HTW BERLIN
UNIVERSITY OF APPLIED SCIENCES

Generating Article Placeholders from Wikidata for Wikipedia: Increasing Access to Free and Open Knowledge

Author:
Lucie-Aimée KAFFEE

First Examiner:
Prof. Dr. Debora WEBER-WULFF
Second Examiner:
Dipl.-Inf. Lydia PINTSCHER

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

International Media and Computing
Faculty IV

March 4, 2016

To Amaru and Laura
And my parents

HTW BERLIN – UNIVERSITY OF APPLIED SCIENCES

Abstract

Faculty IV
International Media and Computing

Bachelor of Science

Generating Article Placeholders from Wikidata for Wikipedia: Increasing Access to Free and Open Knowledge

by Lucie-Aimée Kaffee

The major objective of this thesis is to increase the access to open and free knowledge in Wikipedia by developing a MediaWiki extension called *ArticlePlaceholder*.

ArticlePlaceholders are content pages in Wikipedia auto-generated from information provided by Wikidata.
The criterias for the extension were developed with a requirement analysis and subsequently implemented.
This thesis gives an introduction on the fundamentals of the project and includes the personas, scenarios, user-stories, non-functional and functional requirements for the requirement analysis. The analysis was done in order to implement the features needed to achieve the goal of providing more information for under-resourced languages.
The implementation of these requirements is the main part of the following thesis.

Acknowledgements

I would like to thank my supervisor, Prof. Dr. Weber-Wulff, for her guidance throughout this thesis and her advice on academic practice.

I am very grateful for the opportunity to have worked with my mentor Lydia Pintscher. I thank her for her guidance, encouragement and support.

On a technical level, Marius Hoch supported me whenever necessary. Late night core patches to help my project continue are just one of many reasons I am very thankful to have the opportunity to work with him. He introduced me to the wonders and frustrations of working on a big and amazing open source project. Without him, this project would barely have been at the stage it is now.

The Wikidata team has helped and supported me patiently over the last year and while writing the thesis. Code review, patches, advice—I am very thankful to work with them: Daniel Kinzler, Katie Filbert, Marius, Adam Shorland, Jan Zerebecki, Adrian Heine, Thiemo Mättig, and Jonas Kress.

I am very thankful to have spent my university years with Charlie Kritschmar, with whom I experienced the joys of studying as well as the stressful times. She became a great friend and a very appreciated help whenever it comes to designing.

Without the proof reading I would have been lost—therefore I want to express my sincere gratitude to, among others, Timo Krüger, Gerrit Schäfer, and Andrew Thomas Blake.

An open source project would not be the same without a community. I had the joy of working with and getting to know the communities of Wikidata and Wikipedia. I am very thankful for their input and interest in this project.

Last but not least I want to thank my family and friends for their everlasting love and support.

Disclaimer

This thesis was written in cooperation with **Wikimedia Deutschland**.
The views expressed in this thesis are those of the student and do not necessarily express the views of **Wikimedia Deutschland** or the **Hochschule für Technik und Wirtschaft Berlin**.
The source code written as part of the thesis is published under **GNU General Public License 2.0 or later** and is in active development. Even though it was primarily developed by the student, the Wikidata team at Wikimedia Deutschland contributed to it with input and code review.

Contents

List of Figures

Chapter 1

Introduction

One of the greatest barriers for accessing knowledge on the Internet is language. There is a tendency to provide information in one language—or at most a few—which makes it hard for speakers of all other languages to access that same information. This is also an issue with Wikipedia, a project used all across the globe by all kinds of people. There are many topics that are only covered in a few languages on Wikipedia. People who do not speak these languages do not have access to all the information available, and potentially vital, to them. This important issue needs to be addressed.

The ArticlePlaceholder extension was developed to reduce this informational gap. The objective is to give more people more access to knowledge by making use of Wikipedia's reach and Wikidata's multilingual data.
ArticlePlaceholder uses data to generate content pages on Wikipedia in the community's language and offers the possibility to create actual articles and to translate them. The extension aims at supporting readers as well as editors.

In the course of this thesis, an introduction to the fundamentals needed for the project will be given, as will an overview of the previous work done with regards to Wikidata and Wikipedia.
Since a lot of the development has been user-centric, personas were created to develop scenarios and user stories. From those scenarios non-functional and functional requirements were derived. The implementation realizes many of the requirements and includes the technical aspects of the extension.

Chapter 2

Fundamentals

2.1 Wikipedia

Wikipedia is one of the largest websites worldwide with articles in over 250 languages. As of January 2016, the English Wikipedia alone amounts to 5,000,000 articles. (*Wikipedia:Five million articles — Wikipedia, The Free Encyclopedia*, 2016)

The concept of Wikipedia is fairly simple: an open encyclopaedia anyone can edit. It was launched on January 15, 2001. (*Wikipedia — Wikipedia, The Free Encyclopedia*, 2016)

Each of the over 250 languages have their own Wikipedia, which is linked with *interlanguage links* to Wikipedias in other languages. "Interlanguage links are links from a page in one Wikipedia language to an equivalent page in another language" (*Help:Interlanguage links – Wikipedia, The Free Encyclopedia*, 2016).

Wikipedia can be edited with or without an account, either recording the username or the editor's IP address in the edit history of an article.

Editors can edit Wikipedia via the *Visual Editor*, which provides them with a WYSIWYG interface. WYSIWYG is according to Oxford Dictionaries (n.d.) "[d]enoting the representation of text on-screen in a form exactly corresponding to its appearance on a printout." The traditional way of editing on Wikipedia is writing content in a markup language developed for Wikipedia and its software, MediaWiki. This markup language is called *wikitext*.

Wikipedia is organized by volunteers. To organize and maintain the various aspects of Wikipedia, the communities of volunteers can nominate *administrators*.

> As *Wikipedia:Administrators — Wikipedia, The Free Encyclopedia* (2016) states, administrators are "Wikipedia editors who have been granted the technical ability to perform certain special actions on the English Wikipedia, including the ability to block and unblock user accounts and IP addresses from editing, edit fully protected pages, protect and unprotect pages from editing, delete

> and undelete pages, rename pages without restriction, and use certain other tools."

To decide whether a topic is inside the scope of Wikipedia and therefore should have an article, the editors of each Wikipedia develop their own notability criteria. One can find the general notability criteria as an article on the English Wikipedia. *Wikipedia:Notability — Wikipedia, The Free Encyclopedia* (2016) includes the following:

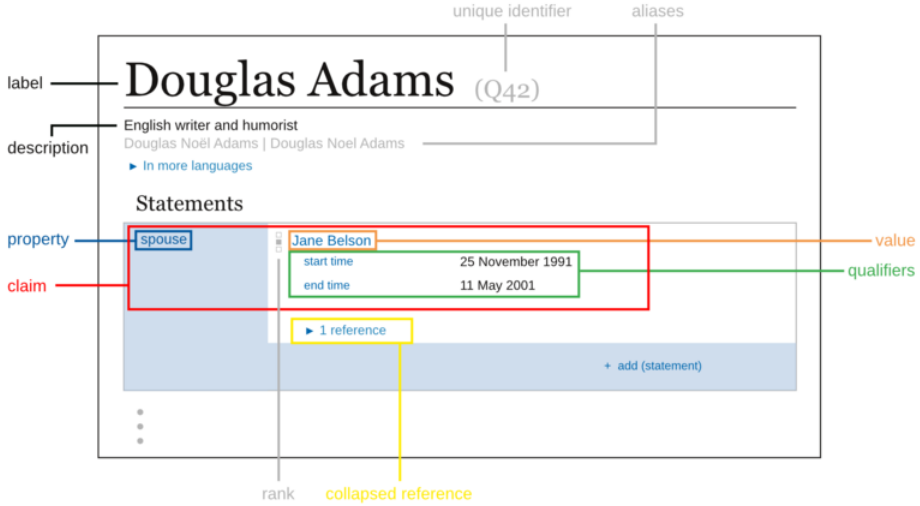
> "If a topic has received significant coverage in reliable sources that are independent of the subject, it is presumed to be suitable for a stand-alone article or list."

Part of every Wikipedia are *SpecialPages*. Those "are pages that have no wikitext, but are generated by the software on demand. They are found in the ''*Special:*'' namespace", states *Help:Special page — Wikipedia, The Free Encyclopedia* (2016).

2.2 Wikidata

Wikidata is the free and open knowledge base created by the Wikimedia movement, which collects multilingual structured data in one central place and makes it publicly available under CC0, a "Public Domain Dedication" (creative commons, n.d.), and therefore free for anyone to copy, use and distribute.

In common with Wikipedia, it can be edited by anyone. Wikidata collects data on different topics such as people, places, events and many more in so called *items*.



Douglas Adams (Q42)

English writer and humorist

Douglas Noël Adams | Douglas Noel Adams

▸ In more languages

Statements

spouse — Jane Belson

start time 25 November 1991

end time 11 May 2001

▸ 1 reference

+ add (statement)

FIGURE 2.1: Example for Wikidata item *"Douglas Adams" (Q42)*, by Kritschmar (2016)

Every item has an identifier, starting with the letter "Q" followed by a unique number. It is therefore clearly identifiable and independent of labels in natural language, which may change or be ambiguous. Additionally this offers the possibility of referring to the same item in multiple languages using only one ID.
Statements contain a *property*, which is the core part and indicates what kind of statement is made, and references to prove the claim. One or more values are attached to this property. For example, a claim for a city could be the property *"Population" (P1082)* with the respective number of inhabitants. There can be multiple values for population for different years, differentiated by *qualifiers*. To identify the important and most recent data, the values can be marked as preferred, normal and deprecated. If the claim has references proving the facts stated, it's a statement.
Data types for the properties are *item*, textitstring, textitquantity, textitname of Wikimedia Commons file, textitgeo coordinate, textittime, textitURL, and simultaneously developed to the ArticlePlaceholder extension are textitmathematical expressions, and textitidentifiers of other databases such as VIAF.
Properties and items are called *entities* in Wikidata.
Items can have *sitelinks*. Sitelinks are links to Wikipedia or other Wikimedia projects such as Wikivoyage.

The aim is to have a knowledge base that supports Wikipedia and can also be used in other contexts in the semantic web. Therefore Wikidata offers an API as well as a SPARQL endpoint. Additionally, the data can be downloaded as a full database dump in the formats JSON, XML and RDF.

2.3 MediaWiki

MediaWiki is the software used to run Wikipedia. It is a "free and open source wikipackage written in PHP" (*MediaWiki*, 2015) usually using a MySQL database. Originally it was developed for Wikipedia but it is currently used by many other Wiki projects inside and outside the Wikimedia world such as Wikimedia Commons and EVA, "the private wiki used to support operations of the International Space Station at NASA Johnson Space Center" (*EVA Wiki — Semantic MediaWiki*, n.d.). It is developed by the Wikimedia Foundation and MediaWiki volunteers. It was initially released on January 25, 2002 (*MediaWiki*, 2015). Due to the extensive use on Wikipedia, MediaWiki is optimized to handle huge amounts of data and high traffic (*Manual:What is MediaWiki? — MediaWiki*, 2013).

To adjust the software to the user's needs, it has 878 configuration settings

(*Manual:Configuration settings (alphabetical) — MediaWiki*, 2016).
MediaWiki also provides the base for many other projects such as Wikibase, the software that powers Wikidata. To enable or change various features, 2.386 extensions are currently available (*Category:All extensions — MediaWiki*, 2016), One of these extensions is Wikibase.

Pages in MediaWiki are grouped into *namespaces* (*Help:Namespaces — MediaWiki*, 2015). Namespaces are used to sort pages according to their purpose. They are indicated in the page title with a prefix such as `Namespace:PageTitle`. Pages without a prefix are in the *mainspace*. Namespaces can be localized.

2.4 Wikibase

Wikibase is an open source project, which was developed for and is mainly used by Wikidata. It is a MediaWiki extension.

Wikibase consists of two parts: a *Wikibase repository* and a *Wikibase client*.
The **Wikibase repository** (in the following text only "repository") stores and manages structured, non-relational and editable data (*Wikibase*, 2016).
The **Wikibase client** (in the following text only "client") enables the retrieval of a repository's data on a wiki (*Wikibase*, 2016). The interlanguage links for example are pulled from the repository via a shared changes database table and a maintenance script (*Extension:Wikibase Client — MediaWiki*, 2015).

In the case of Wikidata and Wikipedia, Wikidata is a repository and any Wikipedia connected to Wikidata is a client.

2.5 Lua and Scribunto

Lua is an open-source scripting language. It is fast and lightweight and is used mainly in embedded systems, often to customize other software (*Lua About*, 2016).

Lua is not object-oriented, but through its tables and meta tables can be easily adapted for multiple purposes. It is a dynamically typed language. According to Ierusalimschy (2013, 9), it supports eight basic data types: *nil*, *boolean*, *number*, *userdata*, *function*, *thread*, *string* and *table*. A wide variety of data structures such as *arrays* can be represented with Lua's tables.
Lua is used with the Scribunto extension in MediaWiki. Scribunto "[p]rovides a framework for embedding scripting languages into MediaWiki pages", according to *Extension:Scribunto — MediaWiki* (2016). At the moment it only

supports Lua.

Scribunto allows MediaWiki users to write their own Lua modules in the `Module` namespace. These can be invoked on any other wiki page (usually via *Templates*) using the following command:

```
{{#invoke:module name|function name|argument}}
```

In the case of a module called *Module:AboutTopic* with the function *showData* and a string parameter with an entity ID, calling the module on a wiki page would look like this:

```
{{#invoke:AboutTopic|showData|Q42}}
```

The code of the Lua modules is run only when the page is parsed, which means that the modules create output only when the page is edited. For this reason, modules can't handle dynamic user input (*Wikipedia:Lua — Wikipedia, The Free Encyclopedia*, 2016). Scribunto can parse Wikitext.

2.6 Template in MediaWiki

A template in MediaWiki is a page in the `Template` namespace, which is created to be included in other Wikipedia pages.

> "Templates usually contain repetitive material that might need to show up on any number of articles or pages" (*Help:Template — Wikipedia, The Free Encyclopedia*, 2016).

Examples for the usage of templates are infoboxes, explained in 3.1: Infobox. Templates can be included on any wiki content page with the following syntax:

```
{{template name|parameter}}
```

2.7 Languages online

As evident in Figure A.1 and Figure A.2, by far the most common language online is English, with 53.8% of the content, followed by Russian, German and Japanese. (W3Techs, 2016a)

Wikipedia is little different to other websites in this regard. English, Swedish, German, Dutch, French and Russian are among the top ten biggest Wikipedias, in terms of the numbers of articles as can be seen in Figure A.3.

Number of articles	Number Wikipedias
over 5.000.000	1
1.0000.000 – 4.000.000	11
1.000 - 1.000.000	227
0- 1.000	48

FIGURE 2.2: Number of articles and number Wikipedias

The English Wikipedia is the only one with over five million articles. In total, eleven Wikipedias consist of between one and four million articles. Even though there are only 48 Wikipedias featuring between zero and a thousand articles, there are 227 Wikipedias with between one thousand and one million articles, as visualized in Figure 2.2. (*List of Wikipedias — Meta*, 2016)

When compared to the numbers of first-language speakers a huge gap is clearly visible. Offline the most widespread languages are Chinese, Spanish, English and Hindi. With the exception of English, none of these languages feature in the top seven biggest Wikipedias—and the Hindi Wikipedia is only the 57th biggest.

> "Using the benchmark of 100,000 Wikipedia pages in any given language, [the 2015 Broadband Report] found that only 53 percent of the world's population has access to sufficient content in their native language to make use of the Internet relevant" found Schwab (2015).

One can conclude that there are only a few languages that are very well resourced, while a vast majority of speakers of other languages does not have access to sufficient information in their native language.

Chapter 3

Previous work

3.1 Infobox

> ”Roughly every third Wikipedia article contains an infobox—a table that displays important facts about the subject in attribute-value form” found Lange, Böhm, and Naumann (2010, 5).

Infoboxes are fixed-format tables on (in case of left to right languages) the top right of Wikipedia pages which display a summary on the data of an article’s topic (*Help:Infobox — Wikipedia, The Free Encyclopedia*, 2015). They are static tables created by editors via templates, and can display data from Wikidata, though as of today most do not. They contribute great value to the articles and allow the reader to get a quick overview of a topic.
It is important to consider how infoboxes work and differentiate the layout of the ArticlePlaceholder from theirs to avoid confusing the two features.

3.2 Reasonator

Reasonator [1] is a visualization of Wikidata items. It is written mostly in JavaScript and has a very similar approach to the ArticlePlaceholder extension. It displays items from Wikidata in a visually appealing way, aimed at readers rather than at Wikidata editors. Reasonator does not stick to the typical Wikipedia layout, but tries to find the best way possible to display the data.
Where appropriate, the items are separated into the classes "people, locations, [and] species" according to *Reasonator — Meta* (2015). Therefore it is possible to display the data in a way that is context-sensitive.

3.3 Lsjbot

In exceptional cases, the number of articles on a Wikipedia stands in no relation to the number of editors or speakers of the Wikipedia’s language.

[1] `https://tools.wmflabs.org/reasonator/`

This is especially interesting when looking at Swedish, a language with only 10.5 million speakers worldwide as Network to Promote Linguistic Diversity (NPLD) (n.d.) states. The Swedish Wikipedia is the second largest Wikipedia in number of articles. As of 15.01.2016 the vast majority (56.3%) of these articles are about taxons. In the Cebuano Wikipedia this is even more pronounced. It has the third most articles of all Wikipedias. (*Wikidata:Statistics — Wikidata*, 2015)

Cebuano is spoken in the Philippines. Cebuano speakers account for up to 24% of the country's population. (Oard et al., 2003) They are the second largest ethnolinguistic group of the Philippines with 18.5 million people (Encyclopædia Britannica, n.d.).
As of 17 January 2015, 95% of all articles on Cebuano Wikipedia are marked in Wikidata as an *"instance of" (P31) "taxon" (Q16521)* (*Wikidata:Statistics — Wikidata*, 2015).

Many of these pages were created by *bots*. A bot in the context of Wikipedia is "an automated or semiautomated tool that carries out repetitive or mundane tasks" (*Wikipedia:Bots — Wikipedia, The Free Encyclopedia*, 2015) written by editors of Wikipedia.
In the previous two examples the bot responsible for most of these pages on taxonomy is *Lsjbot*[2]. As of July 2014 it has created 2.7 million articles, "two thirds of which appear in the Cebuano Wikipedia [...]; the other third appear in the Swedish Wikipedia" (*Lsjbot — Wikipedia, The Free Encyclopedia*, 2016).
Those articles are usually *stub articles* — "article[s] considered too short to give an adequate introduction to a subject", Jemielniak (2014) explains.
Even though it helps providing much more information than these small languages would usually have access to (or their editors would be able to create) the issue remains that most of the time the communities are not able to maintain these articles by themselves. It is not possible for the limited number of editors to watch all the created stub articles and keep the data of these up to date.
Looking at Lsjbot, the goal for the ArticlePlaceholder was to centralize the data and allow editors from different communities to contribute to it and maintain it.

[2] `https://sv.wikipedia.org/wiki/Wikipedia:Projekt_DotNetWikiBot_Framework/Lsjbot/Makespecies`

3.4 Wdsearch.js

Wdsearch.js[3] is a script that includes search results from Wikidata on the Wikipedia search page. It is written in JavaScript and enabled by default on Italian Wikipedia Meijssen (2013) states.
The idea is to tackle the problem that according to Manske (2013) “over 60% of Wikidata items have no corresponding article in the English Wikipedia; once we leave the “top five”, this exceeds 90%.”
Using this script, search results from Wikidata are added to the bottom of the normal Wikipedia search result page. The label of the item contains a link to the Wikidata item. A short description gives the user an overview to determine whether it is the topic they were looking for. The user can choose to read the topic on a different Wikipedia, see multimedia content on Wikimedia Commons or see data on Resonator, using three buttons.

