API Development

🥐Team Scrum-ptious🥐

The Project & Our Contribution

Overall:

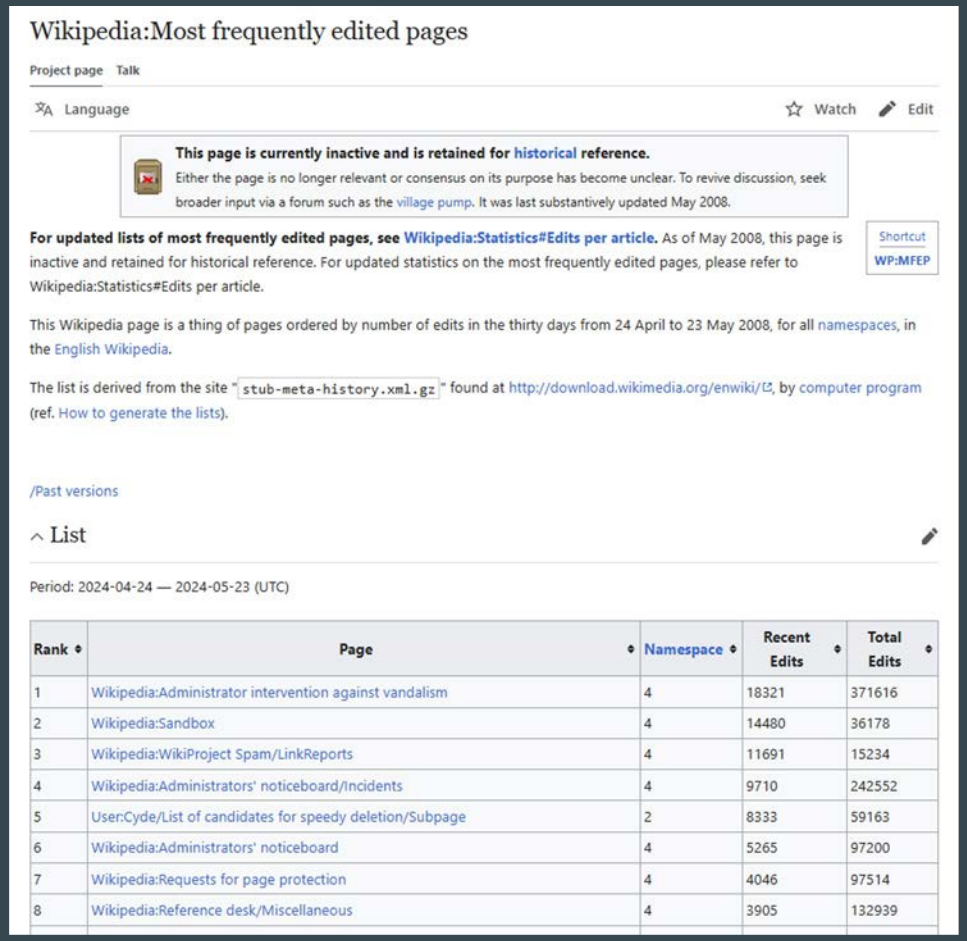
- Ensure content parity across different language versions of Wikipedia articles and future wiki-based platforms.
- Aim to deliver accurate, consistent, and idiomatically correct information.
- Enhance Wikipedia's multilingual content using machine learning for:
  - Accurate translation.
  - Semantic analysis.
  - Edit recommendations for translational consistency.
  - Misinformation targeting.

What We're Doing:

- Developing API/middleware to facilitate efficient data exchange between front-end and back-end developers.

Goals

- Learn how to develop middleware/APIs.
- Efficient collaboration with multiple teams.
- Organize responsibilities utilizing Agile methodology.
- Maintain consistent communication.
- Learn how to interpret and record documentation.



Wikipedia:Most frequently edited pages

Project page Talk

Language Watch Edit

This page is currently inactive and is retained for historical reference.
Either the page is no longer relevant or consensus on its purpose has become unclear. To revive discussion, seek broader input via a forum such as the village pump. It was last substantively updated May 2008.

For updated lists of most frequently edited pages, see Wikipedia:Statistics#Edits per article. As of May 2008, this page is inactive and retained for historical reference. For updated statistics on the most frequently edited pages, please refer to Wikipedia:Statistics#Edits per article.

