

# WikiBase, partendo da zero



**ItWikiCon2020** | 2020-10-25

# Perchè questa presentazione

- Buon pomeriggio e **benvenuti**, mi presento
- Sono qui per raccontarvi una **storia**
  - Sapevo mi servisse qualcosa come WikiData, ma non sapevo cosa / come / dove
  - Quando ho capito che non potevo usare WikiData per la nostra wiki, mi sono imbarcato in un lungo viaggio alla **scoperta** di WikiBase
  - Lungo il viaggio ho scoperto anche tante **altre** cose che qui vorrei raccontarvi

*«The access to Wikidata data is currently restricted to the Wikimedia projects, because of technical limitations. If you have your own instance of MediaWiki, you can't use Wikidata's data using these features. However, you can set up your own Wikibase instance and use data from there in the same way.»*

[https://www.wikidata.org/wiki/Wikidata:How\\_to\\_use\\_data\\_on\\_Wikimedia\\_projects](https://www.wikidata.org/wiki/Wikidata:How_to_use_data_on_Wikimedia_projects)

# Cosa è WikiBase

- È la estensione che muove WikiData
- È composta in effetti da *quattro* estensioni, le due esposte all'**utente** sono:
  - Client
  - Repository
- WikiBase è
  - Disperso
  - (forse) poco usato
  - poco conosciuto



# Cosa faremo

- In questa presentazione, faremo una **passeggiata** virtuale alla scoperta della configurazione di un ambiente WikiBase, appunto partendo da zero
  - Zero, o quasi. Perché Presupponiamo l'installazione di due istanze di MediaWiki
    - Due guide fatte bene: MediaWiki e DigitalOcean
- Seguiremo la **stessa** architettura usata da Wikimedia, ovvero avremo due installazioni di MediaWiki
  - Una wiki con Repository e Client
  - Una wiki solo Client
    - Le chiameremo "Server" e "Client" per **brevità**
  - Va bene così?

# Primi passi

- Si inizia scaricando il **pacchetto** di WikiBase corrispondente alla propria versione di MediaWiki
  - Non è compatibile con versioni datate di MW, ma è stabile dai test che ho fatto su 1.31, 1.34 e 1.35
- Si posiziona nella cartella delle **estensioni**
- Nelle prossime diapositive vedremo nel **dettaglio**
  - L'uso di *Composer*
  - La configurazione base

# Composer

- Composer è un package manager cosiddetto *Application-level*
  - Si occupa di gestire le librerie PHP all'interno di ogni istanza di MediaWiki
- **Fondamentale** per l'installazione di WikiBase
  - Ho imparato che è utilissimo **anche** per gestire altri pacchetti relativi a MediaWiki
  - Molte estensioni, per esempio, possono essere installate **automaticamente** in questo modo eliminando tanto lavoro manuale.
- Si sfrutta il file `composer.local.json`, il cui uso è spiegato qui [https://www.mediawiki.org/wiki/Composer/For\\_extensions#Specify\\_the\\_extensions\\_to\\_be\\_installed](https://www.mediawiki.org/wiki/Composer/For_extensions#Specify_the_extensions_to_be_installed)

# composer.local.json

- I pacchetti da installare vanno inseriti nel **file** `composer.local.json` (che è *separato* dal `composer.json` di MW) e permette l'aggiunta di pacchetti a uso dell'utente
- **Esempio** personalizzato a lato
  - In blu il codice richiesto da WB. In grigio una mia aggiunta, come esempio (sorgente: *packagist.org*)
- **Dopo** aver preparato il file, come abbiamo visto, il comando è `composer install --no-dev`

```
{
  "require": {
    "monolog/monolog": "~1.25.5",
    "lucamauri/page-to-github": "~1.0"
  },
  "extra": {
    "merge-plugin": {
      "include": ["extensions/Wikibase/composer.json"]
    }
  }
}
```

# Configurazione

- Aggiungere a `LocalSettings.php` il codice qui a fianco per attivare l'estensione
  - Nel **Server** ho attivato entrambi i moduli, nel **Client** solo il secondo
- I due "`require_once`" includono file con la configurazione **base**
  - La andremo a modificare, ma guardateli
- Lanciare poi lo script di aggiornamento `php maintenance/update.php`

```
$wgEnableWikibaseRepo = true;  
require_once "$IP/extensions/Wikibase/repo/Wikibase.php";  
require_once "$IP/extensions/Wikibase/repo/ExampleSettings.php";  
  
$wgEnableWikibaseClient = true;  
require_once "$IP/extensions/Wikibase/client/WikibaseClient.php";  
require_once  
"$IP/extensions/Wikibase/client/ExampleSettings.php";
```