The script helps users find more information on a topic they are looking for, but as of now, the user is forced to visit another website. Knowledge of how to navigate and use this other website is required.
The idea of including Wikidata search results is something ArticlePlaceholders needs to incorporate.

[3] https://en.wikipedia.org/w/index.php?title=MediaWiki:Wdsearch.js&action=raw&ctype=text/javascript

Chapter 4

Personas

According to Lidwell, Holden, and Butler (2010, 182), personas are created to "understand and perfectly meet the needs of the critical few [rather] than to poorly meet the needs of many". To understand the needs of the targeted users of the ArticlePlaceholder extension, personas were developed based on the existing research on Wikipedia users (*Personas for product development – MediaWiki*, 2016) and the more general behavior of people in different countries. From these personas, scenarios and user stories were developed to derive functional and non-functional requirements that were used to implement the extension.

4.1 Rashidi: Reader of a small Wikipedia

Rashidi (29) lives in Kaduna in north-west Nigeria. His native language is Hausa, which is "a major world language with more first-language speakers than any other sub-Saharan African language—an estimated 30 million or more—most of whom live in northern Nigeria and southern areas of the neighbouring Republic of Niger" (Jaggar, 2001, p1).
Hausa Wikipedia however has only 1,356 pages. (See Appendix for a table of Hausa Wikipedia Statistic) Rashidi knows that Hausa Wikipedia exists, but he never actually uses it.
He is a musician and is very interested in music theory.
He accesses the Internet mostly on his phone. When he searches for information, he uses Google. But since there is little content in his language that he finds interesting, he rarely uses the Internet to research information. Instead, he talks with friends or refers to journals and books.

4.2 Edha: Editor of a small Wikipedia

Edha (27) is an architect in Bhubaneswar, Odisha, India. Her native language is Odia. "Odia is an Indo-Aryan language spoken by about 33 million people mainly in the Indian state of Odisha, and also in West Bengal, Jharkhand, and Gujarat.” (Ager, 2016) She speaks English also.

Edha started contributing to the English Wikipedia in 2014 and then decided to write her first article in Odia on a topic related to her field of work. She speaks both Odia and English, fluently, and in consequence translates many articles.

Odia Wikipedia has a very small community. There are hardly any people editing, since the Wikipedia has only a handful of articles, and few people are aware that a Wikipedia in their language exists. Most people prefer to read articles on bigger Wikipedias, and she is occasionally frustrated by the scant recognition her work gets due to the limited number of readers.

She does not know any programming but is very confident with a computer because of her time at the university and the fact that she outlines most of her architectural plans on the computer at work.

At home she uses a smartphone extensively to read news articles and watch movies. She edits Wikipedia articles on her smartphone, as she is only able to access a computer at her workplace.

Besides architecture, art is her main interest. She enjoys spending her free time reading about art history and going to galleries.

She got involved in Wikipedia through a friend who was already an editor and helped her during her first steps.

4.3 Julian: Editor of a big Wikipedia

Julian (17) is an active editor of Wikipedia. He is a German pupil attending his last year of high school before going on to study at a university. He is also involved in various software projects as a developer, and spends his free time programming, reading books and watching TV series.

He knows his way around computers pretty well and enjoys working with them.

Julian is an active Wikipedia editor and has been involved since 2009. His reason for contributing is his wish to improve the quality of articles and provide better access to free knowledge for everyone.

Even though he is a long-time contributor, he does not feel part of the German Wikipedia community. He has never felt comfortable with the way people discuss articles. The discussion culture is the only area of Wikipedia that could be improved in his opinion so as to make it more inclusive and less elitist.

Nonetheless, he was part of the project on Wikipedia for junior editors (*Jungwikipedianer*), knows many other editors, and enjoys discussing the most recent Wikipedia news with them in private.

His interests are mostly related to computer science and philosophy, and he uses Wikipedia to achieve a deeper understanding of these topics.

4.4 Catrin: Reader of a big Wikipedia

Catrin (46) is a social worker living in Paris, France. She works with children and adolescents and gives lectures at a university on child protection. In her scant free time she enjoys reading and going to museums. With her children majoring in different fields, she enjoys keeping up to date with their topics, too.

She is a native speaker of French and does not speak any other languages.

In early 2000 she taught children and senior citizens proper use of a computer and the Internet and therefore knows the basics of those fields. She has not caught up with the latest technologies, however.

She accesses the Internet at home from her phone.

Although in the past she would spend a lot of time in the library, nowadays due to limited time she acquires most of her knowledge online.

She has never edited Wikipedia but uses the project to read up on the topics she is interested in. She uses Wikipedia primarily to get an overview of a topic, and then uses the references for further research.

She enjoys the project and understands the structure as well as the idea that anyone would be able to edit, even though she is not contributing to the articles herself.

4.5 Heather: Wikidata editor

Heather (32) is a US-American software developer. She is mainly involved with the *OpenStreetMap* project and spends much of her time contributing to open source projects or organizing meetups for projects she is contributing to.

She mainly contributes to English Wikipedia and has written many articles over the past eleven years.

She has known about Wikidata since the start of the project and was one of the first contributors. She has a collection of bots running on Wikidata, which add data on topics she is interested in.

The thing that bothers her the most on Wikidata are missing references, which are even now often not added automatically, but have to be manually added by editors. For edits on the page without bots she uses the Wikidata games frequently, which are a "set of "games" that help to improve Wikidata" (*Wikidata Game*, n.d.).

Chapter 5

Scenarios and user stories

From the previous personas, scenarios were developed to clarify what the personas expectations for the extension could be.

> “A scenario is a story adapted and structured for engineering use. The purpose of a scenario is to communicate a situation (...) in a series of steps.” (Alexander & Beus-Dukic, 2009, 98)

Based on the scenarios below, the following user stories could be identified.

The user stories follow the style of the Connextra development team (*Connextra format*), which is

> *As a [user role],*
> *I want [function],*
> *So that[benefit]*

(*ConnextraStoryCard*, n.d.; Strong, 2013) The user stories are part of every scenario and important to identify the following non-functional and functional requirements because they abstract from the scenarios.

5.1 Scenario Rashidi

Rashidi listens to Jazz on the radio. He loves Louis Armstrong and wants to find more information about him. He uses *Google* to look up Louis Armstrong and arrives at the ArticlePlaceholder on Hausa Wikipedia. The ArticlePlaceholder displays Louis Armstrong’s date of birth (*4 August 1901*) and place of birth (*New Orleans*) (*Q1779 — Wikidata*, 2016). Rashidi can get more information on New Orleans, for example, to learn that New Orleans is a city in the United States of America.

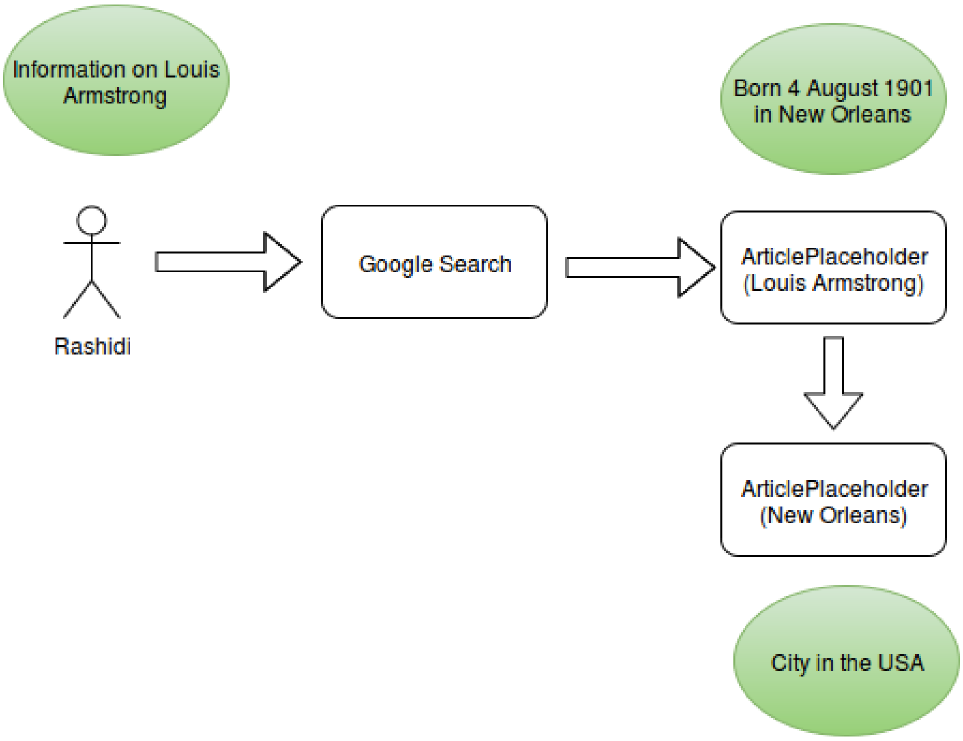

Information on Louis Armstrong
Born 4 August 1901 in New Orleans
Rashidi
Google Search
ArticlePlaceholder (Louis Armstrong)
ArticlePlaceholder (New Orleans)
City in the USA

FIGURE 5.1: Scenario Rashidi

User stories Rashidi
As a reader of a small Wikipedia, I want to access all information available in my language, so that I can use my Wikipedia without disadvantages.

As a reader of a Wikipedia, I want to search for a topic and get all information available even if there is no article yet, so that I can read up on the topic.

As an editor and reader of a very small Wikipedia, I want to have a pleasant layout of any content page, so that I do not have to adjust to new styles.

5.2 Scenario Edha

Edha talks with friends about an exhibition they have just visited. She hears about Rembrandt, a Dutch painter. As soon as she gets home she realizes she forgot to ask what century he lived in and where in the Netherlands he was born. She wants to look up the information on Wikipedia in her own language.

She opens Wikipedia on her phone and enters “Rembrandt” in the searchbar. Because there is no article on Rembrandt yet, she finds the ArticlePlaceholder for *“Rembrandt” (Q5598)* (*Q5598 — Wikidata*, 2016) in the results page. Since

it also displays the description of the item (*Dutch 17th century painter and etcher*) on the search page next to the name of the painter, she can validate with that description that she has found the right information.
The page displays data on Rembrandt, and Edha can find the place of birth on the ArticlePlaceholder, which is *Leiden*. She has not heard of the place and searches for it on Wikipedia again. The description states it is a *city and municipality in South Holland, Netherlands* (*Q43631 — Wikidata*, 2016) already on the search page. That is the information she was looking for, so she goes back to the ArticlePlaceholder. Since in the meantime someone has edited the item on Wikidata, she now finds additional information about him on the ArticlePlaceholder. It also informs her that his date of birth is *15 July 1606* (*Q5598 — Wikidata*, 2016).

In the next few days she further researches Rembrandt in her local library. She decides to create an article with this additional information. She goes back to the ArticlePlaceholder to get some of the information and references which are already there. Then she clicks the button *create an article*, which is labeled in her language. She is able to add the title for the page, which is translated *Rembrandt (painter)* in English to make clear which Rembrandt she is referring to. This title differs from the title of the ArticlePlaceholder, which is only *Rembrandt*. She can then enter the information she has gathered, and create an actual article. She would have the option to translate the article from English, too, but decides to prefers to create a new article.

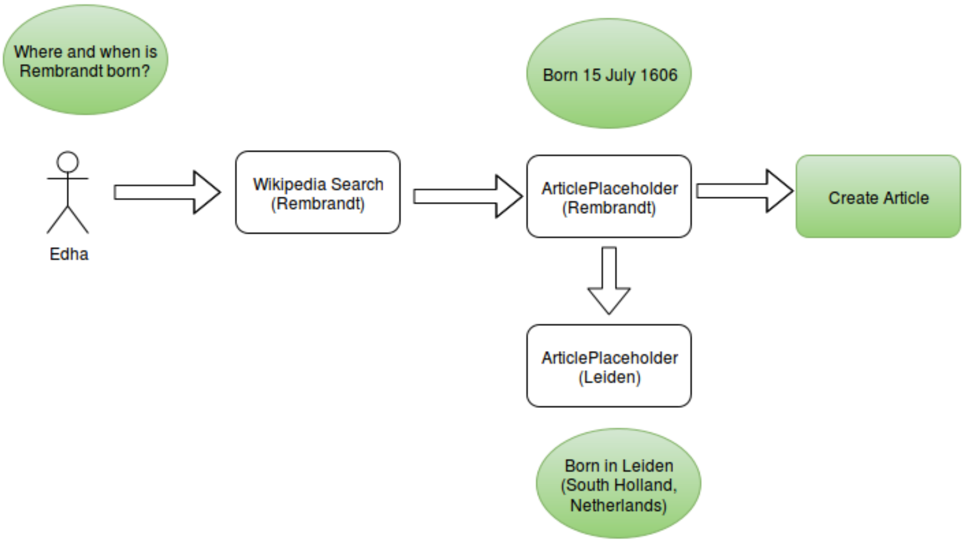

Where and when is Rembrandt born?
Born 15 July 1606
Edha
Wikipedia Search (Rembrandt)
ArticlePlaceholder (Rembrandt)
Create Article
ArticlePlaceholder (Leiden)
Born in Leiden (South Holland, Netherlands)

FIGURE 5.2: Scenario Edha

User stories Edha
As a reader of a small Wikipedia, I want any content page to work on every

device independently of screen size, so that I can access Wikipedia everywhere.

As an editor, I want to have the option to translate articles from an ArticlePlaceholder, so that I am able to contribute to the diversity of articles on my Wikipedia.

As an editor, I want to have an easy way to get from an ArticlePlaceholder to the edit page, so that I can create an article when appropriate.

5.3 Scenario Julian

Julian has heard that the ArticlePlaceholder extension has been enabled on German Wikipedia. He is aware that he will seldomly need it since German Wikipedia has quite a lot of articles.
As part of a presentation in school, he wants to research *KDE e.V.*, "a registered non-profit organization that represents the KDE Community in legal and financial matters" (KDE e.V., n.d.). He searches for it on Wikipedia and gets to the ArticlePlaceholder. He uses the data references provided there. He dislikes the way the data is displayed though, and wants to adjust the layout. He has contributed to a module before, so he knows how to write Lua code. He wants to overwrite one of the functions. After a discussion with the local community, who support Julian's changes, he adds the code after testing it, and the ArticlePlaceholder is adjusted to the community's needs.

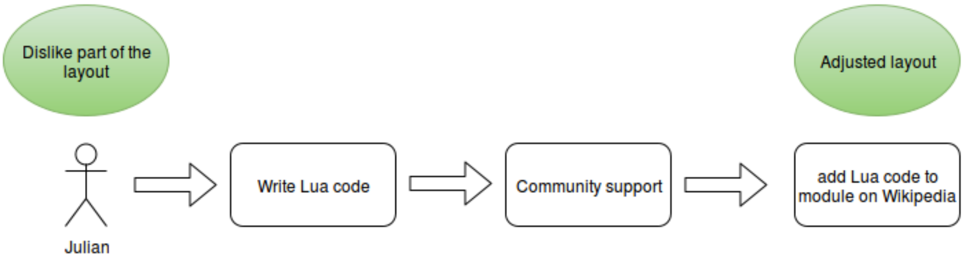

Dislike part of the layout
Adjusted layout
Julian
Write Lua code
Community support
add Lua code to module on Wikipedia

FIGURE 5.3: Scenario Julian

User stories Julian
As an editor familiar with Lua, I want to be able to adjust the layout of the ArticlePlaceholder on my Wikipedia, so that I can adjust it to the needs of my community.

As an editor, I want to influence the order in which information is shown in, so that I can adapt it to what is most important for my community. An order which may differ from that prefered by other communities.

5.4 Scenario Catrin

One of Catrin's students tells her he was born in *Kasımpaşa*. She has never heard of it and wants to research it back at home on French Wikipedia. She enters the term in the search bar. Since there is no article on the French Wikipedia, an ArticlePlaceholder is displayed. It has no description, so she can only see the name on the search page. She learns it is an *instance of neighborhood* and is *located in the administrative territorial entity Beyoğlu*. Since she has not heard of Beyoğlu yet either, she continues her research and arrives at the French Wikipedia article Beyoğlu. There she finds out Beyoğlu is a district of Istanbul, Turkey.

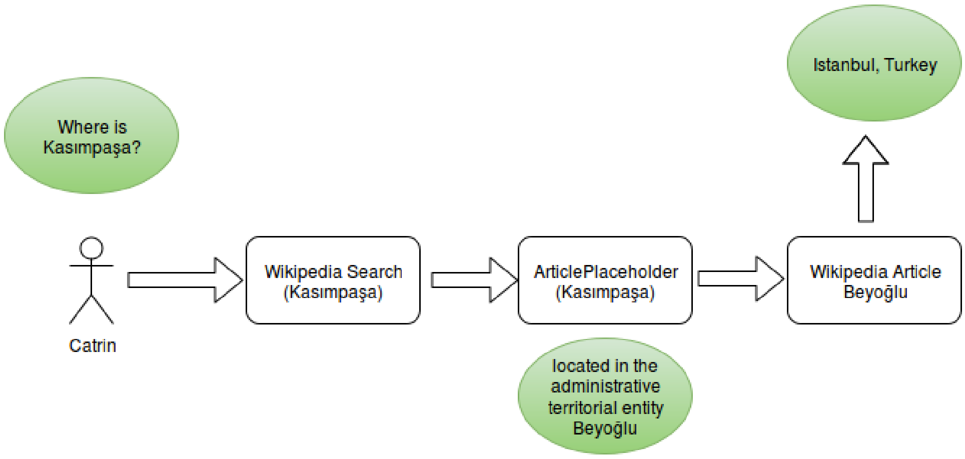

Where is Kasımpaşa?
Catrin
Wikipedia Search (Kasımpaşa)
ArticlePlaceholder (Kasımpaşa)
located in the administrative territorial entity Beyoğlu
Wikipedia Article Beyoğlu
Istanbul, Turkey

FIGURE 5.4: Scenario Catrin

User stories Catrin
As a reader, I want to see the most important information on a page first, so that I do not have to read through everything to get to what really interests me.

As a reader, I want to be able to clearly differentiate between a generated content page and a manually written article, so that I can classify the information in the appropriate context.

5.5 Scenario Heather

Heather has worked on the Wikidata item *"Ada Lovelace" (Q7259)* over the last few days. She has added data and references mostly by hand. Heather wants to see what the item would look like on a Wikipedia that does not have an article on *Ada Lovelace* yet.
She is involved with various small language Wikipedias and even though she does not speak the language, she can find the SpecialPage, which will lead her to an ArticlePlaceholder. Then Heather enters the item ID there to

see what it looks like.
On entering the same item ID on English Wikipedia, she is redirected to the existing article on *Ada Lovelace*.