Shortcut
WP:MFEP

This Wikipedia page is a thing of pages ordered by number of edits in the thirty days from 24 April to 23 May 2008, for all namespaces, in the English Wikipedia.

The list is derived from the site "`stub-meta-history.xml.gz`" found at http://download.wikimedia.org/enwiki/, by computer program (ref. How to generate the lists).

/Past versions

List

Period: 2024-04-24 — 2024-05-23 (UTC)

Rank	Page	Namespace	Recent Edits	Total Edits
1	Wikipedia:Administrator intervention against vandalism	4	18321	371616
2	Wikipedia:Sandbox	4	14480	36178
3	Wikipedia:WikiProject Spam/LinkReports	4	11691	15234
4	Wikipedia:Administrators' noticeboard/Incidents	4	9710	242552
5	User:Cyde/List of candidates for speedy deletion/Subpage	2	8333	59163
6	Wikipedia:Administrators' noticeboard	4	5265	97200
7	Wikipedia:Requests for page protection	4	4046	97514
8	Wikipedia:Reference desk/Miscellaneous	4	3905	132939

Project Users and Their Needs

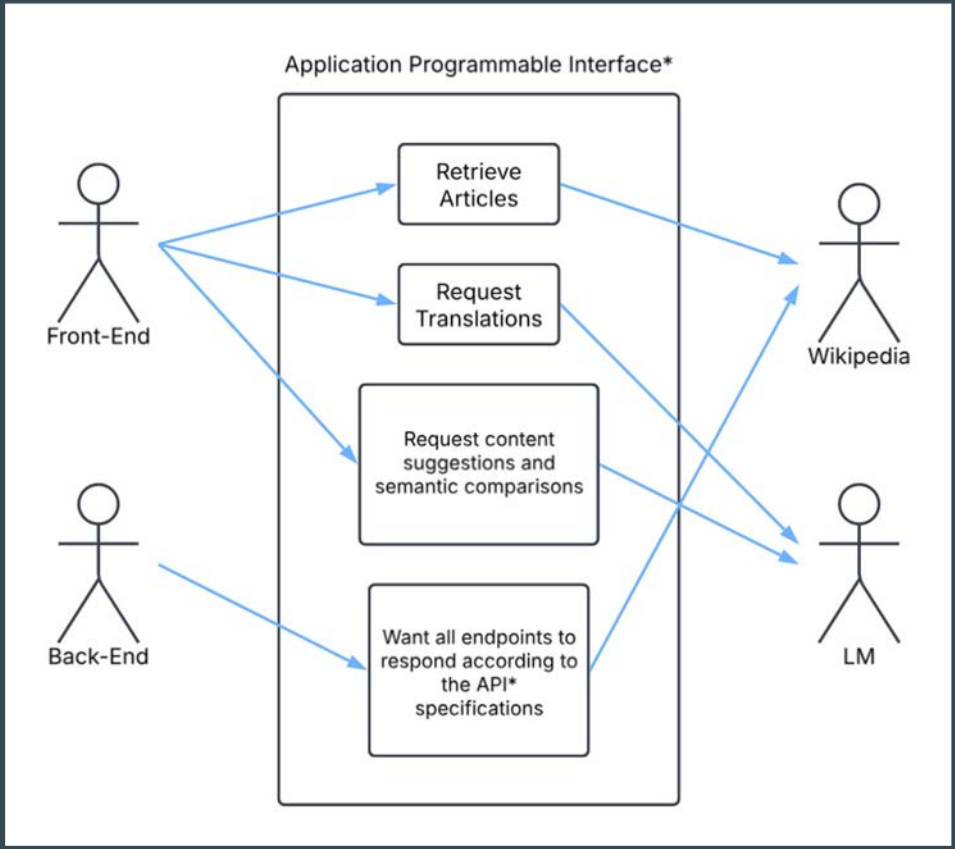
Front-end developers:

- Retrieve Wikipedia articles in multiple languages via the API.
- Request translations and display them alongside original content.
- Request content suggestions and semantic comparisons via the API from the LM.

Back-end developers:

- Respond with requested translations.
- Provide semantic comparisons in a format that can be interpreted by the front end.