# Prove

- Il Server funzionerà immediatamente
  - se tutta la configurazione è stata fatta per bene, nella pagina **Speciale** apparirà una intera sezione a proposito di WikiBase in cui, tra le altre cose, si potrà **creare** il primo *Item*
- Il Client, al **primo** avvio restituirà un errore nel database perché la connessione con il Server non è ancora stata configurata
  - Vedremo, nelle diapositive che seguono, come farlo

# Configurare il client

- La **descrizione** completa delle configurazioni è [https://doc.wikimedia.org/Wikibase/master/php/md\\_docs\\_topics\\_options.html](https://doc.wikimedia.org/Wikibase/master/php/md_docs_topics_options.html) noi qui discutiamo di quelle basilari
  - A lato un esempio **generico** delle impostazioni base

```
$wgWBClientSettings['repoUrl'] = 'https://server/wiki';  
$wgWBClientSettings['repoArticlePath'] =  
'/index.php/$1';  
$wgWBClientSettings['repoScriptPath'] = '/';  
$wgWBClientSettings['siteGlobalID'] = 'wikiname';  
$wgWBClientSettings['repositories']['']['repoDatabase']  
= 'wbwikidb';  
$wgWBClientSettings['repositories']['']['changesDatabase']  
= 'wbwikidb';  
$wgWBClientSettings['repositories']['']['siteGroup'] =  
'wikigroup';
```

# Configurare il client - URL

- I primi parametri hanno a che fare con la posizione fisica del **server** e delle relative cartelle
  - Se si è incerti si possono verificare in `Special:Version`

```
$wgWBClientSettings['repoUrl'] = 'https://server/wiki';  
$wgWBClientSettings['repoArticlePath'] =  
'/index.php/$1';  
$wgWBClientSettings['repoScriptPath'] = '/';  
$wgWBClientSettings['siteGlobalID'] = 'wikiname';  
$wgWBClientSettings['repositories']['']['repoDatabase']  
= 'wbwikidb';  
$wgWBClientSettings['repositories']['']['changesDatabase']  
= 'wbwikidb';  
$wgWBClientSettings['repositories']['']['siteGroup'] =  
'wikigroup';
```

# Configurare il client - database

- Queste due configurazioni indicano al client quale sia il **database** da leggere, normalmente si tratta del database come denominato nel Server
  - Non ho trovato descrizione tecnica esaustiva del secondo dei due parametri

```
$wgWBClientSettings['repoUrl'] = 'https://server/wiki';  
$wgWBClientSettings['repoArticlePath'] =  
'/index.php/$1';  
$wgWBClientSettings['repoScriptPath'] = '/';  
$wgWBClientSettings['siteGlobalID'] = 'wikiname';  
$wgWBClientSettings['repositories']['']['repoDatabase']  
= 'wbwikidb';  
$wgWBClientSettings['repositories']['']['changesDatabase']  
= 'wbwikidb';  
$wgWBClientSettings['repositories']['']['siteGroup'] =  
'wikigroup';
```

# Configurare il client - identificativi

- Queste due impostazioni rappresentano il *nome* e il *gruppo* con cui il **Server** identifica questo client
  - Manipoleremo queste informazioni nella tabella `sites` del Server
  - Saranno fondamentali per configurare e usare i **SiteLinks**

```
$wgWBClientSettings['repoUrl'] = 'https://server/wiki';  
$wgWBClientSettings['repoArticlePath'] =  
'/index.php/$1';  
$wgWBClientSettings['repoScriptPath'] = '/';  
$wgWBClientSettings['siteGlobalID'] = 'wikiname';  
$wgWBClientSettings['repositories']['']['repoDatabase']  
= 'wbwikidb';  
$wgWBClientSettings['repositories']['']['changesDatabase']  
= 'wbwikidb';  
$wgWBClientSettings['repositories']['']['siteGroup'] =  
'wikigroup';
```