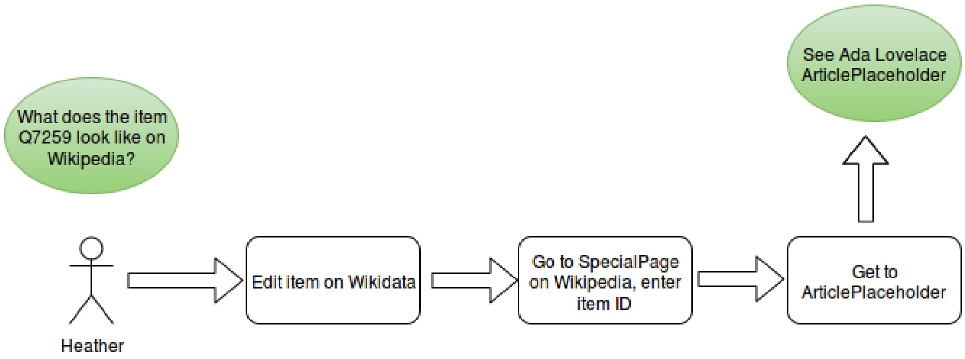

What does the item Q7259 look like on Wikipedia?
Heather
Edit item on Wikidata
Go to SpecialPage on Wikipedia, enter item ID
Get to ArticlePlaceholder
See Ada Lovelace ArticlePlaceholder

FIGURE 5.5: Scenario Heather, small Wikipedia

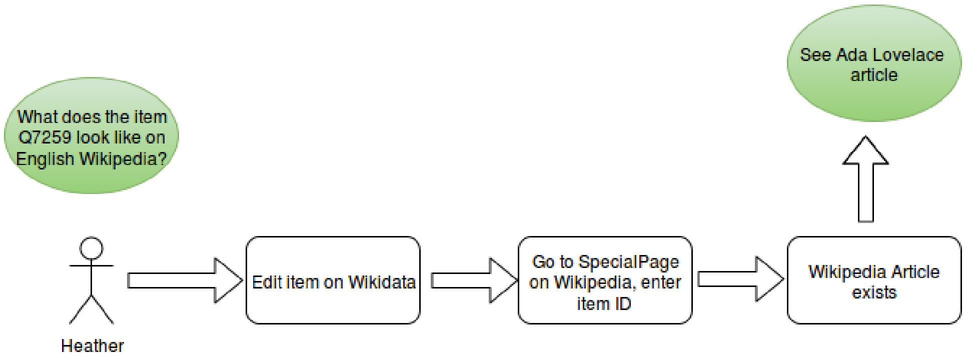

What does the item Q7259 look like on English Wikipedia?
Heather
Edit item on Wikidata
Go to SpecialPage on Wikipedia, enter item ID
Wikipedia Article exists
See Ada Lovelace article

FIGURE 5.6: Scenario Heather, English Wikipedia

User stories Heather

As a reader familiar with Wikidata, I want to see the ArticlePlaceholder of a certain item, so that I can link to the placeholder rather than the Wikidata item.

Chapter 6

Non-functional requirements

The extension aims to provide access to previously unavailable information to readers of Wikipedia.
The requirements should all follow this main objective.
The user should be able to discover the ArticlePlaceholder-generated pages while browsing the Wikipedia or researching a certain topic.
While the content pages are auto-generated, editors of the Wikipedia projects should still have a sense of ownership regarding the content. They need to keep control over the pages as much as possible, so as to be able to adjust them to their community's needs.
At the same time the extension should provide a default that makes it attractive and engaging to use. This is especially true for speakers of the target languages that are under-represented in Wikimedia projects, which have small communities and a small number of articles. The smaller Wikipedias often do not have enough editors to write content, so the editors can't be expected to make the effort of being involved with the technical aspects, too.
The pages should engage and encourage editing when appropriate. Therefore, they should invite readers to create articles - possibly with the information provided by Wikidata. In addition, they need to have an option to translate existing articles from any other language the reader may speak.
Wikipedia is a large and widely used project, and has built a reputation as a trustworthy source of knowledge. The extension must maintain this status.

The ArticlePlaceholders must clearly differ from human-written articles in order to make the reader aware when evaluating the content that it is auto-generated from pure data. It is also important to clarify that this is not a static, finished page and should motivate the reader to participate.

Chapter 7

Functional requirements

The functional requirements are based on the non-functional requirements. Additional requirements are derived from the technical environment and its specifications.

To be able to display the data of Wikidata, ArticlePlaceholder are based on Wikibase and MediaWiki.
As MediaWiki is written in PHP, the main part of the extension has to make use of that language and the services provided by MediaWiki and Wikibase. The programming language for scripts on Wikipedia is Lua. Therefore the parts that will be user-editable are written in Lua, so they can be overwritten by the local communities.
The default for the content pages presents the information and data in a useful way, and to make it visually appealing, it should make use of images when possible.
For users without access to a computer (as in the scenario of Edha) the layout of the extension is required to be responsive and work on mobile devices.
Some of the following requirements are scheduled for later implementation during the course of the project.

7.1 Naming

Since the users need to be able to find the ArticlePlaceholder easily it is implemented as a SpecialPage. This SpecialPage should have a short name that is convenient to remember. Therefore it is called *AboutTopic*.

7.2 Display ArticlePlaceholder

The pages displaying the content of a Wikidata item focus on a clear differentiation between the layout of the ArticlePlaceholder and what a user-written article looks like. They still contain elements of the well-known Wikipedia layout to appeal to readers and fit in with the other content as the references at the bottom.

A default layout is needed, but it is important to give the user control over every part of the page. In doing so, users such as Julian are enabled to adjust the layout to their communities' needs.

7.3 Internationalization

It is important to have the whole extension ready to be localized from the beginning. This is necessary for the extension to be used by an international community.

> "Internationalization refers to the adaptation of products to support or enable localization for international markets. Key features of internationalization have always been the support of international natural language character sets, separation of locale-specific features such as translatable strings from the software code base and the addition of functionality or features specific to foreign markets. Without internationalization, localizing a product can be very challenging" states Esselink (n.d., 23).

Since the data on Wikidata is multilingual, this focuses more on aspects like labels on buttons and the name of the SpecialPage itself.

7.4 Discover ArticlePlaceholder

There are multiple ways for a reader to get to an ArticlePlaceholder.

7.4.1 SpecialPage URL

Beside the input form of the SpecialPage there should be a way to directly link to an ArticlePlaceholder page as users as Heather would require.

7.4.2 Search

In order to get to an ArticlePlaceholder while looking for a topic, the generated pages are added to the search on Wikipedia. There are different possible positions for the results in the search page. They could be attached to the bottom of the results, or placed to the right of the regular search results.

It is important to point out that the notability criterias on a Wikipedia are not necessarily the same as on Wikidata. The results in the search should be limited to notable items.
Showing only notable items is especially important since article creation should only be encouraged when it makes sense. This avoids the creation of

articles, that would be deleted by other users later on. Otherwise the community building and trust the ArticlePlaceholer is supposed to be working on would be compromised.
Since it is hard to decide which content is actually notable, the items appearing in the search should be limited to the ones having at least one statements and two sitelinks to the same project (like Wikipedia or Wikivoyage).

Catrin's scenario shows how a user would start their research with an ArticlePlaceholder. Therefore whenever possible the ArticlePlaceholder should link to existing Wikipedia articles. Due to the limitation of the data received from Wikidata, this is not in the scope of this thesis.

Many users will research information on web search engines such as Google and arrive at Wikipedia from the result page. In order to make the data provided by the ArticlePlaceholder accessible for those users, it will be necessary for the generated pages to be indexable.

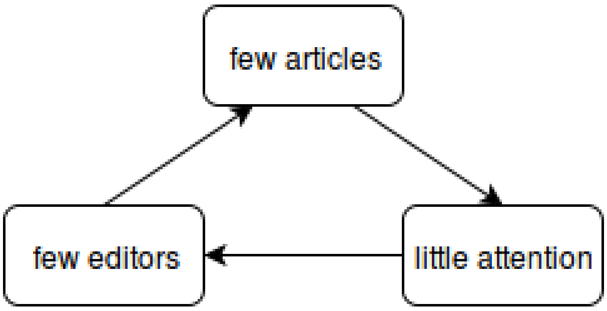


FIGURE 7.1: Problem for small Wikipedias

As described in Figure 7.1 a problem of the small Wikipedia projects is being stuck in a vicious circle: Since they have few articles, they do not receive a lot of attention via search engines such as Google, so they are not able to attract new editors and therefore they do not consist of many articles.

Making ArticlePlaceholder accessible for search engines will be an important point in the following development of the extension.

7.4.3 Red links

> "A red link (...) signifies a link to a page that is either non-existent or deleted." (*Wikipedia:Red link — Wikipedia, The Free Encyclopedia*, 2015)

Red links are used frequently in Wikipedia to indicate an article which is does not yet exist, but should. Today it leads the user to an empty *create article* page. In the future it should instead bring them to an ArticlePlaceholder, offering the option of creating an article. This is part of the topic of *smart red links*, which is discussed in the section 8.1: Smart red links.

7.5 Article creation

Starting at the ArticlePlaceholder created by the extension, the user may want to create an article to provide more information and give this data a context that only natural language can provide.
Two different strategies are in consideration at the moment: Creating an article from scratch, or translating an existing article into the user's language. The challenge is to encourage editors to add a new article if appropriate, and at the same time prevent vandalism such as "any addition, removal, or change of content, in a deliberate attempt to damage Wikipedia. Examples of typical vandalism are adding irrelevant obscenities and crude humor to a page, illegitimately blanking pages, and inserting obvious nonsense into a page." (*Wikipedia:Vandalism — Wikipedia, The Free Encyclopedia*, 2016)
The title for the article they want to create might differ from the label of the item, due to the local Wikipedia's rules and customs. Therefore a user must be able to enter a title for the page that does not exist on that Wikipedia yet. After entering the title of the new article, the user is redirected to an empty editing page, where they can add their content.
In the future it is intended to add the data provided by Wikidata to the editing page so the user can use it while writing the article.

7.5.1 Content translation

Using the content translation tool[1], the user should be able to translate an article from an existing one in another language. This is not within the scope of this thesis; the focus for now is to create articles from scratch.

7.6 Ordering of statement groups

The data in Wikidata is not ordered in a meaningful manner. ArticlePlaceholder aims at readers looking for an overview of the topic and therefore only need the most important data on a topic.

To keep the editors in control, there should be a list of properties maintained on-wiki. They are ordered by their importance.
The ordered list includes labels of the properties as well as their numbers for users unfamiliar with Wikidata. The list can be annotated with comments

[1] `https://www.mediawiki.org/wiki/Content_translation`

on each line to provide additional information and help other editors understand the context of the property and its position in the ordered page.
This list is only editable by administrators on the Wikipedia so as to reduce constant changing of the displayed information.
This list is then used to order and filter the statements on each ArticlePlaceholder page.

7.7 Identifier

Identifiers are IDs used for an entity in other ontologies. For the reader to be able to conveniently look up more information on sites other than Wikimedia projects, identifiers need to be distinguishable from other statements also in the layout, and always show up on the ArticlePlaceholder. To always be displayed, they should not be included in the list of ordered statements.

7.8 Caching

Since the generated content pages will be accessed frequently, it will be necessary to implement some form of caching for them. In Figure A.4 is an example for the page views per day of the English Wikipedia's *MainPage* to emphasize how urgent the need for caching will be.
With caching it would be possible to limit the amount of requests to Wikidata services too, which will show an improvement in the performance. This is not part of this thesis.

7.9 Unit tests

It is especially important for the main code in PHP to be covered completely by automated testing, especially by unit tests.
It is necessary to set up continuous integration in a way that is integrated with MediaWiki and Wikibase. Then the code will conform to coding conventions, but also allows for running the unit tests automatically. This will reduce the time spent on code reviews, testing and bug fixing.
This has been done in cooperation with the Wikidata team while working on the extension. Neither the integration tests nor the tests for the Lua code are covered in this thesis.

Chapter 8

Implementation

Due to MediaWiki being written in PHP and requiring its extensions to be in the same language to use its functionality, the core part of the ArticlePlaceholder extension is written in PHP. The programming language PHP is “a popular general-purpose scripting language that is especially suited to web development” (*PHP*, n.d.).

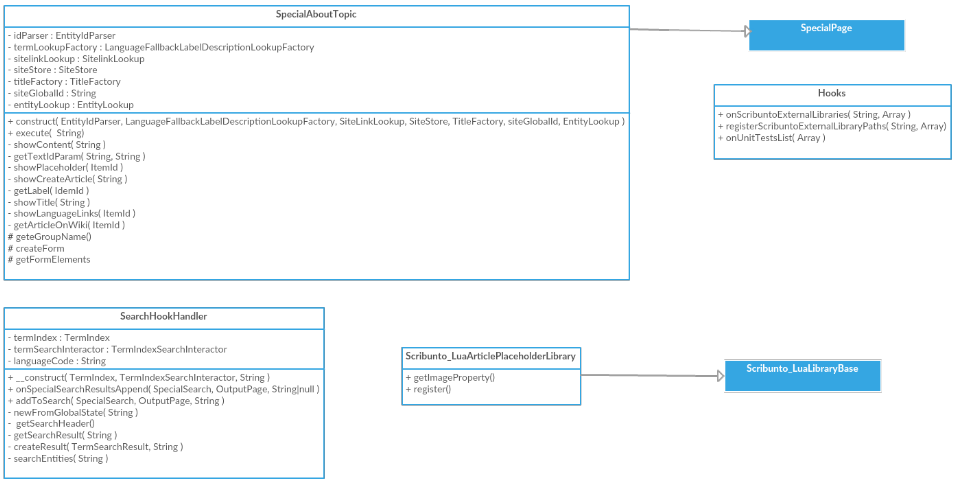

SpecialAboutTopic
- idParser : EntityIdParser
- termLookupFactory : LanguageFallbackLabelDescriptionLookupFactory
- sitelinkLookup : SitelinkLookup
- siteStore : SiteStore
- titleFactory : TitleFactory
- siteGlobalId : String
- entityLookup : EntityLookup
+ construct(EntityIdParser, LanguageFallbackLabelDescriptionLookupFactory, SiteLinkLookup, SiteStore, TitleFactory, siteGlobalId, EntityLookup)
+ execute(String)
- showContent(String)
- getTextIdParam(String, String)
- showPlaceholder(ItemId)
- showCreateArticle(String)
- getLabel(IdemId)
- showTitle(String)
- showLanguageLinks(ItemId)
- getArticleOnWiki(ItemId)
# geteGroupName()
# createForm
# getFormElements
SpecialPage
Hooks
+ onScribuntoExternalLibraries(String, Array)
+ registerScribuntoExternalLibraryPaths(String, Array)
+ onUnitTestsList(Array)
SearchHookHandler
- termIndex : TermIndex
- termSearchInteractor : TermIndexSearchInteractor
- languageCode : String
+ __construct(TermIndex, TermIndexSearchInteractor, String)
+ onSpecialSearchResultsAppend(SpecialSearch, OutputPage, String|null)
+ addToSearch(SpecialSearch, OutputPage, String)
- newFromGlobalState(String)
- getSearchHeader()
- getSearchResult(String)
- createResult(TermSearchResult, String)
- searchEntities(String)
Scribunto_LuaArticlePlaceholderLibrary
+ getImageProperty()
+ register()
Scribunto_LuaLibraryBase

FIGURE 8.1: Class diagram of PHP classes

Therefore all parts that need services provided by MediaWiki or otherwise work with the functionality of MediaWiki must be written in PHP—such as the SpecialPage for users to create an ArticlePlaceholder as well as the integration in the search.

The ordering of statement groups is written mainly in PHP in order to be reusable—and even though it was developed for this extension, it will be part of the Wikibase code. It could have been implemented in Lua as well, but then it would have been useful almost solely for this extension.

The layout and display logic of the ArticlePlaceholder on the Wikipedia—the most important part from a user's perspective—has been implemented as a Lua module. This module is similar to libraries in other programming languages since it is overwritable by users with setter and getter functions. It is invoked from the SpecialPage via a Template (`Template:AboutTopic`).

The template calls the `Module:AboutTopic`'s function `showData`. The named template and module are on the respective language Wikipedia.

8.1 Smart red links

Red links on Wikipedia display the title of an article that does not exist yet, but is desired, by linking to an empty editing page. It is difficult to connect a title to a Wikidata item since the item does not necessarily have the same label as the title of the page. Therefore connecting red links to ArticlePlaceholder is rather demanding, and not within the scope of this thesis.
However, there is an ongoing discussion on how this could be realized, which is summarized in this chapter (*Extension:ArticlePlaceholder/Smart red links — MediaWiki*, 2016).

The ideal plan would be to have *smart red links* which are connected to a Wikidata item. In order for this to work, Wikidata sitelinks could be used. Basically there would be *virtual sitelinks*, so editors could connect links to non-existing articles on Wikidata the same way that links to existing articles would be added. This way, the text of the article and its Wikitext would not change. On Wikidata, those links could either be in an own section or be displayed in red.
There are multiple ways for editors to add those sitelinks to Wikidata. Existing red links would have to be migrated by editors. Clicking a red link that is not yet connected could lead the editors to a page asking them for the same article in another language and then automatically add the red link to the corresponding Wikidata item. If none exists there would still be the possibility to just add the Wikidata item ID.
If editors add a red link, they could be asked to add it to the corresponding Wikidata item before saving their edit.
In the case of an editor using content translation, the link would be added automatically since the article is connected to an article in another language.

As this would also give editors an overview over wanted pages in Wikidata, there would be more benefits than only having ArticlePlaceholder connected to red links, such as being able to find translatable articles easily.

8.2 Setting up the extension

To enable and configure an extension in MediaWiki, the extension needs to contain a file called `ExtensionName.php`, which in this case would be

`ArticlePlaceholder.php`. `ArticlePlaceholder.php` contains declarations and definitions that are important for this extension.
To register the extension in MediaWiki, in the file `LocalSettings.php` the ArticlePlaceholder is included with `require_once "$IP/extensions/ArticlePlaceholder/ArticlePlaceholder.php";`.
Additionally the extension uses the global variable `$wgArticlePlaceholderImageProperty`. This is needed for the user installing the extension to specify the image property of their repository (*Manual:Extension registration — MediaWiki*, 2016). In the case of Wikidata this would be *"image" (P18)*.

8.3 SpecialPage

The SpecialPage `Special:AboutTopic` uses multiple services provided either by Wikibase or MediaWiki. It also extends the class `SpecialPage` provided by MediaWiki. The class is called `SpecialAboutTopic` and can be found in the namespace `ArticlePlaceholder\Specials`.
The SpecialPage serves two purposes: It creates the form for the SpecialPage itself, and is used to display the ArticlePlaceholder.
From a user's perspective there are two ways of passing an entity ID to the extension. The first option is to enter the ID on the SpecialPage. The SpecialPage is created by an HTML form in the function `createForm()`, which adds HTML to the `OutputPage` object provided by MediaWiki.
When the entity ID is passed, there is a check with the `EntityLookup` provided by Wikibase whether this entity actually exists.
If the entity ID does not exist, the user will get the form again with an additional error message.

The second option to get to an ArticlePlaceholder is to pass the ID directly via the URL, for example in the following way.

```
Special:AboutTopic/Q5279
```

If the entity ID exists, the SpecialPage needs to check with the `SiteLinkLookup` service by Wikibase whether there is already an article on this wiki for the item. The function `getArticleOnWiki( $entityId)` is called. When an article exists, it returns the title of that article. The user is then forwarded to that article with the `redirect( $url )` function of the `OutputPage` class. The flowchart for the described code can be seen in Figure 8.2.