Use Case Diagram



Functional Requirements

What the system must do to meet user needs

~ Retrieve articles from Wikipedia's database

~ Provide medium of communication between endpoints within the system, as well as external systems (Wikipedia)

~ Request translations and display them alongside original content

~ Request content suggestions and semantic comparisons via the API from the LM

Nonfunctional Requirements

How the system should perform under various conditions

Performance: Request optimization

Error Handling: The API should return standardized error messages

Logging & Monitoring: Incoming and outgoing requests/responses should be logged for debugging and analytics.

Flexibility: System should provide for users with various system capabilities

Product Backlog Highlights

~ Retrieve Articles from Multiple Wikipedia Languages (Front-end) ~

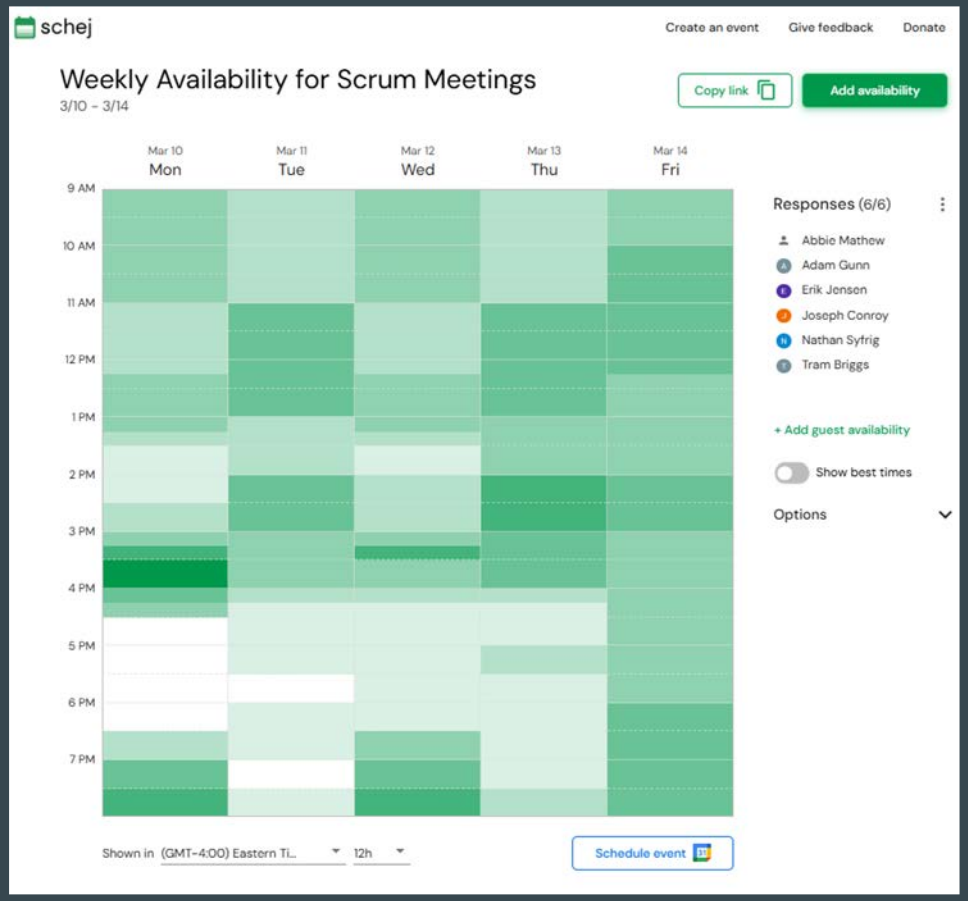
~ Ensure APIs Align with Front-end Requirements (Back-end) ~

~ Consistent Response Structures Across All Endpoints (Back-end) ~

~ Error-Handling Mechanisms for Meaningful Error Responses (Back-end) ~

Sprint Planning Strategies

- Assess team strengths, and divide tasks accordingly.
- Ensure each sprint is about a fourth of the product backlog.
- Choose tasks in logical succession.
- Consider potential roadblocks.
- Seek guidance from Scrum Master and Tech Lead.



Lessons Learned

Interacting with Client:

- Maintain consistent communication without making assumptions.
- Scheduling tools are very effective to plan meetings.

Teamwork: Reflect on how the team coordinated sprint planning, distributed tasks, and addressed challenges.

- Weekly team meetings.
- Always establishing agreement when making decisions.

Thank You!

Questions?