# Funziona, quasi

- A questo punto è già possibile **usare** i dati del Server in una pagina del Client usando il cosiddetto *Arbitrary Access*, spiegato qui [https://www.wikidata.org/wiki/Wikidata:How\\_to\\_use\\_data\\_on\\_Wikimedia\\_projects#Arbitrary\\_access](https://www.wikidata.org/wiki/Wikidata:How_to_use_data_on_Wikimedia_projects#Arbitrary_access)
  - Si può per esempio usare il wikicode `{{#statements:birth name|from=Q42}}` per visualizzare una proprietà *birth name* dall'elemento *Q42*
    - Si tratta di un esempio da WikiData, ovviamente in ogni istanza personalizzata di WikiBase queste estrae risultati personalizzati
- Questo è versatile ma scomodo: si vuole **normalmente** usare il *Direct Access*, da usare nei template, per esempio
  - Ovvero quella funzionalità con cui si estraggono solo le proprietà `{{#statements:P1477}}` perché Server e Client **sanno** quale pagina è collegata a quale entità

# Prime difficoltà

- Per quanto possa sembrare strano, fino alla primavera del 2019, da nessuna parte era spiegato **chiaramente** come dovessero essere configurati il Server e il Client per fare funzionare il Direct Access
  - Dopo alcune domande mie e di altri utenti ([https://www.wikidata.org/wiki/Wikidata:Contact\\_the\\_development\\_team/Archivio/2019/03#Help\\_with\\_Direct\\_Access\\_on\\_custom\\_Wikibase](https://www.wikidata.org/wiki/Wikidata:Contact_the_development_team/Archivio/2019/03#Help_with_Direct_Access_on_custom_Wikibase)), alcuni di noi hanno contribuito a chiarire la documentazione, come descritto in <https://phabricator.wikimedia.org/T218282>
- La chiave per la soluzione del problema sono i **sitelinks**, ovvero quella caratteristica del Server che permette di collegare una entità a una pagina su una o più istanze di MediaWiki
  - Questo permette alla pagina di una wiki configurata come Client di **accedere direttamente** all'entità
  - Permette anche di conoscere le pagine su altre **wiki** che sono collegate all'entità
    - Agisce quindi come una sorta di "external link"

# La tabella `sites`

- Il modo che hanno le istanze di MW di conoscere e **interfacciarsi** con altri siti è, appunto, la omonima tabella `sites`
  - Non serve *solo* a WB, ma qui la vediamo con questo scopo
- La tabella `sites` può essere popolata in due modi
  - Per tutti i siti rilevanti (anche esterni alla fondazione, ma arcinoti) per Wikimedia
    - tramite script `populateSitesTable.php`
  - Per i propri siti personali
    - tramite vari metodi, il più semplice dei quali è **l'importazione** di un file XML
- Vediamo come impostare un proprio elenco personale



# Il file XML...

- Un file XML formattato come l'esempio a lato è il modo più comodo per importare siti nella tabella `sites`
- La documentazione relativa è disponibile all'indirizzo [https://www.mediawiki.org/wiki/Manual:Site\\_table](https://www.mediawiki.org/wiki/Manual:Site_table)

```
<?xml version="1.0"?>
<sites xmlns="http://www.mediawiki.org/xml/sitelist-1.0/" version="1.0">
  [...]
  <site type="mediawiki">
    <globalid>enma</globalid>
    <group>wikia</group>
    <path type="page_path">https://memory-alpha.fandom.com/wiki/$1</path>
    <path type="file_path">https://memory-alpha.fandom.com/$1</path>
    <languagecode>en</languagecode>
  </site>
  [...]
  <site type="mediawiki">
    <globalid>enwiki</globalid>
    <group>wikipedia</group>
    <path type="page_path">https://en.wikipedia.org/wiki/$1</path>
    <path type="file_path">https://en.wikipedia.org/w/$1</path>
    <languagecode>en</languagecode>
  </site>
  [...]
</sites>
```