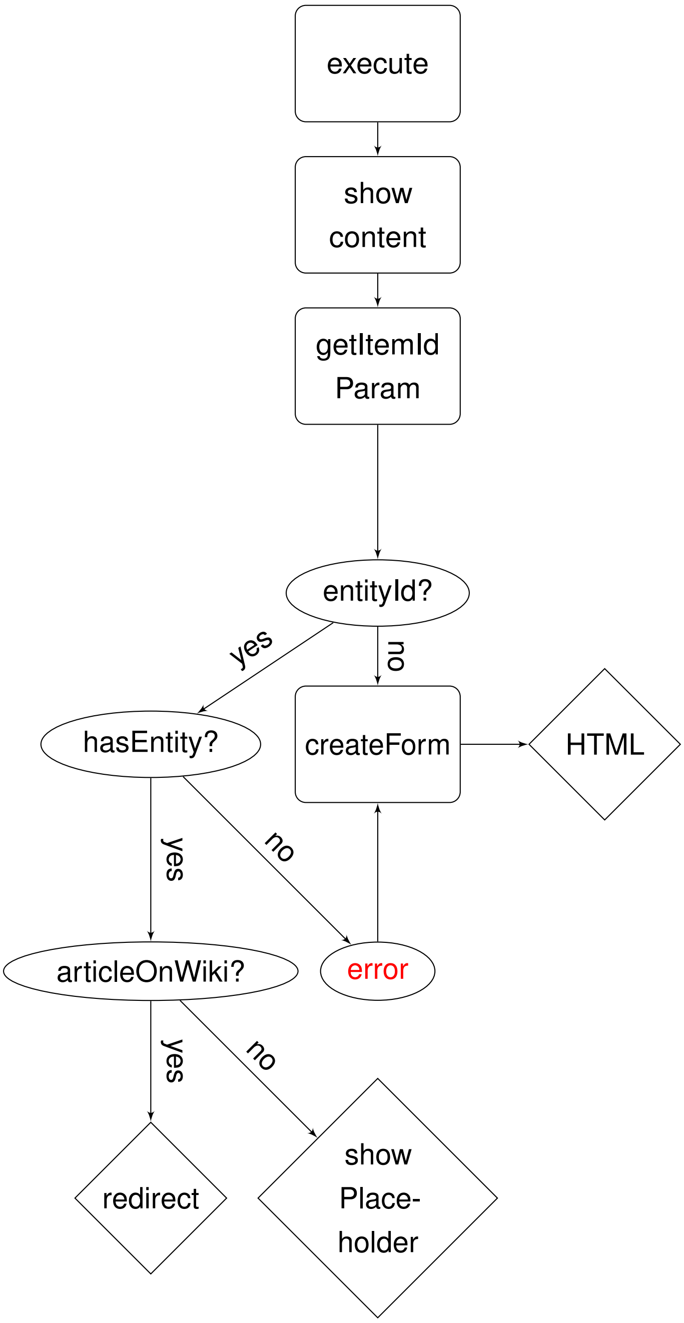


FIGURE 8.2: Flowchart: ArticlePlaceholder creation

If there is no article for the passed item, then an ArticlePlaceholder is created. In order to do this, a template on the wiki is invoked, which calls the Lua module. Additionally `OOUI` is enabled, which is needed for the button creation in PHP. The label of the item is passed to the JavaScript module `ext.articleplaceholder.createArticle`. The label is used in PHP also to set the title of the page. Links to other Wikipedia language versions (language links) are set in PHP. The sitelinks are read from the `SiteStore` service provided by Wikibase. They are constructed by combining the language code and the page name, with a colon in between. For example, the page linking to English Wikipedia article on "Ada Lovelace" would be

`en:Ada Lovelace`.
Showing a placeholder can be described as in Figure 8.3.

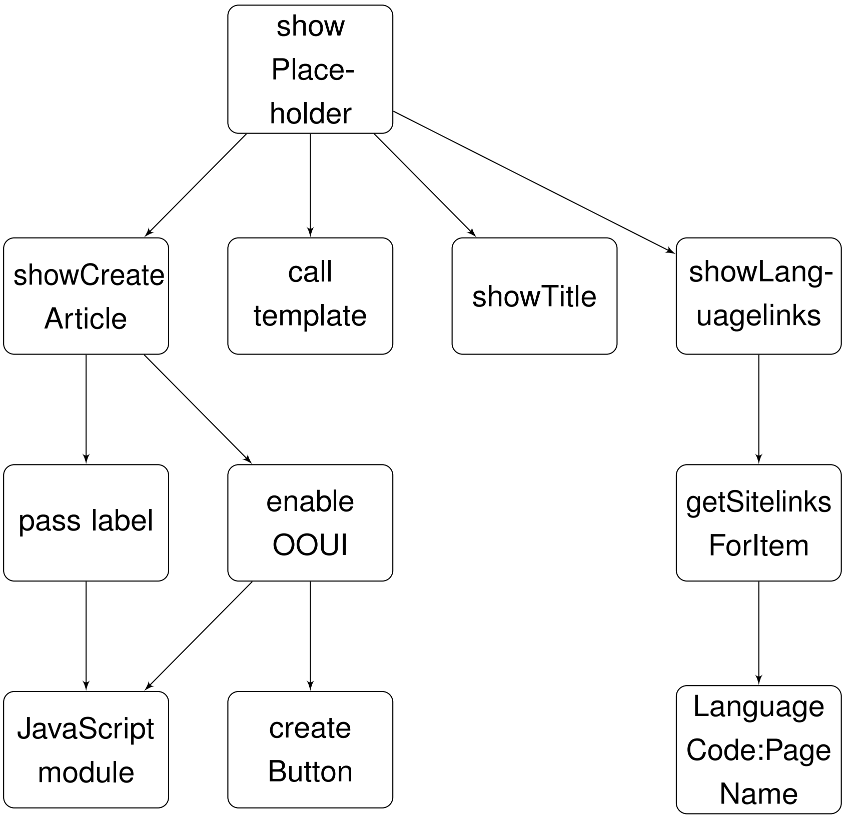
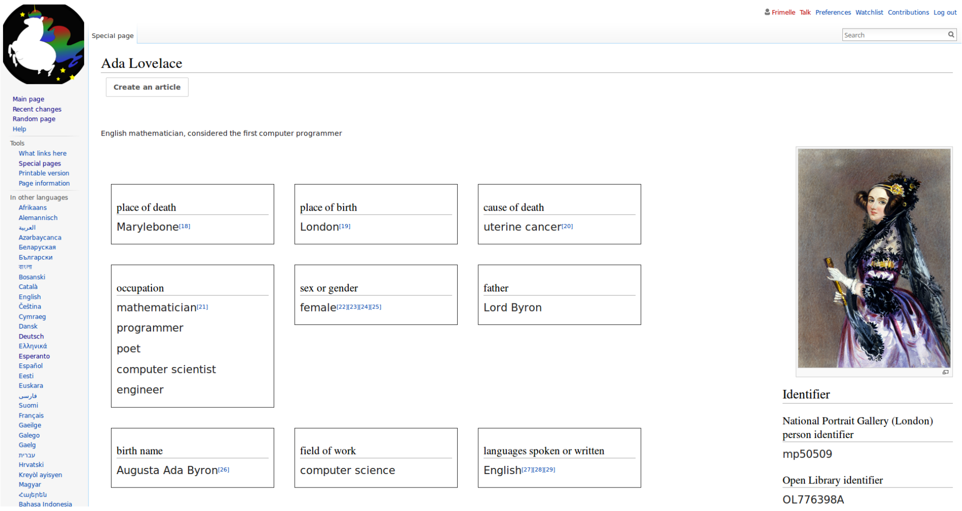


FIGURE 8.3: Flowchart: Showing an ArticlePlaceholder

8.4 Display

The display is one of the main parts of the ArticlePlaceholder, as it is user-facing and therefore needs an initial default design.



Ada Lovelace

English mathematician, considered the first computer programmer

place of death
Marylebone[18]

place of birth
London[19]

cause of death
uterine cancer[20]

occupation
mathematician[21]
programmer
poet
computer scientist
engineer

sex or gender
female[22][23][24][25]

father
Lord Byron

birth name
Augusta Ada Byron[26]

field of work
computer science

languages spoken or written
English[27][28][29]

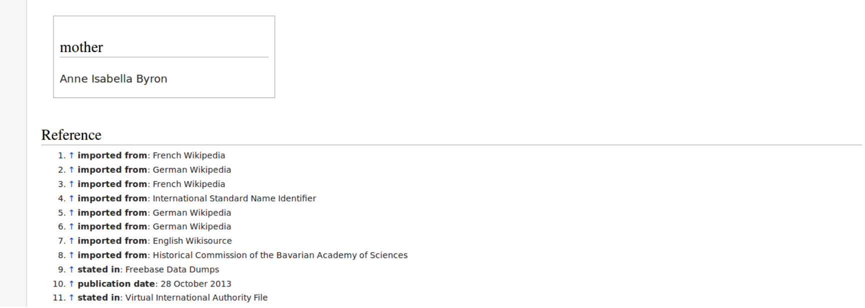
Identifier

National Portrait Gallery (London) person identifier
mp50509

Open Library identifier
OL776398A

FIGURE 8.4: Screenshot of the ArticlePlaceholder layout as of January 2016

The core part of the page is formed by the statement groups. These consist of a property, one or multiple values, and their qualifiers.
Each statement group is in a box with a black border and arranged in a tile layout. The number of statement boxes per row differs depending on the width of the screen.



mother
Anne Isabella Byron

Reference

1. ↑ **imported from**: French Wikipedia
2. ↑ **imported from**: German Wikipedia
3. ↑ **imported from**: French Wikipedia
4. ↑ **imported from**: International Standard Name Identifier
5. ↑ **imported from**: German Wikipedia
6. ↑ **imported from**: German Wikipedia
7. ↑ **imported from**: English Wikisource
8. ↑ **imported from**: Historical Commission of the Bavarian Academy of Sciences
9. ↑ **stated in**: Freebase Data Dumps
10. ↑ **publication date**: 28 October 2013
11. ↑ **stated in**: Virtual International Authority File

FIGURE 8.5: Screenshot of the ArticlePlaceholder layout, Reference section as of January 2016

The main image of the item is on the right side (for left to right languages) of the page, where it would also be in articles.
The label of the property is the header for each group, which is separated from the body by a horizontal line. Multiple values for the same statement are listed. Their qualifiers are directly under the corresponding value in a smaller font size.

References can be found in their own section at the bottom of the page, conforming to the existing layout of Wikipedia, see Figure 8.5.
The identifiers of an item should be in a distinct position so as to be both emphasized and distinguishable from the other statements. They are listed below the main image of the article to the right of the statement groups.

8.4.1 Renderer

Every part of an entity—or item in the case of the ArticlePlaceholder—needs a *renderer* to be displayed. The renderers are written in Lua. They pull data from Wikidata and render it to HTML, and set CSS classes for the HTML elements, in order to style them with the CSS module `ext.articleplaceholder.defaultDisplay.css`.

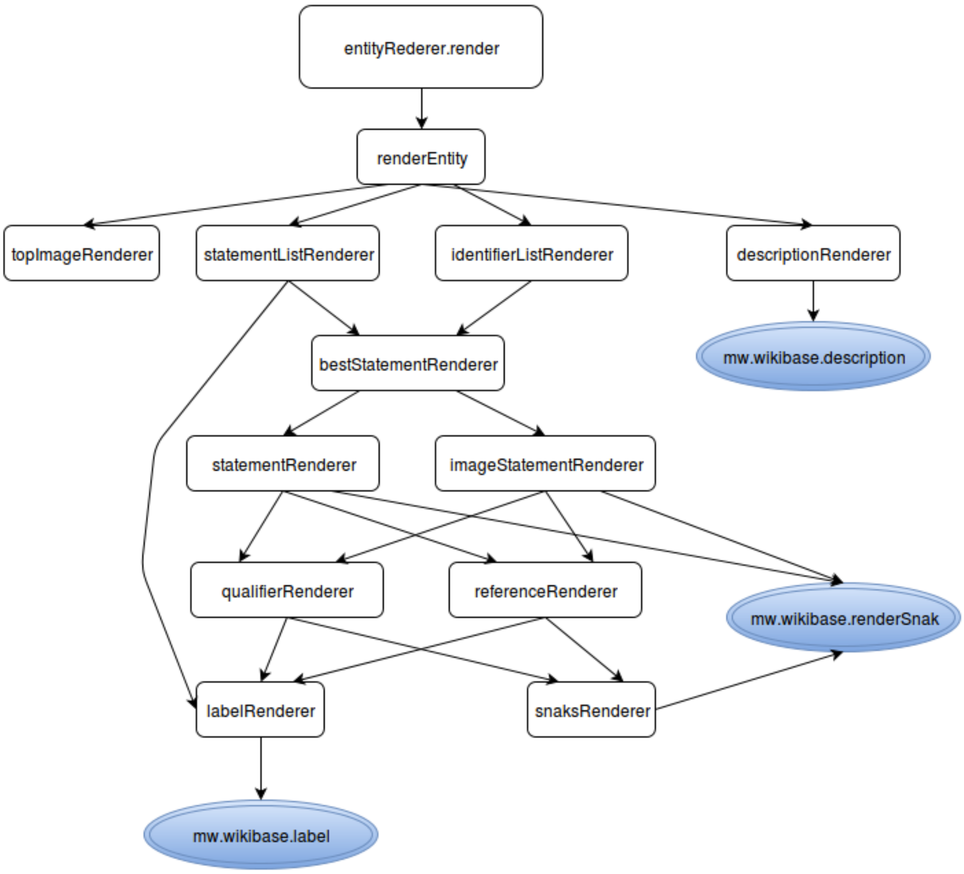


FIGURE 8.6: Diagram of all renderer functions

The HTML output of the module is displayed on the `Special:AboutTopic` ArticlePlaceholder.
The MediaWiki conventions for naming Lua modules is

`mw.ext.extensionName.moduleName`

In order to match them, the default renderer for ArticlePlaceholders is called

`mw.ext.articlePlacholder.entityRenderer`

For individual renderers the entity renderer has getter and setter functions. Therefore every part of the display logic can be overwritten locally on Wikipedia.

On the page `Module:AboutTopic` on the Wikipedia is the Lua module which is invoked by `Template:AboutTopic`. On the entity renderer instance obtained, a user with the appropriate rights can call the getter and setter functions.

The renderers are as interchangeable as possible, so that for example the `snaksRenderer` can be used by the `referenceRenderer` as well as the `qualifierRenderer`. Therefore the scopes of the renderers are therefore limited to their functionality.

Wikibase provides a Scribunto interface with functions to access the repository. Whenever possible, the functions provided by this interface were used. This way, the description renderer for example simply wraps `mw.wikibase.description`. The `descriptionRenderer` of the ArticlePlacheolder's EntityRenderer is still needed as a separate service, since there must be a possibility for the user to overwrite the renderer. Additionally, a CSS class is wrapped around the result of Wikibase's description renderer in order to style it properly.

The entity ID of the item, an ArticlePlaceholder is created for, is passed to the EntityRenderer module. The Lua table for the entity is loaded in the `renderEntity` function. This table represents the whole entity with all its respective data. Therefore it is a very performance-costly operation.

8.4.2 Identifiers

To get a list of all external identifiers in Wikidata, the item *"Wikidata property representing a unique identifier" (Q19847637)*, is used. All properties representing an identifier are an *"instance of" (P31)* this item.

To query all those identifiers, Wikidata's SPARQL endpoint[1] is used. The following query returns a list of IDs. This list can be downloaded as a file with the data in its JSON representation.

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT ?identifier WHERE {
   ?identifier  wdt:P31 wd:Q19847637 .
}
```

[1] https://query.wikidata.org

In the Lua script `Main.lua` the script `Identifier.lua` is generated from the JSON output.
`Identifier.lua` returns a Lua table, which contains tables with the identifier mapped to `true` to check with an average of `O(1)` whether a property is an identifier or not. This script is included in the Lua module and is used to get access to the table with the identifier. It is static, as it would cause too many requests to the SPARQL endpoint to check for every call to the SpecialPage again if the data on the identifier changed. This means that if the identifiers on Wikidata are edited, or a new one is added, the ArticlePlaceholder would not immediately reflect this change.

This implementation was chosen because there was no data type for identifiers in Wikidata while the extension was developed. As of now it has been introduced and is deployed now.
Until the data type was deployed, it was sufficient to have a static list of property IDs in the extension. This list includes all identifiers used in Wikidata, and will be exchanged with the proper handling of all statements with the identifier data type. The method for checking a property ID can be adjusted to check for the datatype of the respective property.

The code to create the list of identifiers is not part of the extension, only the created file `Identifier.lua` is included in the extension, in the path `ArticlePlaceholder/includes/Lua`. The source code for creating the list can be found on GitHub[2].

8.4.3 Create article button (JavaScript

For the layout it is necessary to have a button at the top of the page asking for the creation of the article.
The JavaScript module to create this button uses *OOjs UI*, which is included in MediaWiki and "is a library that allows developers to rapidly create front-end web applications that operate consistently across a multitude of browsers" (*OOjs UI — MediaWiki*, 2015).
A dialog appears and asks the user for a title for the new article. The default title refers to the label of the item, which can be adjusted.
An option for users who have JavaScript disabled is intended. But since the number of users with JavaScript disabled is about 3% (*Analytics/Reports/-Clients without JavaScript — MediaWiki*, 2015) the priority is rather low and is not considered for implementation now.

Via the function `showCreateArticle( $label )` the button is added

[2] `https://github.com/frimelle/identifier-to-lua`

to the page in the PHP class `SpecialAboutTopic`. The label of the ArticlePlaceholder is passed in order to display it in the dialog window that is created in the JavaScript module `ext.articleplaceholder.createArticle`.
In this module, an event listener is added to the button so it will open the dialog window on click, the `CreateArticleDialog`. This dialog has an input field, displaying the label of the item. This string can be changed by the user. In the dialog is an additional submit button, which is an instance of the class `OO.ui.ButtonWidget`. In order to redirect the user to an edit page when the submit button is clicked, an event listener is added to the button. Before redirecting there is a check to see if the page that is supposed to be edited already exists. If that's not the case, the user is redirected with the help of `document.location.href`.
In the case of the page already existing, an error message is added to the dialog window at the bottom, and the user is required to enter a new title.

8.4.4 Style elements (CSS)

In order to style the layout elements, CSS classes are assigned in the Lua module. Their looks as well as the create article button's look are adjusted in `ext.articleplaceholder.defaultDisplay.css`.
So as not to conflict with other MediaWiki style elements, the CSS classes are prefixed with `articleplaceholder-`.
The `divs` are nested so that, for example, the main image and the identifier (which both have their own `div`) are in one `div` named after its position in the layout `articleplaceholder-sidebar`. Additionally the identifiers are all in one `div`, the `articleplaceholder-identifierlist`.
The number of columns containing the statement groups differs depending on the number of statements. When the browser window is smaller, the number of boxes per row adjusts accordingly. Initially it was planned to adjust them to a tiling layout, but since tiling layout in pure CSS would expect the boxes to be ordered vertically in columns, this was not possible. It is crucial to show the most important information first, otherwise the ordering of statement groups would be pointless.
In order to be responsive, the extension makes use of *media queries*. Media queries are a convenient way to add CSS styles for elements that need to be adjusted for different devices (Hay, 2013, 43).
In MediaWiki with the *resource loader* `$wgResourceModules` media queries can be assigned to CSS modules. This allows loading another CSS module (`ext.articleplaceholder.defaultDisplaySmall.css`), when the screen size is smaller then 930 pixels. This prevents the overlapping of the sidebar with the image and identifiers and the statement groups.

8.5 Search

In order to be able to find results from the ArticlePlaceholder in the MediaWiki search, it is necessary to use the `SearchHookHandler` provided by MediaWiki (*Manual:Hooks/SpecialSearchResultsAppend — MediaWiki*, 2015). With the help of Wikibase's `TermIndexSearchInteractor`, items with labels or aliases matching the given search term are displayed. To be accessible to both client and repository, the `TermIndexSearchInteractor` and its respective tests had to be moved from Wikibase repository to *Wikibase lib*, which "provides common functionality for Wikibase and Wikibase Client" (*Extension:WikibaseLib — MediaWiki*, 2015). This was done by the author in a *commit* to the Wikibase *Gerrit repository* (*Move the TermIndexSearchInteractor to lib — Gerrit*, 2015). Gerrit is a "web based code review and repository management for the Git version control system" (*Gerrit Code Review*, n.d.). If the result is not empty, the results are added to the search page as Wikitext at the bottom of the search results page.
In the course of time it is planned to filter the search results so as to display only items which are notable. Notable in this case would mean items that have at least one statement and a certain number of sitelinks.

8.6 Ordering of statement groups

To find a technical solution to this problem, there was a "Request for comment" page set up (*Requests for comment/Statement group ordering — MediaWiki*, 2015).

The collection of ordered property IDs is stored on one page in the *MediaWiki namespace*. This namespace "is used to hold system messages and other important content" (*Help:Namespaces — MediaWiki*, 2015). It can be edited only by administrators. This page is called `MediaWiki:Wikibase-SortedProperties`. Since the page name is set in the code, it is a constant variable.
The code for the ordering is not merged yet, since it's supposed to become part of Wikibase instead of being accessible only in the ArticlePlaceholder extension. That allows the possibility of reusing the code without having to build up a dependency from Wikibase to the extension.

For now, the page has to be on the local Wikipedia. If that's not the case, or if the page is empty or filled with non-text content, the corresponding exception is thrown. For that purpose the `ArticlePlaceholderException` was created, which extends PHP's `RuntimeException` to catch more specific exceptions and allow for more specific error handling.

In the class `PropertyOrderProvider` the page is parsed into a PHP array. In this array, the property IDs are the keys and the ordinal numbers the values.

To get the content of the page, the MediaWiki function `getNativeData()` is used. This page content is parsed in the function `parseList( $pageContent )`. To remove all comments in the HTML multiple line comment style, the following regular expression is used.

```
@<!--.*?-->@s
```

The `@` is the delimiter. All comments matching this pattern are replaced with empty strings.
After removing all comments, all Strings matching the following regular expression are written into an array.

```
@^\*\s*([Pp]\d+)@m
```

This regular expression matches all strings containing an asterisk, optional whitespace, and a upper or lower case *P* followed by a number. The array is flipped, to have the properties as keys. This is important when it comes to using the array in Lua. It is not used in the `mw.ext.articlePlaceholder.entityRenderer.lua` module yet, but will be in the future.
An instance of the class is created in the `ArticlePlaceholderServices`. In the class `Scribunto_LuaArticlePlaceholderLibrary` in the `ArticlePlaceholder\Lua` namespace with the help of `ArticlePlaceholderServices` the array with the property order is returned in the function `getPropertyOrder()`, which then can be called in the Lua module. This is explained in further detail in the following Chapter 8.7: Including PHP in Lua.
In `mw.ext.articlePlaceholder.entityRenderer.lua` the property IDs of the item will be ordered according to the ordinal numbers of the sorted array.
After the sorting is executed, it is quite easy to limit the number of statements, since less important statements can be identified by their ordinal number.

8.7 Including PHP in Lua

To pass the return values of PHP functions to Lua, Scribunto provided functionality is used.
The functions accessible by the Lua code are declared in the class `Scribunto_LuaArticlePlaceholderLibrary`. This class extends `Scribunto_LuaLibraryBase` to register the Lua interface.

To get the image property set in LocalSettings.php, the global variable $wgArticlePlaceholderImageProperty is wrapped in an array and returned in the function getImageProperty(). Lua functions can have multiple return values (Ierusalimschy, 2013), thus Scribunto needs an array to model that.

In Lua, these PHP callbacks are copied from mw_interface to the local variable php. From this variable the PHP functions in Scribunto_LuaArticle PlaceholderLibrary can be invoked. Therefore the image property ID can now be used in Lua as well, by invoking getImageProperty() in mw.ext.articleplaceholder.entityRenderer.

Getting the array with the ordered properties requires a service, the ArticlePlaceholderService.It returns an instance of itself. Its function getPropertyOrderProvider() returns an instance of the PropertyOrderProvider described in Chapter 8.6: Ordering of statement groups. This follows the *Singleton pattern*, which enforces, according to Bishop (2007), "that only one instance of a class will be created". In Scribunto_LuaArticle PlaceholderLibrary in the function Scribunto_LuaArticlePlaceholderLibrary this service is called and the associative array of ordered properties is returned.

Like the image property, the function can be invoked in the Lua module.

8.8 Internationalization

The extension includes an i18n folder to be able to localize all strings needed by the extension. This is the way predefined by MediaWiki (*Manual:Developing extensions — MediaWiki*, 2016). Every message needed by the extension has a unique message key. These keys are mapped to the text in the different languages. The key-value pairs are stored as JSON.

The keys need to be unique not only in the ArticlePlaceholder extension but also must not conflict with MediaWiki's and its other extensions' keys. Therefore they start with articleplaceholder-. Due to MediaWiki conventions, they are all lower case and may not contain spaces.

The documentation for every message is stored in the qqq.json file. The other messages are in the JSON files with the appropriate language key. While developing, it is only necessary to define the English and documentation messages, since the other messages are translated by the community on translatewiki.net.

> *translatewiki.net — Wikipedia, The Free Encyclopedia* (2015) states "translatewiki.net is a web-based translation platform, powered by the Translate extension for MediaWiki [...] [It has] 6000 translators for over 50[,000] pages from over [twenty] projects

> including MediaWiki, OpenStreetMap, Mifos, Encyclopedia of Life and MantisBT."

To load the messages, their directory is added to `$wgMessagesDirs`. `$wgMessagesDirs` is an associative array that maps the extension name to the appropriate message directory for the MediaWiki software to extract the messages and their keys.
The messages can now be used in PHP, JavaScript and Lua with their respective methods provided by MediaWiki. In PHP the message can be loaded using the function `wfMessage` with the message key as a parameter. The syntax for Lua is `mw.message.new( message-key ):plain()` and for JavaScript `mw.message( message-key ).escaped()`. Due to the similar syntax of Lua and JavaScript, the function call is similar, too.
In the last two examples, the messages are additionally escaped.

8.9 Unit tests

MediaWiki, and therefore Wikibase too, use *PHPUnit* for automated testing. PHPUnit is a test framework for PHP.
The tests for each class are named according to the class names with the suffix *Test*.

> "When we are writing a test in which we cannot (or choose not to) use a real depended-on component (DOC), we can replace it with a *Test Double*. The *Test Double* doesn't have to behave exactly like the real DOC; it merely has to provide the same API as the real one so that the SUT [system under test] *thinks* it is the real one" states Meszaros (2007).

PHPUnit provides the option to create so called *mocks* as test doubles. In the test class `SearchHookHandlerTest`, for example, the SpecialPage `SpecialSearch` is mocked to work on a test double.
PHPUnit allows using data providers. Those data providers return an array of arrays, which provide data for a test method. This way, one method can test several different cases with multiple datasets. The arrays can contain a message explaining what the test is about, a parameter passed to the tested method, and the expected value returned.
Each test method can contain one or more assertion. Those are able to test whether the result of the tested function equals the expected value.

8.10 Deployment

Adding new components to a software package requires, according to Carzaniga et al. (1998), "release, installation, activation, deactivation, update, and removal of components. These activities constitute a large and complex process that we refer to as software deployment."

To deploy the ArticlePlaceholder extension to Wikipedia, coordination with Wikimedia staff and volunteers is needed.
In Wikimedia, people with appropriate rights, volunteers and staff, can deploy. They have a schedule, which can be adjusted to an extension author's needs. Since the author does not have deployment rights on the Wikimedia cluster, cooperation with Wikimedia staff was necessary in order to perform the steps of Figure 8.7, which describes the deployment process.

Test system
From the beginning there was a test setup available on Wikimedia Labs[3]. Wikimedia Labs is "the Wikimedia Foundation (WMF)'s cloud computing environment for developing software for the Foundation's operations. It also hosts bots and tools maintained and used by the community to maintain the [F]oundation's projects" (*Wikipedia:Wikimedia Labs — Wikipedia, The Free Encyclopedia*, 2015). Besides the bots and scripts running on it, Wikimedia Labs is used for various tools and to test software by both volunteers and WMF staff.

The test setup was established to give editors and the public a first insight into what the extension looks like, what features it offers, and to let them participate in the development. It strives to start a discussion on what can be improved in order for it to be adjusted to the community's needs.

Security review
To deploy an extension to the Wikimedia cluster, it is necessary to pass a security review by a Security Engineer of the Wikimedia Foundation.

> "This review will ensure that controls identified in the design review have been implemented. Additionally, the team will attempt to ensure that the project avoids common implementation flaws" (*Wikimedia Security Team/Security reviews – MediaWiki*, 2016).

[3] `articleplaceholder.wmflabs.org/mediawiki`

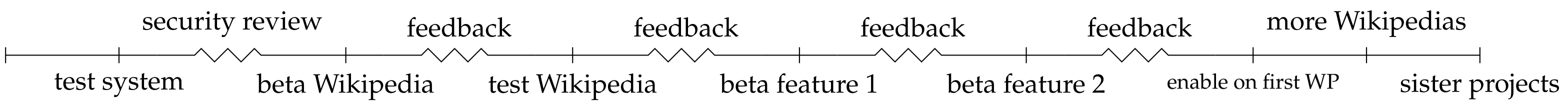


FIGURE 8.7: Time line of ArticlePlaceholder deployment

Beta Cluster
The Beta Cluster[4] is the *WMF integration environment* and used for testing. To deploy the extension there, someone with the appropriate deployment rights needed to do code review and then deploy the extension. First issues with a real Wikipedia cluster were identified and adjusted in this step.
Meanwhile the communities of all Wikipedias were given the opportunity to be among the first Wikipedias to employ the extension. Interested Wikipedia communities were asked to start discussion pages to gather the support of the community. (For the email sent to the to the *Wikipedia Technical Ambassadors* mailing list, and an extract of the following discussion on the community survey pages see Appendix B: Community communication.)

Test Wikipedia
Test Wikipedia[5] is a test setup and part of the *production cluster* by Wikimedia. It is connected to Test Wikidata[6]. From a development perspective, this is the last step of deployment before enabling the extension on a real Wikipedia. Herein lies another chance to gain more publicity and collect input before the Beta Feature stage. Issues can be found and fixed before they lead to problems on a Wikipedia that is used by a community to do their daily work.

The previous deployments were finalized by the time this thesis was written. The following steps will need more adjustment, communication and preparation.

Beta Feature
Beta Features are experimental functionalities that can be enabled by registered users in their preferences.

> "The primary purpose of Beta Features is to allow for Wikimedia designers and engineers (from the Wikimedia Foundation and community alike) to roll out technical improvements in an environment where large numbers of users can test, give feedback, and use these features in real-world settings." (*Beta Features — MediaWiki*, 2015)