# ...importato in sites

- Il file si importa con lo **script** di manutenzione `importSites.php`
  - Incidentalmente, esiste anche il corrispondente `exportSites.php`
- Il risultato sono **righe** della tabella popolate in questa maniera
  - Vedete <https://phabricator.wikimedia.org/T221149> per una discussione sul codice linguaggio

site_id	site_global_key	site_type	site_group	site_source	site_language	site_protocol	site_domain	site_data
1	enma	mediawiki	wikia	local	en	https	moc.modnaf.ahpla-yromem.	a:1:{s:5:"paths";a:2:{s:9:"page_path";s:39:"https://memory-alpha.fandom.com/wiki/\$1";s:9:"file_path";s:34:"https://memory-alpha.fandom.com/\$1";}}
2	enwiki	mediawiki	wikipedia	local	en	https	gro.aidepikiw.ne.	a:1:{s:5:"paths";a:2:{s:9:"page_path";s:32:"https://en.wikipedia.org/wiki/\$1";s:9:"file_path";s:29:"https://en.wikipedia.org/w/\$1";}}

# Configurare il Server

- La descrizione completa delle configurazioni è [https://doc.wikimedia.org/Wikibase/master/php/md\\_docs\\_topics\\_options.html](https://doc.wikimedia.org/Wikibase/master/php/md_docs_topics_options.html)
  - noi qui discutiamo di quelle più rilevanti

```
$wgWBRepoSettings['siteLinkGroups'] = [ 'wikipedia',  
    'wikigroup', 'wikia', 'special' ];  
  
$wgWBRepoSettings['specialSiteLinkGroups'] = array(  
    'commons', 'wikidata', 'gamepedia' );  
  
$wgWBRepoSettings['formatterUrlProperty'] = 'P5';  
  
$wgWBRepoSettings['conceptBaseUri'] = $wgServer .  
$wgScriptPath . '/index.php/Special:EntityData/';  
// 'http://server/wiki/entity/';
```

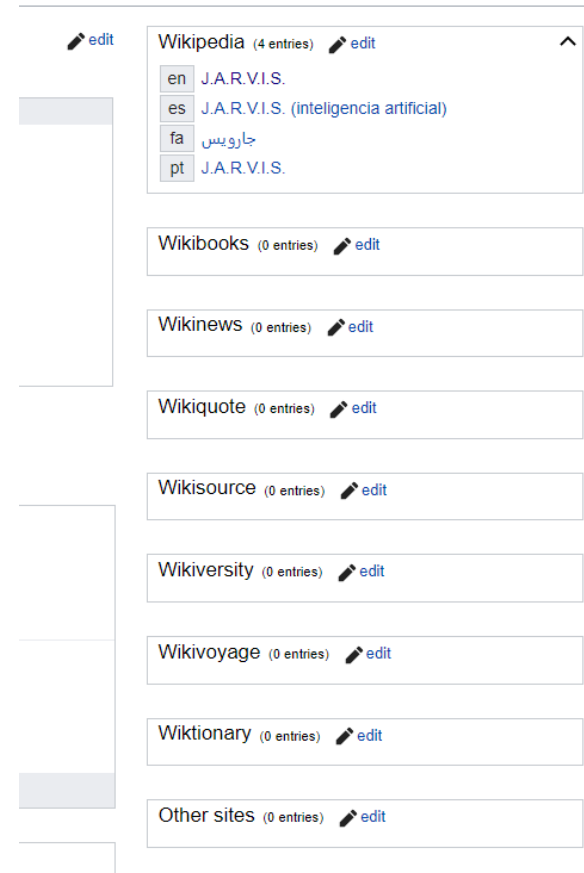
# Configurare il Server - SiteLinks

- Le prime due righe definiscono i siti che devono essere **inclusi** nella colonna dei SiteLinks
- I nomi definiti qui sono quelli del campo `site_group` nella tabella `sites` visti in **precedenza**

```
$wgWBRepoSettings['siteLinkGroups'] = [ 'wikipedia',  
    'wikigroup', 'wikia', 'special' ];  
  
$wgWBRepoSettings['specialSiteLinkGroups'] = array(  
    'commons', 'wikidata', 'gamepedia' );  
  
$wgWBRepoSettings['formatterUrlProperty'] = 'P5';  
  
$wgWBRepoSettings['conceptBaseUri'] = $wgServer .  
$wgScriptPath . '/index.php/Special:EntityData/';  
// 'http://server/wiki/entity/';
```

# SiteLinks

- Ecco un **esempio** di SiteLinks dalla pagina del Q65089566
  - Lo *item* a proposito della immaginaria AI JARVIS
- Vedrete questa colonna ora con occhi nuovi per la sua **importanza**
  - Notate in particolare i *groups* e lo *special*