The next step will be to deploy the extension as a beta feature. While having certain requirements for the test setup, the beta feature is supposed to be actually used by the community and therefore will have to fulfill more requirements, building upon what was already achieved before.
After the first deployment as a beta feature, the feedback from the community

[4] `http://en.wikipedia.beta.wmflabs.org/wiki/Main_Page`
[5] `https://test.wikipedia.org/wiki/Main_Page`
[6] `https://test.wikidata.org/wiki/Wikidata:Main_Page`

needs to be evaluated and the extension needs to be adjusted accordingly. The adjusted beta feature needs not only to consider the feedback again but also to gain the support and trust of the community to be enabled on a Wikipedia by default.

Deployment to a Wikipedia, enable by default

To have it actually deployed, the extension must implement the feedback and concerns collected earlier. Ideally it would be deployed at first on a Wikipedia with a limited number of articles and editors, so as to see if it actually fulfills the needs of the readers it is aimed at, and is accepted by existing communities.

Deployment to more Wikipedias

It would then be possible to deploy it to any other Wikipedias that wanted to participate.

Deployment to sister projects

> “**Wikimedia sister projects** are all the publicly available wikis operated by the Wikimedia Foundation, including Wikipedia.” (*Wikipedia:Wikimedia sister projects — Wikipedia, The Free Encyclopedia*, 2016)

In the future, it would be great to adjust the extension so as to make it useful to other Wikimedia projects as well. This is beyond the scope of the thesis but something to keep in mind during development.

One possible forthcoming project could be Wikivoyage, due to its structural similarity to Wikipedia.

Chapter 9

Conclusion and outlook

The main goal of this thesis was to increase access to free and open knowledge by providing access to information in languages that are often not considered. Thanks to the implementation of the ArticlePlaceholder, it is now possible to generate content pages on Wikipedia for Wikidata's free and open data.
This is not only a theoretical improvement but also desired by the communities as evident in the input pages on the local Wikipedias.

To evaluate how useful the extension will be for the different Wikipedias, it will be necessary to observe how many meaningful articles will be created with its help and whether they will render bot created stubs superfluous. Possible further research could look into natural language processing and creating text from data.

The development of the ArticlePlaceholder is ongoing.
In the future, many of the features mentioned in this thesis, but not yet realized will be certainly implemented such as the option of content translation, starting the article creation with data from Wikidata, adjusting the extension for sister projects, and making the ArticlePlaceholder pages indexable. The extension offers a basis to realize this and already implements many of the features that may change the accessibility of under-resourced languages for the better.

Appendix A

Figures

A.1 Internet Content Languages

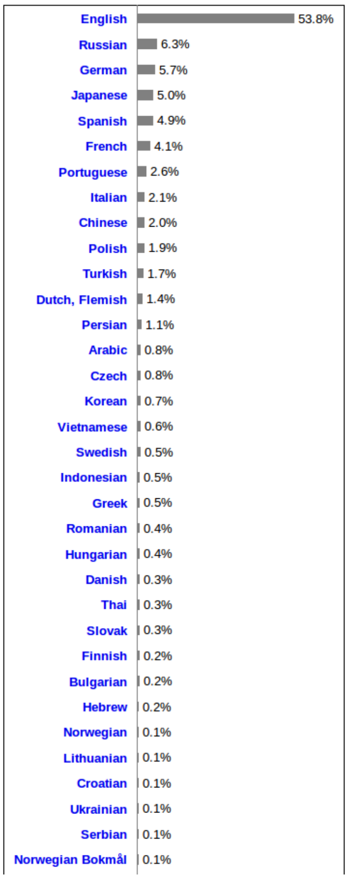


FIGURE A.1: Usage of content languages for websites [Screenshot] by W3Techs (2016b)

A.2 List of languages by number of native speakers

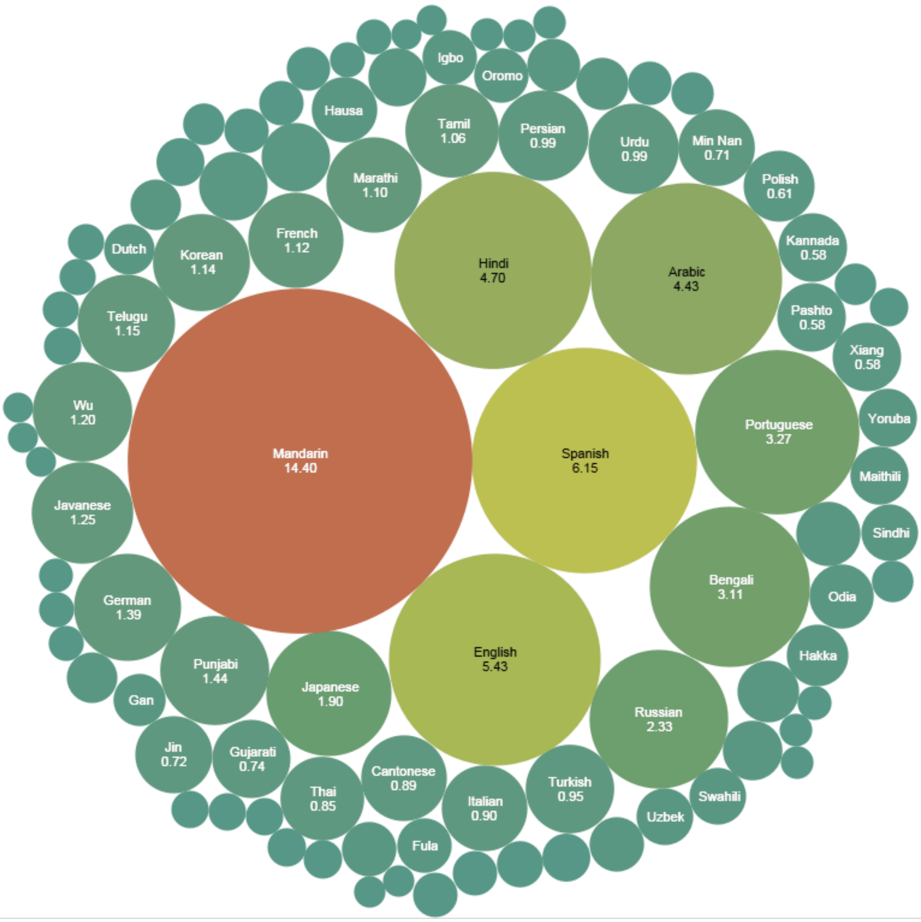
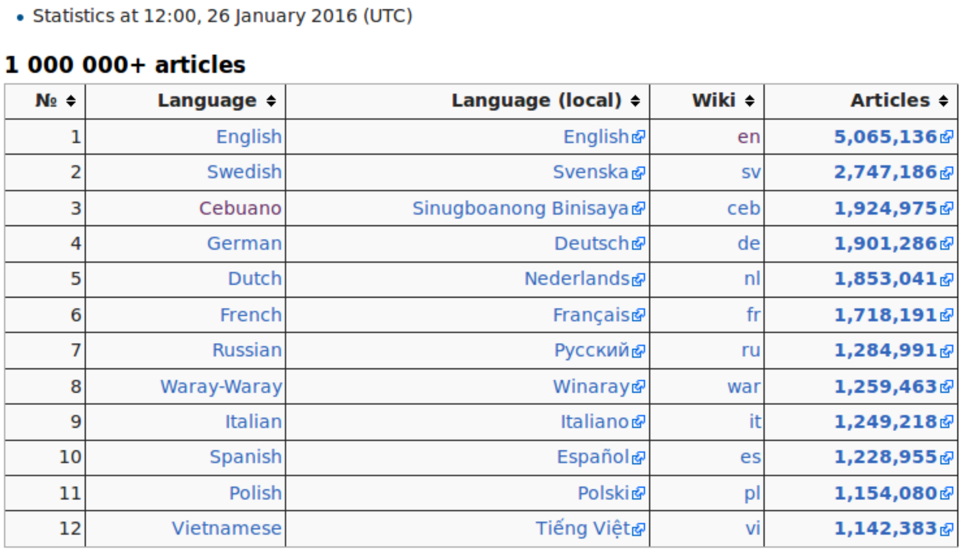


FIGURE A.2: List of languages by number of native speakers by Jroehl (2015)

A.3 Lists of Wikipedias with over 1.000.000 articles



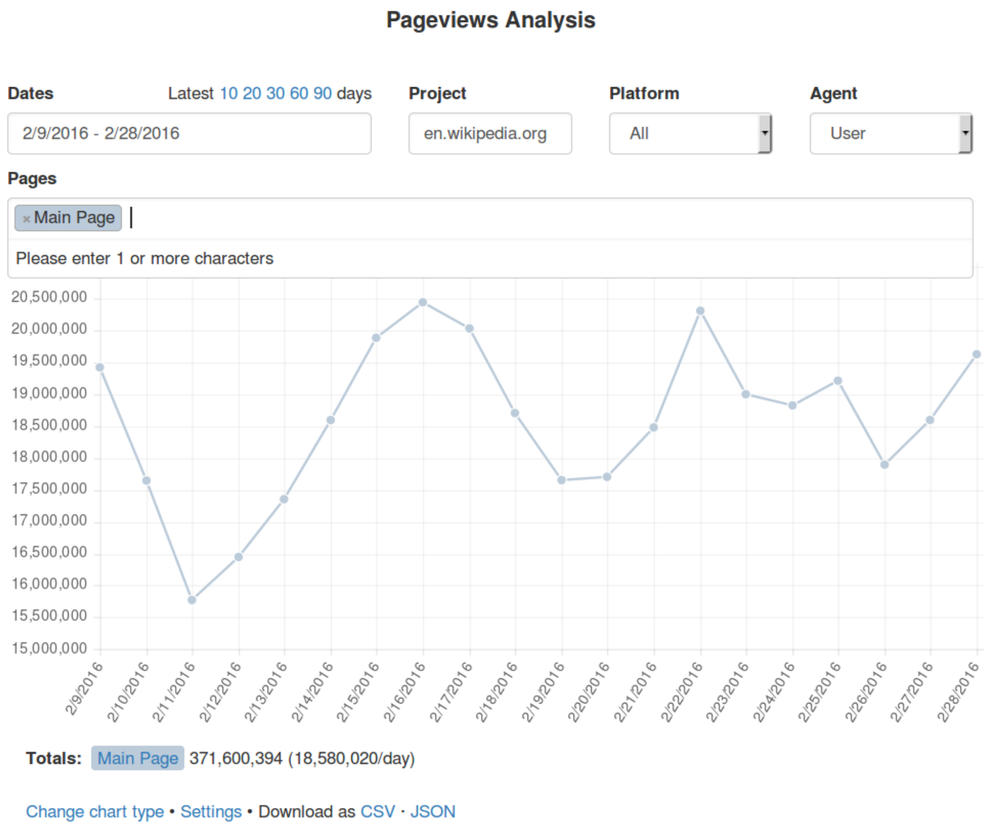
- Statistics at 12:00, 26 January 2016 (UTC)

1 000 000+ articles

№	Language	Language (local)	Wiki	Articles
1	English	English	en	**5,065,136**
2	Swedish	Svenska	sv	**2,747,186**
3	Cebuano	Sinugboanong Binisaya	ceb	**1,924,975**
4	German	Deutsch	de	**1,901,286**
5	Dutch	Nederlands	nl	**1,853,041**
6	French	Français	fr	**1,718,191**
7	Russian	Русский	ru	**1,284,991**
8	Waray-Waray	Winaray	war	**1,259,463**
9	Italian	Italiano	it	**1,249,218**
10	Spanish	Español	es	**1,228,955**
11	Polish	Polski	pl	**1,154,080**
12	Vietnamese	Tiếng Việt	vi	**1,142,383**

FIGURE A.3: *List of Wikipedias with over 1.000.000 articles [Screenshot]* (n.d.)

A.4 Pageviews Analysis



Pageviews Analysis

Dates: 2/9/2016 - 2/28/2016 (Latest 10 20 30 60 90 days)

Project: en.wikipedia.org

Platform: All

Agent: User

Pages: × Main Page

Please enter 1 or more characters

Totals: Main Page 371,600,394 (18,580,020/day)

FIGURE A.4: Pageviews Analysis [Screenshot] by MusikAnimal et al. (2016)

Appendix B

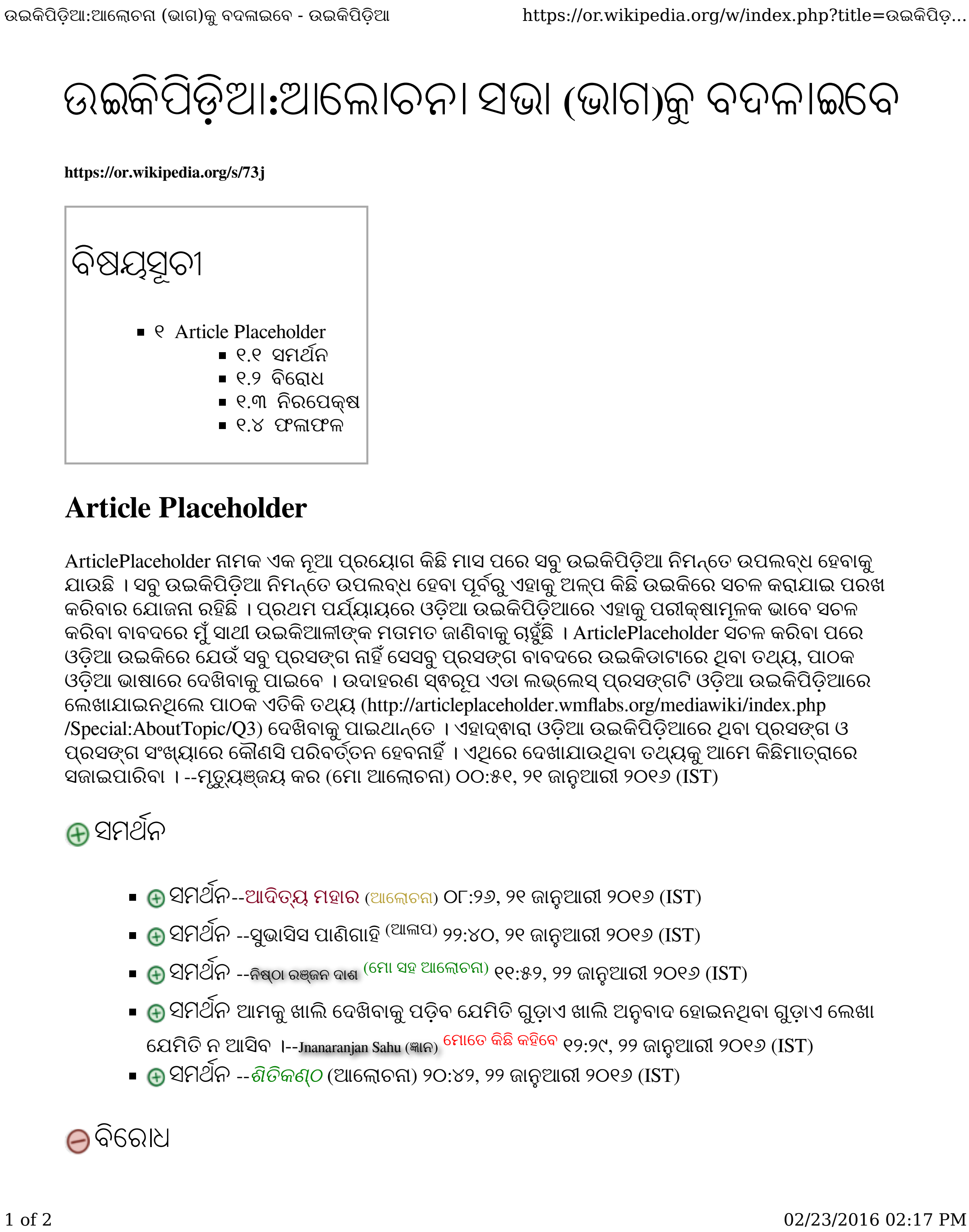
Community communication

[Wikitech-ambassadors] Looking for small Wikipedias to test a new feature for them

> As part of my Bachelor's thesis I worked on an extension called "ArticlePlaceholder" `https://www.mediawiki.org/wiki/Extension:ArticlePlaceholder` over the last months.
>
> One of the biggest barriers for accessing the knowledge Wikipedia provides is language.
>
> There are many topics that are only covered in few, big Wikipedias. People who don't speak any of these languages don't have access to all the information available potentially vital to them.
>
> The Article Placeholder extensions aims at smaller Wikipedias to support them in increasing access to data available on Wikidata. Article Placeholders are automatically generated content pages in Wikipedia or other mediawiki projects displaying data from Wikidata. They are clearly not actual articles but an overview of data on a topic which does not have an article yet. The design of the page and its content is under the control of the local community via Lua and templates but we will provide defaults so smaller Wikipedias can work with them without having to worry about the technical side of it.
>
> I have a test setup on Labs with an example for Ada Lovelace `http://articleplaceholder.wmflabs.org/mediawiki/index.php/Special:AboutTopic/Q3`
>
> The reader can find these pages by searching for a topic and gets results if there is an Item on Wikidata with the respective label and/or alias.

> The reader would benefit a lot since even if there is no article on a topic yet, they will still have basic information provided in their language. But it also might increase the numbers of editors due to increased usefulness of that Wikipedia.
>
> We are now looking for the first Wikipedias to support the extension by deploying it and giving their input. I am still developing the extension and the first Wikipedias to try it will naturally have a larger say in how it evolves.
>
> If your Wikipedia would like to give it a try please let me know. We would start it as a beta feature.
>
> Thank you,
>
> Lucie (Frimelle)

(Kaffee, 2016)



ଉଇକିପିଡ଼ିଆ:ଆଲୋଚନା (ଭାଗ)କୁ ବଦଳାଇବେ - ଉଇକିପିଡ଼ିଆ https://or.wikipedia.org/w/index.php?title=ଉଇକିପିଡ଼...

ଉଇକିପିଡ଼ିଆ:ଆଲୋଚନା ସଭା (ଭାଗ)କୁ ବଦଳାଇବେ

https://or.wikipedia.org/s/73j

ବିଷୟସୂଚୀ

Article Placeholder

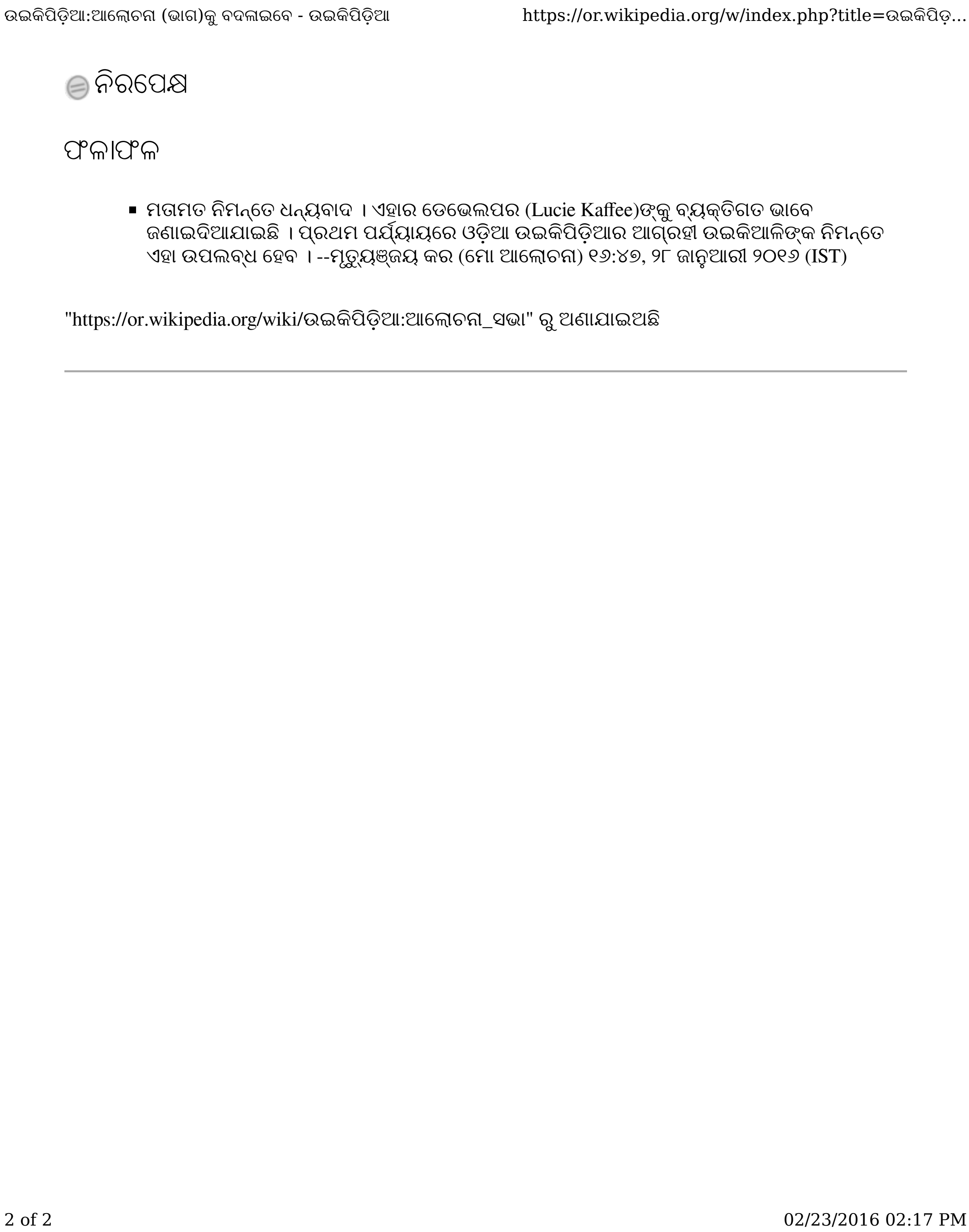
ArticlePlaceholder ନାମକ ଏକ ନୂଆ ପ୍ରୟୋଗ କିଛି ମାସ ପରେ ସବୁ ଉଇକିପିଡ଼ିଆ ନିମନ୍ତେ ଉପଲବ୍ଧ ହେବାକୁ ଯାଉଛି । ସବୁ ଉଇକିପିଡ଼ିଆ ନିମନ୍ତେ ଉପଲବ୍ଧ ହେବା ପୂର୍ବରୁ ଏହାକୁ ଅଳ୍ପ କିଛି ଉଇକିରେ ସଚଳ କରାଯାଇ ପରଖ କରିବାର ଯୋଜନା ରହିଛି । ପ୍ରଥମ ପର୍ଯ୍ୟାୟରେ ଓଡ଼ିଆ ଉଇକିପିଡ଼ିଆରେ ଏହାକୁ ପରୀକ୍ଷାମୂଳକ ଭାବେ ସଚଳ କରିବା ବାବଦରେ ମୁଁ ସାଥୀ ଉଇକିଆଳୀଙ୍କ ମତାମତ ଜାଣିବାକୁ ଚାହୁଁଛି । ArticlePlaceholder ସଚଳ କରିବା ପରେ ଓଡ଼ିଆ ଉଇକିରେ ଯେଉଁ ସବୁ ପ୍ରସଙ୍ଗ ନାହିଁ ସେସବୁ ପ୍ରସଙ୍ଗ ବାବଦରେ ଉଇକିଡାଟାରେ ଥିବା ତଥ୍ୟ, ପାଠକ ଓଡ଼ିଆ ଭାଷାରେ ଦେଖିବାକୁ ପାଇବେ । ଉଦାହରଣ ସ୍ୱରୂପ ଏଡା ଲଭ୍‌ଲେସ୍ ପ୍ରସଙ୍ଗଟି ଓଡ଼ିଆ ଉଇକିପିଡ଼ିଆରେ ଲେଖାଯାଇନଥିଲେ ପାଠକ ଏତିକି ତଥ୍ୟ (http://articleplaceholder.wmflabs.org/mediawiki/index.php/Special:AboutTopic/Q3) ଦେଖିବାକୁ ପାଇଥାନ୍ତେ । ଏହାଦ୍ୱାରା ଓଡ଼ିଆ ଉଇକିପିଡ଼ିଆରେ ଥିବା ପ୍ରସଙ୍ଗ ଓ ପ୍ରସଙ୍ଗ ସଂଖ୍ୟାରେ କୌଣସି ପରିବର୍ତ୍ତନ ହେବନାହିଁ । ଏଥିରେ ଦେଖାଯାଉଥିବା ତଥ୍ୟକୁ ଆମେ କିଛିମାତ୍ରାରେ ସଜାଇପାରିବା । --ମୃତ୍ୟୁଞ୍ଜୟ କର (ମୋ ଆଲୋଚନା) ୦୦:୫୧, ୨୧ ଜାନୁଆରୀ ୨୦୧୬ (IST)

⊕ ସମର୍ଥନ

- ⊕ ସମର୍ଥନ--ଆଦିତ୍ୟ ମହାର (ଆଲୋଚନା) ୦୮:୨୬, ୨୧ ଜାନୁଆରୀ ୨୦୧୬ (IST)
- ⊕ ସମର୍ଥନ --ସୁଭାସିସ ପାଣିଗାହି (ଆଳାପ) ୨୨:୪୦, ୨୧ ଜାନୁଆରୀ ୨୦୧୬ (IST)
- ⊕ ସମର୍ଥନ --ନିଷ୍ଠା ରଞ୍ଜନ ଦାଶ (ମୋ ସହ ଆଲୋଚନା) ୧୧:୫୨, ୨୨ ଜାନୁଆରୀ ୨୦୧୬ (IST)
- ⊕ ସମର୍ଥନ ଆମକୁ ଖାଲି ଦେଖିବାକୁ ପଡ଼ିବ ଯେମିତି ଗୁଡ଼ାଏ ଖାଲି ଅନୁବାଦ ହୋଇନଥିବା ଗୁଡ଼ାଏ ଲେଖା ଯେମିତି ନ ଆସିବ ।--Jnanaranjan Sahu (ଜ୍ଞାନ) ମୋତେ କିଛି କହିବେ ୧୨:୨୯, ୨୨ ଜାନୁଆରୀ ୨୦୧୬ (IST)
- ⊕ ସମର୍ଥନ --*ଶିତିକଣ୍ଠ* (ଆଲୋଚନା) ୨୦:୪୨, ୨୨ ଜାନୁଆରୀ ୨୦୧୬ (IST)

⊖ ବିରୋଧ

1 of 2 02/23/2016 02:17 PM



ଉଇକିପିଡ଼ିଆ:ଆଲୋଚନା (ଭାଗ)କୁ ବଦଳାଇବେ - ଉଇକିପିଡ଼ିଆ https://or.wikipedia.org/w/index.php?title=ଉଇକିପିଡ଼...

ନିରପେକ୍ଷ

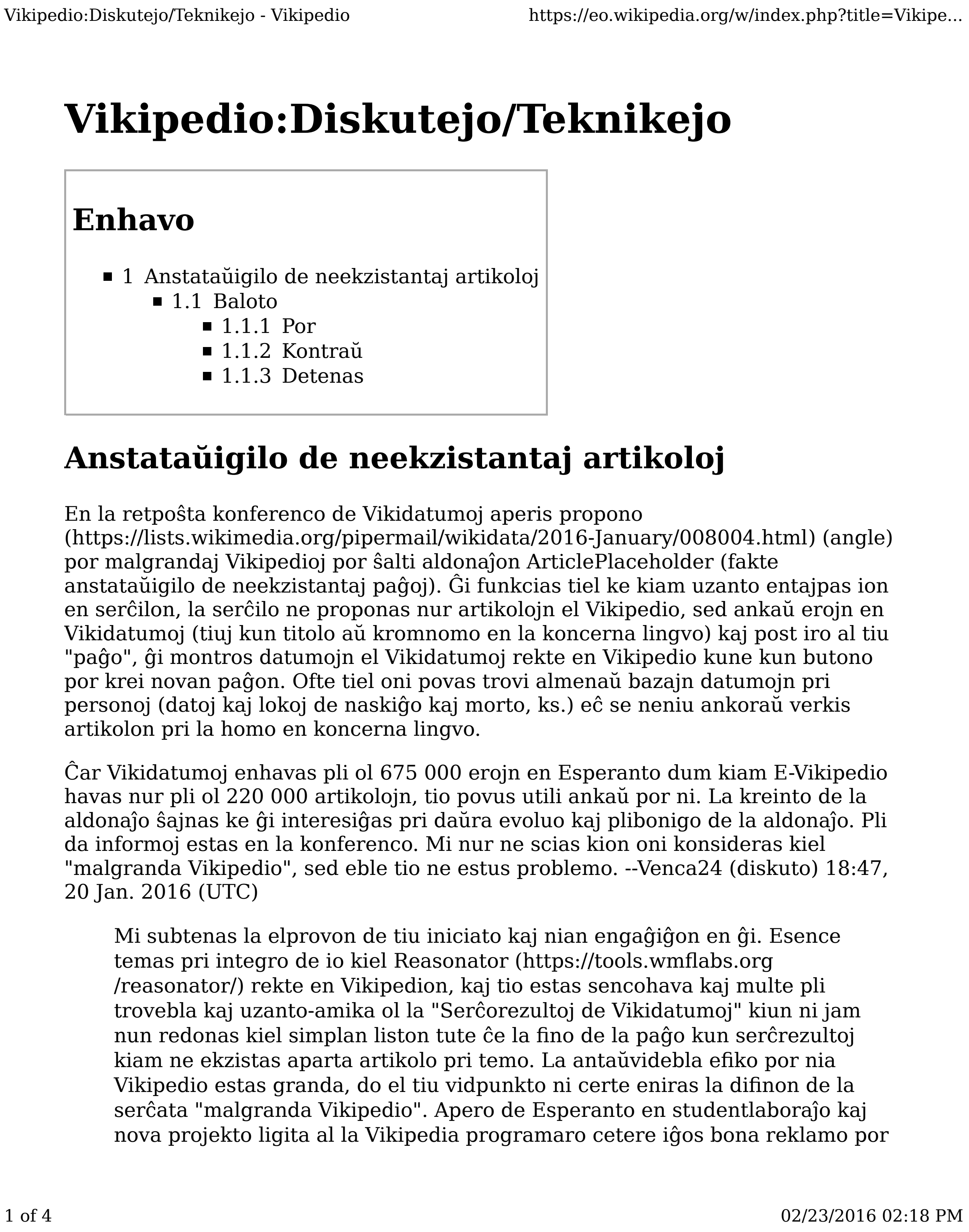
ଫଳାଫଳ

- ମତମତ ନିମନ୍ତେ ଧନ୍ୟବାଦ । ଏହାର ଡେଭେଲପର (Lucie Kaffee)ଙ୍କୁ ବ୍ୟକ୍ତିଗତ ଭାବେ ଜଣାଇଦିଆଯାଇଛି । ପ୍ରଥମ ପର୍ଯ୍ୟାୟରେ ଓଡ଼ିଆ ଉଇକିପିଡ଼ିଆର ଆଗ୍ରହୀ ଉଇକିଆଳିଙ୍କ ନିମନ୍ତେ ଏହା ଉପଲବ୍ଧ ହେବ । --ମୃତ୍ୟୁଞ୍ଜୟ କର (ମୋ ଆଲୋଚନା) ୧୬:୪୭, ୨୮ ଜାନୁଆରୀ ୨୦୧୬ (IST)

"https://or.wikipedia.org/wiki/ଉଇକିପିଡ଼ିଆ:ଆଲୋଚନା_ସଭା" ରୁ ଅଣାଯାଇଅଛି

2 of 2 02/23/2016 02:17 PM

(*Wikipedia:Village Pump – Wikipedia, The Free Encyclopedia*, n.d.)



Vikipedio:Diskutejo/Teknikejo - Vikipedio https://eo.wikipedia.org/w/index.php?title=Vikipe...

Vikipedio:Diskutejo/Teknikejo

Enhavo

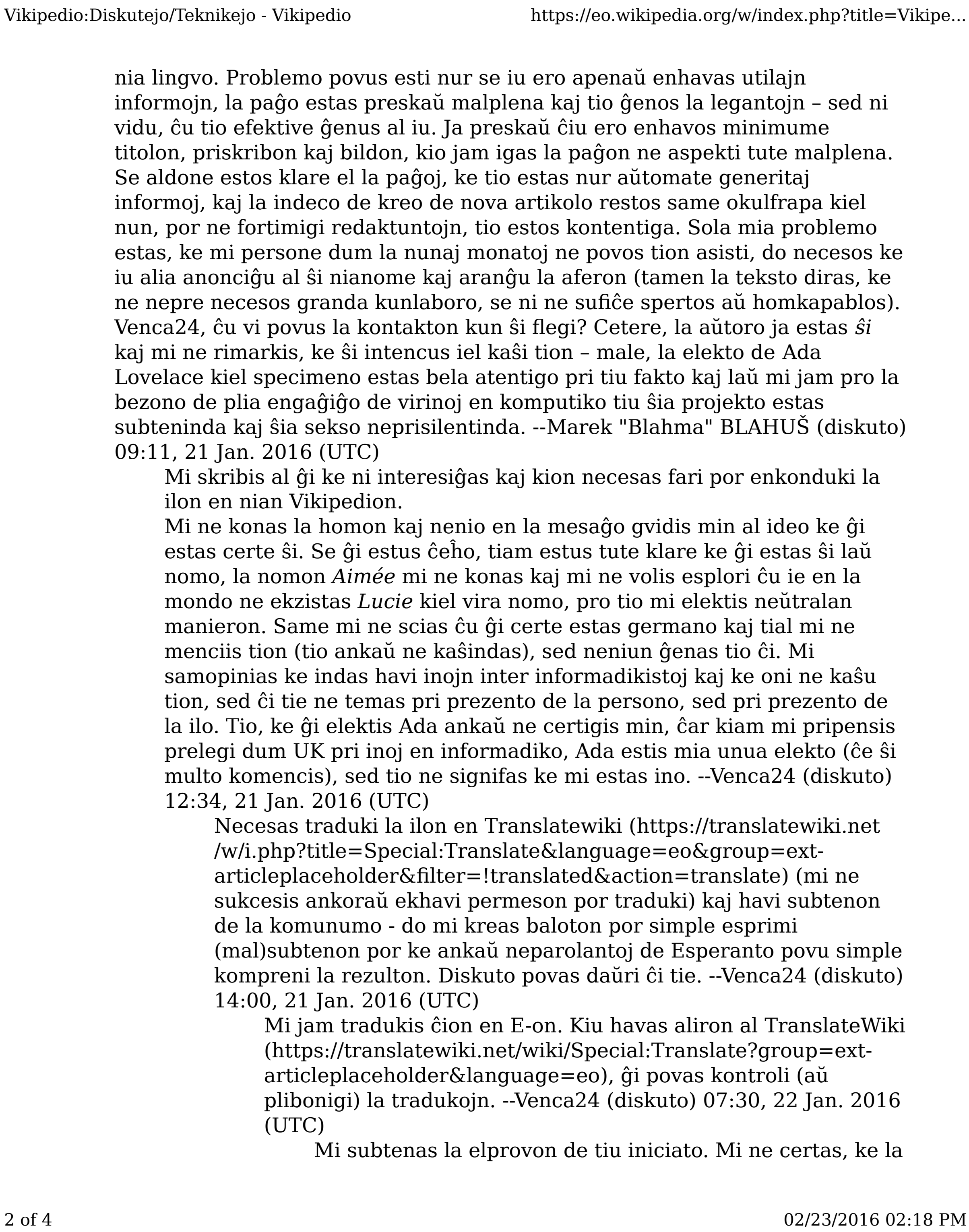
Anstataŭigilo de neekzistantaj artikoloj

En la retpoŝta konferenco de Vikidatumoj aperis propono (https://lists.wikimedia.org/pipermail/wikidata/2016-January/008004.html) (angle) por malgrandaj Vikipedioj por ŝalti aldonaĵon ArticlePlaceholder (fakte anstataŭigilo de neekzistantaj paĝoj). Ĝi funkcias tiel ke kiam uzanto entajpas ion en serĉilon, la serĉilo ne proponas nur artikolojn el Vikipedio, sed ankaŭ erojn en Vikidatumoj (tiuj kun titolo aŭ kromnomo en la koncerna lingvo) kaj post iro al tiu "paĝo", ĝi montros datumojn el Vikidatumoj rekte en Vikipedio kune kun butono por krei novan paĝon. Ofte tiel oni povas trovi almenaŭ bazajn datumojn pri personoj (datoj kaj lokoj de naskiĝo kaj morto, ks.) eĉ se neniu ankoraŭ verkis artikolon pri la homo en koncerna lingvo.

Ĉar Vikidatumoj enhavas pli ol 675 000 erojn en Esperanto dum kiam E-Vikipedio havas nur pli ol 220 000 artikolojn, tio povus utili ankaŭ por ni. La kreinto de la aldonaĵo ŝajnas ke ĝi interesiĝas pri daŭra evoluo kaj plibonigo de la aldonaĵo. Pli da informoj estas en la konferenco. Mi nur ne scias kion oni konsideras kiel "malgranda Vikipedio", sed eble tio ne estus problemo. --Venca24 (diskuto) 18:47, 20 Jan. 2016 (UTC)

> Mi subtenas la elprovon de tiu iniciato kaj nian engaĝiĝon en ĝi. Esence temas pri integro de io kiel Reasonator (https://tools.wmflabs.org/reasonator/) rekte en Vikipedion, kaj tio estas sencohava kaj multe pli trovebla kaj uzanto-amika ol la "Serĉorezultoj de Vikidatumoj" kiun ni jam nun redonas kiel simplan liston tute ĉe la fino de la paĝo kun serĉrezultoj kiam ne ekzistas aparta artikolo pri temo. La antaŭvidebla efiko por nia Vikipedio estas granda, do el tiu vidpunkto ni certe eniras la difinon de la serĉata "malgranda Vikipedio". Apero de Esperanto en studentlaboraĵo kaj nova projekto ligita al la Vikipedia programaro cetere iĝos bona reklamo por

1 of 4 02/23/2016 02:18 PM



Vikipedio:Diskutejo/Teknikejo - Vikipedio https://eo.wikipedia.org/w/index.php?title=Vikipe...

nia lingvo. Problemo povus esti nur se iu ero apenaŭ enhavas utilajn informojn, la paĝo estas preskaŭ malplena kaj tio ĝenos la legantojn – sed ni vidu, ĉu tio efektive ĝenus al iu. Ja preskaŭ ĉiu ero enhavos minimume titolon, priskribon kaj bildon, kio jam igas la paĝon ne aspekti tute malplena. Se aldone estos klare el la paĝoj, ke tio estas nur aŭtomate generitaj informoj, kaj la indeco de kreo de nova artikolo restos same okulfrapa kiel nun, por ne fortimigi redaktuntojn, tio estos kontentiga. Sola mia problemo estas, ke mi persone dum la nunaj monatoj ne povos tion asisti, do necesos ke iu alia anonciĝu al ŝi nianome kaj aranĝu la aferon (tamen la teksto diras, ke ne nepre necesos granda kunlaboro, se ni ne sufiĉe spertos aŭ homkapablos). Venca24, ĉu vi povus la kontakton kun ŝi flegi? Cetere, la aŭtoro ja estas *ŝi* kaj mi ne rimarkis, ke ŝi intencus iel kaŝi tion – male, la elekto de Ada Lovelace kiel specimeno estas bela atentigo pri tiu fakto kaj laŭ mi jam pro la bezono de plia engaĝiĝo de virinoj en komputiko tiu ŝia projekto estas subteninda kaj ŝia sekso neprisilentinda. --Marek "Blahma" BLAHUŠ (diskuto) 09:11, 21 Jan. 2016 (UTC)

Mi skribis al ĝi ke ni interesiĝas kaj kion necesas fari por enkonduki la ilon en nian Vikipedion.

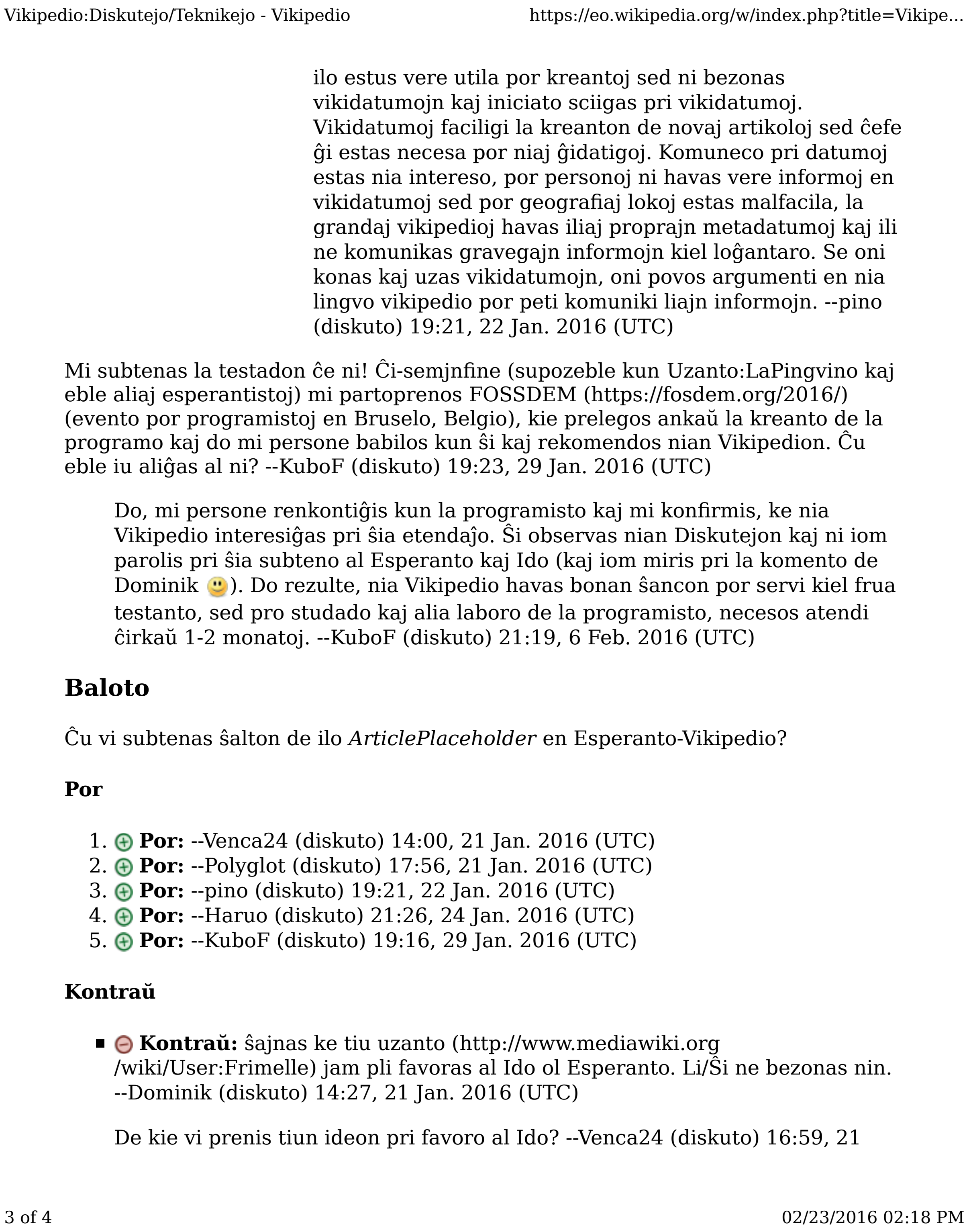
Mi ne konas la homon kaj nenio en la mesaĝo gvidis min al ideo ke ĝi estas certe ŝi. Se ĝi estus ĉeĥo, tiam estus tute klare ke ĝi estas ŝi laŭ nomo, la nomon *Aimée* mi ne konas kaj mi ne volis esplori ĉu ie en la mondo ne ekzistas *Lucie* kiel vira nomo, pro tio mi elektis neŭtralan manieron. Same mi ne scias ĉu ĝi certe estas germano kaj tial mi ne menciis tion (tio ankaŭ ne kaŝindas), sed neniun ĝenas tio ĉi. Mi samopinias ke indas havi inojn inter informadikistoj kaj ke oni ne kaŝu tion, sed ĉi tie ne temas pri prezento de la persono, sed pri prezento de la ilo. Tio, ke ĝi elektis Ada ankaŭ ne certigis min, ĉar kiam mi pripensis prelegi dum UK pri inoj en informadiko, Ada estis mia unua elekto (ĉe ŝi multo komencis), sed tio ne signifas ke mi estas ino. --Venca24 (diskuto) 12:34, 21 Jan. 2016 (UTC)

Necesas traduki la ilon en Translatewiki (https://translatewiki.net/w/i.php?title=Special:Translate&language=eo&group=ext-articleplaceholder&filter=!translated&action=translate) (mi ne sukcesis ankoraŭ ekhavi permeson por traduki) kaj havi subtenon de la komunumo - do mi kreas baloton por simple esprimi (mal)subtenon por ke ankaŭ neparolantoj de Esperanto povu simple kompreni la rezulton. Diskuto povas daŭri ĉi tie. --Venca24 (diskuto) 14:00, 21 Jan. 2016 (UTC)

Mi jam tradukis ĉion en E-on. Kiu havas aliron al TranslateWiki (https://translatewiki.net/wiki/Special:Translate?group=ext-articleplaceholder&language=eo), ĝi povas kontroli (aŭ plibonigi) la tradukojn. --Venca24 (diskuto) 07:30, 22 Jan. 2016 (UTC)

Mi subtenas la elprovon de tiu iniciato. Mi ne certas, ke la

2 of 4 02/23/2016 02:18 PM



Vikipedio:Diskutejo/Teknikejo - Vikipedio https://eo.wikipedia.org/w/index.php?title=Vikipe...

> ilo estus vere utila por kreantoj sed ni bezonas vikidatumojn kaj iniciato sciigas pri vikidatumoj. Vikidatumoj faciligi la kreanton de novaj artikoloj sed ĉefe ĝi estas necesa por niaj ĝidatigoj. Komuneco pri datumoj estas nia intereso, por personoj ni havas vere informoj en vikidatumoj sed por geografiaj lokoj estas malfacila, la grandaj vikipedioj havas iliaj proprajn metadatumoj kaj ili ne komunikas gravegajn informojn kiel loĝantaro. Se oni konas kaj uzas vikidatumojn, oni povos argumenti en nia lingvo vikipedio por peti komuniki liajn informojn. --pino (diskuto) 19:21, 22 Jan. 2016 (UTC)

Mi subtenas la testadon ĉe ni! Ĉi-semjnfine (supozeble kun Uzanto:LaPingvino kaj eble aliaj esperantistoj) mi partoprenos FOSSDEM (https://fosdem.org/2016/) (evento por programistoj en Bruselo, Belgio), kie prelegos ankaŭ la kreanto de la programo kaj do mi persone babilos kun ŝi kaj rekomendos nian Vikipedion. Ĉu eble iu aliĝas al ni? --KuboF (diskuto) 19:23, 29 Jan. 2016 (UTC)

> Do, mi persone renkontiĝis kun la programisto kaj mi konfirmis, ke nia Vikipedio interesiĝas pri ŝia etendaĵo. Ŝi observas nian Diskutejon kaj ni iom parolis pri ŝia subteno al Esperanto kaj Ido (kaj iom miris pri la komento de Dominik 🙂). Do rezulte, nia Vikipedio havas bonan ŝancon por servi kiel frua testanto, sed pro studado kaj alia laboro de la programisto, necesos atendi ĉirkaŭ 1-2 monatoj. --KuboF (diskuto) 21:19, 6 Feb. 2016 (UTC)

Baloto

Ĉu vi subtenas ŝalton de ilo *ArticlePlaceholder* en Esperanto-Vikipedio?

Por

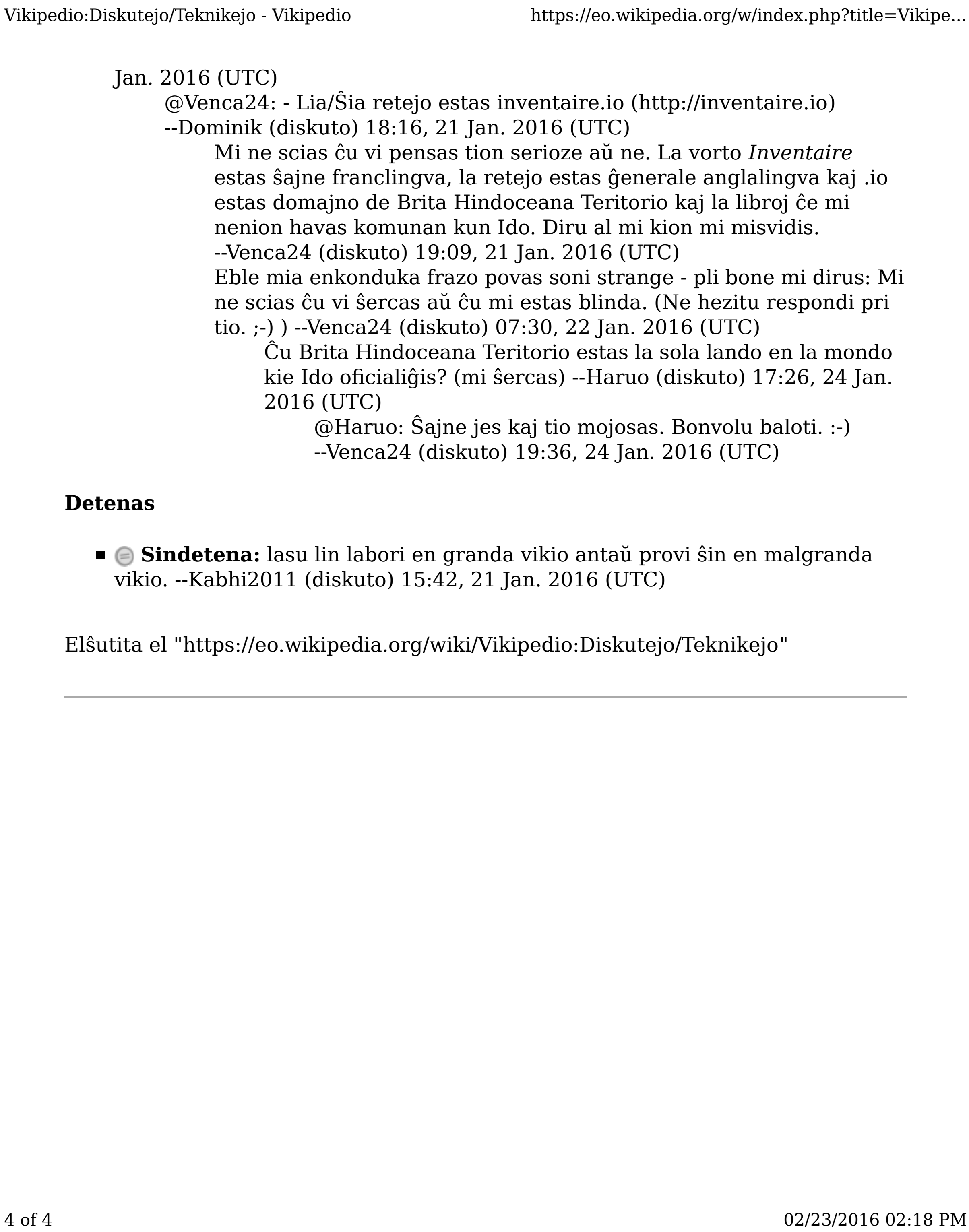
1. ⊕ **Por:** --Venca24 (diskuto) 14:00, 21 Jan. 2016 (UTC)
2. ⊕ **Por:** --Polyglot (diskuto) 17:56, 21 Jan. 2016 (UTC)
3. ⊕ **Por:** --pino (diskuto) 19:21, 22 Jan. 2016 (UTC)
4. ⊕ **Por:** --Haruo (diskuto) 21:26, 24 Jan. 2016 (UTC)
5. ⊕ **Por:** --KuboF (diskuto) 19:16, 29 Jan. 2016 (UTC)

Kontraŭ

- ⊖ **Kontraŭ:** ŝajnas ke tiu uzanto (http://www.mediawiki.org/wiki/User:Frimelle) jam pli favoras al Ido ol Esperanto. Li/Ŝi ne bezonas nin. --Dominik (diskuto) 14:27, 21 Jan. 2016 (UTC)

  De kie vi prenis tiun ideon pri favoro al Ido? --Venca24 (diskuto) 16:59, 21

3 of 4 02/23/2016 02:18 PM



Vikipedio:Diskutejo/Teknikejo - Vikipedio https://eo.wikipedia.org/w/index.php?title=Vikipe...

Jan. 2016 (UTC)

@Venca24: - Lia/Ŝia retejo estas inventaire.io (http://inventaire.io) --Dominik (diskuto) 18:16, 21 Jan. 2016 (UTC)

Mi ne scias ĉu vi pensas tion serioze aŭ ne. La vorto *Inventaire* estas ŝajne franclingva, la retejo estas ĝenerale anglalingva kaj .io estas domajno de Brita Hindoceana Teritorio kaj la libroj ĉe mi nenion havas komunan kun Ido. Diru al mi kion mi misvidis. --Venca24 (diskuto) 19:09, 21 Jan. 2016 (UTC)

Eble mia enkonduka frazo povas soni strange - pli bone mi dirus: Mi ne scias ĉu vi ŝercas aŭ ĉu mi estas blinda. (Ne hezitu respondi pri tio. ;-)) --Venca24 (diskuto) 07:30, 22 Jan. 2016 (UTC)

Ĉu Brita Hindoceana Teritorio estas la sola lando en la mondo kie Ido oficialiĝis? (mi ŝercas) --Haruo (diskuto) 17:26, 24 Jan. 2016 (UTC)

@Haruo: Ŝajne jes kaj tio mojosas. Bonvolu baloti. :-) --Venca24 (diskuto) 19:36, 24 Jan. 2016 (UTC)

Detenas

- **Sindetena:** lasu lin labori en granda vikio antaŭ provi ŝin en malgranda vikio. --Kabhi2011 (diskuto) 15:42, 21 Jan. 2016 (UTC)

Elŝutita el "https://eo.wikipedia.org/wiki/Vikipedio:Diskutejo/Teknikejo"

4 of 4 02/23/2016 02:18 PM

(*Vikipedio:Diskutejo/Teknikejo — Wikipedia, The Free Encyclopedia*, 2016)

Bibliography

Ager, S. (2016). *Odia.* Retrieved from http://www.omniglot.com/writing/oriya.htm (Online; accessed 15-February-2016)

Alexander, I. F., & Beus-Dukic, L. (2009). *Discovering Requirements: How to Specify Products and Services.* John Wiley & Sons.

Analytics/Reports/Clients without JavaScript — MediaWiki. (2015). Retrieved from https://www.mediawiki.org/w/index.php?title=Analytics/Reports/Clients_without_JavaScript&oldid=1526034 (Online; accessed 23-November-2015)

Beta Features — MediaWiki. (2015). Retrieved from https://www.mediawiki.org/w/index.php?title=Beta_Features&oldid=1967117 (Online; accessed 27-December-2015)

Bishop, J. (2007). *C# 3.0 Design Patterns.* O'Reilly Media, Inc.

Carzaniga, A., Fuggetta, A., Hall, R. S., Heimbigner, D., van der Hoek, A., & Wolf, A. L. (1998). *A Characterization Framework for Software Deployment Technologies* (Tech. Rep.). Boulder, CO 80309 USA: Department of Computer Science, University of Colorado.

Category:All extensions — MediaWiki. (2016). Retrieved from https://www.mediawiki.org/w/index.php?title=Category:All_extensions&oldid=2048016 (Online; accessed 15-February-2016)

Connextrastorycard. (n.d.). Retrieved from http://agilecoach.typepad.com/photos/connextra_user_story_2001/connextrastorycard.html (Online; accessed 21-February-2016)

creative commons. (n.d.). *CC0 1.0 Universal (cc0 1.0) Public Domain Dedication.* Retrieved from https://creativecommons.org/publicdomain/zero/1.0/ (Online; accessed 26-February-2016)

Encyclopædia Britannica. (n.d.). *Cebuano language.* Retrieved from http://www.britannica.com/topic/Cebuano-language (Online; accessed 14-February-2016)

Esselink, B. (n.d.). *The Evolution of Localization* (Tech. Rep.). Solution Architect, Lionbridge. Retrieved from http://www.intercultural.urv.cat/media/upload/domain_317/arxius/Technology/Esselink_Evolution.pdf (Online;

accessed 20-February-2016)

EVA Wiki — Semantic MediaWiki. (n.d.). Retrieved from `https://www.semantic-mediawiki.org/w/index.php?title=EVA_Wiki&oldid=40320` (Online; accessed 23-February-2016)

Extension:ArticlePlaceholder/Smart red links — MediaWiki. (2016). Retrieved from `https://www.mediawiki.org/w/index.php?title=Extension:ArticlePlaceholder/Smart_red_links&oldid=2004881` (Online; accessed 15-February-2016)

Extension:Scribunto — MediaWiki. (2016). Retrieved from `https://www.mediawiki.org/w/index.php?title=Extension:Scribunto&oldid=2049393` (Online; accessed 15-February-2016)

Extension:Wikibase Client — MediaWiki. (2015). Retrieved from `https://www.mediawiki.org/w/index.php?title=Extension:Wikibase_Client&oldid=1952656` (Online; accessed 15-February-2016)

Extension:WikibaseLib — MediaWiki. (2015). Retrieved from `https://www.mediawiki.org/w/index.php?title=Extension:WikibaseLib&oldid=1516751` (Online; accessed 16-February-2016)

Gerrit Code Review. (n.d.). Retrieved from `https://www.gerritcodereview.com/` (Online; accessed 28-February-2016)

Hay, S. (2013). *Responsive Design Workflow.* New Riders.

Help:Infobox — Wikipedia, The Free Encyclopedia. (2015). Retrieved from `https://en.wikipedia.org/w/index.php?title=Help:Infobox&oldid=693815565` (Online; accessed 28-December-2015)

Help:Interlanguage links – Wikipedia, The Free Encyclopedia. (2016). Retrieved from `https://en.wikipedia.org/w/index.php?title=Help:Interlanguage_links&oldid=705364400` (Online; accessed 3-March-2016)

Help:Namespaces — MediaWiki. (2015). Retrieved from `https://www.mediawiki.org/w/index.php?title=Help:Namespaces&oldid=1973970` (Online; accessed 15-February-2016)

Help:Special page — Wikipedia, The Free Encyclopedia. (2016). Retrieved from `https://en.wikipedia.org/w/index.php?title=Help:Special_page&oldid=697158833` (Online; accessed 14-February-2016)

Help:Template — Wikipedia, The Free Encyclopedia. (2016). Retrieved from `https://en.wikipedia.org/w/index.php?title=Help:`

Template&oldid=702835208 (Online; accessed 15-February-2016)

Ierusalimschy, R. (2013). *Programming in Lua* (Third ed.). Rio de Janeiro.

Jaggar, P. J. (2001). *Hausa* (Rev Upd ed.). John Benjamins Publishing.

Jemielniak, D. (2014). *Common Knowledge?: An Ethnography of Wikipedia*. Stanford University Press.

Jroehl. (2015). *List of languages by number of native speakers, packed bubbles.* Retrieved from https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers#/media/File:List_of_languages_by_number_of_native_speakers.png (Online; accessed 19-February-2016)

Kaffee, L.-A. (2016). *[wikitech-ambassadors] Looking for small Wikipedias to test a new feature for them [E-Mail].* Retrieved from https://www.mail-archive.com/wikitech-ambassadors@lists.wikimedia.org/msg01294.html (Online; accessed 20-February-2016)

KDE e.V. (n.d.). *KDE e.V. Homepage.* Retrieved from https://ev.kde.org/ (Online; accessed 26-February-2016)

Kritschmar, C. (2016). *File:WikidataTermDescriptorDiagram.png.* Retrieved from https://commons.wikimedia.org/w/index.php?title=File:WikidataTermDescriptorDiagram.png&oldid=185996484 (Online; accessed 21-February-2016)

Lange, D., Böhm, C., & Naumann, F. (2010). *Extracting Structured Information from Wikipedia Articles to Populate Infoboxes*. Universitätsverlag Potsdam.

Lidwell, W., Holden, K., & Butler, J. (2010). *Universal Principles of Design* (Rev Upd ed.). Rockport Publishers.

List of Wikipedias — Meta. (2016). Retrieved from https://meta.wikimedia.org/w/index.php?title=List_of_Wikipedias&oldid=15355062 (Online; accessed 19-February-2016)

List of wikipedias with over 1.000.000 articles [screenshot]. (n.d.). Retrieved from https://meta.wikimedia.org/wiki/List_of_Wikipedias (Online; accessed 19-February-2016)

Lsjbot — Wikipedia, The Free Encyclopedia. (2016). Retrieved from https://en.wikipedia.org/w/index.php?title=Lsjbot&oldid=701640506 (Online; accessed 14-February-2016)

Lua About. (2016). Retrieved from http://www.lua.org/about.html (Online; accessed 15-February-2016)

Manske, M. (2013). *Wikidata, or: Wikipedia, the other 60%.* Retrieved from http://magnusmanske.de/wordpress/?p=108 (Online; accessed 18-January-2016)

Manual:Configuration settings (alphabetical) — MediaWiki. (2016). Retrieved from https://www.mediawiki.org/w/index.php?title=Manual:Configuration_settings_%28alphabetical%29&oldid=2035951 (Online; accessed 15-February-2016)

Manual:Developing extensions — MediaWiki. (2016). Retrieved from https://www.mediawiki.org/w/index.php?title=Manual:Developing_extensions&oldid=2046456 (Online; accessed 20-February-2016)

Manual:Extension registration — MediaWiki. (2016). Retrieved from https://www.mediawiki.org/w/index.php?title=Manual:Extension_registration&oldid=2028274 (Online; accessed 15-February-2016)

Manual:Hooks/SpecialSearchResultsAppend — MediaWiki. (2015). Retrieved from https://www.mediawiki.org/w/index.php?title=Manual:Hooks/SpecialSearchResultsAppend&oldid=1779511 (Online; accessed 15-February-2016)

Manual:What is MediaWiki? — MediaWiki. (2013). Retrieved from https://www.mediawiki.org/w/index.php?title=Manual:What_is_MediaWiki%3F&oldid=743778 (Online; accessed 17-November-2015)

MediaWiki. (2015). Retrieved from https://www.mediawiki.org/w/index.php?title=MediaWiki&oldid=1446376 (Online; accessed 17-November-2015)

Meijssen, G. (2013). *Wikidata complements search in the Italian Wikipedia.* Retrieved from http://ultimategerardm.blogspot.de/2013/12/wikidata-complements-search-in-italian.html (Online; accessed 28-December-2015)

Meszaros, G. (2007). *xUnit Test Patterns: Refactoring Test Code.* Addison-Wesley.

Move the termindexsearchinteractor to lib — Gerrit. (2015). Retrieved from https://gerrit.wikimedia.org/r/#/c/243723/ (Online; accessed 15-February-2016)

MusikAnimal, Kaldari, & Forns, M. R. (2016). *Pageviews analysis [screenshot].* Retrieved from https://tools.wmflabs.org/pageviews/#start=2016-02-09&end=2016-02-28&project=en.wikipedia.org&platform=all-access&agent=user&pages=Main_Page (Online; accessed 29-February-2016)

Network to Promote Linguistic Diversity (NPLD). (n.d.). *Swedish.* Retrieved from http://www.npld.eu/about-us/swedish/ (Online; accessed 28-December-2015)

Oard, D. W., Doermann, D., Dorr, B., He, D., Resnik, P., Weinberg, A., . . .

Knight, K. (2003). Desparately Seeking Cebuano. *Proceedings of HLT-NAACL.*

Oojs ui — MediaWiki. (2015). Retrieved from `https://www.mediawiki.org/w/index.php?title=OOjs_UI&oldid=1919519` (Online; accessed 16-February-2016)

Oxford Dictionaries. (n.d.). *Wysiwyg.* Retrieved from `http://www.oxforddictionaries.com/definition/english/wysiwyg` (Online; accessed 18-February-2016)

Personas for product development – MediaWiki. (2016). Retrieved from `https://www.mediawiki.org/w/index.php?title=Personas_for_product_development&oldid=2055106` (Online; accessed 26-February-2016)

Php. (n.d.). Retrieved from `http://php.net/` (Online; accessed 15-February-2016)

Q1779 — Wikidata. (2016). Retrieved from `https://www.wikidata.org/w/index.php?title=Q1779&oldid=304854061` (Online; accessed 20-February-2016)

Q43631 — Wikidata. (2016). Retrieved from `https://www.wikidata.org/w/index.php?title=Q43631&oldid=299844501` (Online; accessed 15-February-2016)

Q5598 — Wikidata. (2016). Retrieved from `https://www.wikidata.org/w/index.php?title=Q5598&oldid=303384882` (Online; accessed 15-February-2016)

Reasonator — Meta. (2015). Retrieved from `https://meta.wikimedia.org/w/index.php?title=Reasonator&oldid=10813994` (Online; accessed 28-December-2015)

Requests for comment/Statement group ordering — MediaWiki. (2015). Retrieved from `https://www.mediawiki.org/w/index.php?title=Requests_for_comment/Statement_group_ordering&oldid=1936386` (Online; accessed 15-February-2016)

Schwab, K. (2015). The Internet Isn't Available in Most Languages. *The Atlantic.* Retrieved from `http://www.theatlantic.com/technology/archive/2015/11/the-internet-isnt-available-in-most-languages/417393/` (Online; accessed 17-February-2016)

Strong, C. (2013). *What Makes A Good User Story.* Retrieved from `http://www.strongandagile.co.uk/index.php/what-makes-a-good-user-story/` (Online; accessed 21-February-2016)

translatewiki.net — Wikipedia, The Free Encyclopedia. (2015). Retrieved from `https://en.wikipedia.org/w/index.php?title=Translatewiki.net&oldid=693306628` (Online; accessed

15-February-2016)

Vikipedio:diskutejo/teknikejo — Wikipedia, The Free Encyclopedia. (2016). Retrieved from https://eo.wikipedia.org/w/index.php?title=Vikipedio:Diskutejo/Teknikejo&oldid=5894252 (Online; accessed 23-February-2016)

W3Techs. (2016a). *Usage of content languages for websites.* Retrieved from http://w3techs.com/technologies/overview/content_language/all (Online; accessed 19-February-2016)

W3Techs. (2016b). *Usage of content languages for websites [screenshot].* Retrieved from http://w3techs.com/technologies/overview/content_language/all (Online; accessed 19-February-2016)

Wikibase. (2016). Retrieved from http://wikiba.se/ (Online; accessed 15-February-2016)

Wikidata Game. (n.d.). Retrieved from https://tools.wmflabs.org/wikidata-game/# (Online; accessed 20-February-2016)

Wikidata:Statistics — Wikidata. (2015). Retrieved from https://www.wikidata.org/w/index.php?title=Wikidata:Statistics/Wikipedia&oldid=236393460 (Online; accessed 10-January-2016)

Wikimedia Security Team/Security reviews – MediaWiki. (2016). Retrieved from https://www.mediawiki.org/w/index.php?title=Wikimedia_Security_Team/Security_reviews&oldid=2024601 (Online; accessed 28-February-2016)

Wikipedia:Administrators — Wikipedia, The Free Encyclopedia. (2016). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia:Administrators&oldid=702113925 (Online; accessed 14-February-2016)

Wikipedia:Bots — Wikipedia, The Free Encyclopedia. (2015). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia:Bots&oldid=662582073 (Online; accessed 18-June-2015)

Wikipedia:Five million articles — Wikipedia, The Free Encyclopedia. (2016). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia:Five_million_articles&oldid=704715513 (Online; accessed 19-February-2016)

Wikipedia:Lua — Wikipedia, The Free Encyclopedia. (2016). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia:Lua&oldid=699528865 (Online; accessed 15-February-2016)

Wikipedia:Notability — Wikipedia, The Free Encyclopedia. (2016). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia:Notability&oldid=703184391 (Online; accessed

14-February-2016)

Wikipedia:Red link — Wikipedia, The Free Encyclopedia. (2015). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia:Red_link&oldid=691209909 (Online; accessed 23-November-2015)

Wikipedia:Vandalism — Wikipedia, The Free Encyclopedia. (2016). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia:Vandalism&oldid=705478736 (Online; accessed 20-February-2016)

Wikipedia:village pump – Wikipedia, The Free Encyclopedia. (n.d.). Retrieved from https://or.wikipedia.org/w/index.php?title=%E0%AC%89%E0%AC%87%E0%AC%95%E0%AC%BF%E0%AC%AA%E0%AC%BF%E0%AC%A1%E0%AC%BC%E0%AC%BF%E0%AC%86:%E0%AC%86%E0%AC%B2%E0%AD%8B%E0%AC%9A%E0%AC%A8%E0%AC%BE_%E0%AC%B8%E0%AC%AD%E0%AC%BE&oldid=230153 (Online; accessed 23-February-2016)

Wikipedia:Wikimedia Labs — Wikipedia, The Free Encyclopedia. (2015). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia:Wikimedia_Labs&oldid=693539763 (Online; accessed 27-December-2015)

Wikipedia:Wikimedia sister projects — Wikipedia, The Free Encyclopedia. (2016). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia:Wikimedia_sister_projects&oldid=699790946 (Online; accessed 17-February-2016)

Wikipedia — Wikipedia, The Free Encyclopedia. (2016). Retrieved from https://en.wikipedia.org/w/index.php?title=Wikipedia&oldid=705741476 (Online; accessed 19-February-2016)