# Razionalizzare SiteLinks e Interwiki

- Nelle mie peregrinazioni in WikiBase, mi sono reso conto che è molto comodo trattare SiteLink e Interwiki come un **tutt'uno**
- Mi sono creato quindi un **file** CSV con queste colonne e con l'elenco di tutti i siti necessari sia ai SiteLink che agli Interwiki

globalkey / iw_prefix	group	page_path / iw_url	iw_api	file_path	languagecode	description
enma	wikia	https://memory-alpha.fandom.com/wiki/\$1	https://memory-alpha.fandom.com/api.php	https://memory-alpha.fandom.com/\$1	en	Memory Alpha (inglese)
enwiki	wikipedia	https://en.wikipedia.org/wiki/\$1	https://en.wikipedia.org/w/api.php	https://en.wikipedia.org/w/\$1	en	Wikipedia (inglese)

# Razionalizzare SiteLinks e Interwiki

- Ho scritto poi un tool che, a partire da questo elenco, **genera**
  - Il file XML da dare in pasto allo script di importazione come sopra
  - Una serie di *statement* SQL da usare per inserire i prefissi InterWiki nella apposita tabella
  - Un frammento di codice LUA che unisce *Key* e descrizioni
- Il vantaggio grande è che in questa maniera la stessa *Key* di un sito vale **anche** come prefisso interwiki e le relative **descrizioni** si trovano in un unico posto. Si può gestire così in maniera più consistente i dati
- Per approfondire potete leggere questo post <https://dev.to/lucamauri/organize-interwiki-and-sites-in-mediawiki-for-best-efficiency-360e>

# Configurare il Server – Formatter URL

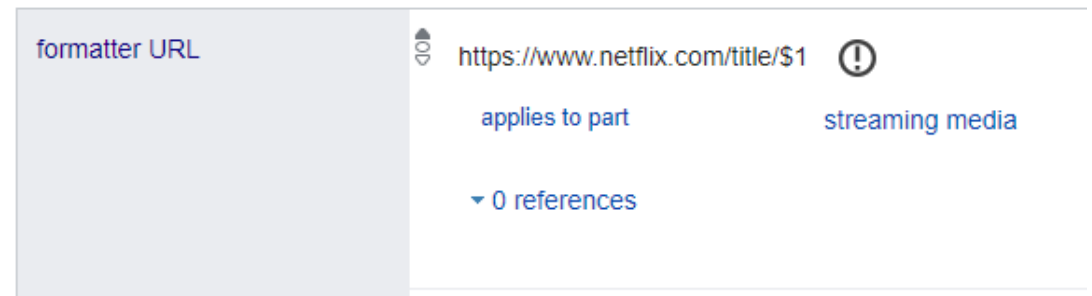
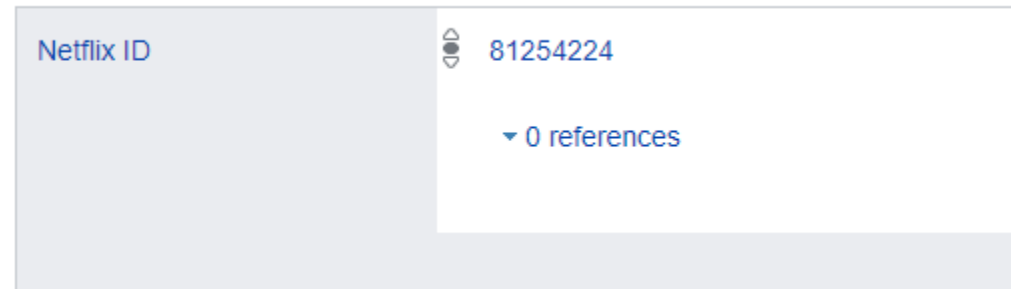
- Questo parametro definisce globalmente una *Property*, scelta e definita dall'utente, che contiene un **modello** che, associato a una stringa, genera un URI completo di riferimento
  - vediamo un esempio pratico

```
$wgWBRepoSettings['siteLinkGroups'] = [ 'wikipedia',  
    'wikigroup', 'wikia', 'special' ];  
  
$wgWBRepoSettings['specialSiteLinkGroups'] = array(  
    'commons', 'wikidata', 'gamepedia' );  
  
$wgWBRepoSettings['formatterUrlProperty'] = 'P5';  
  
$wgWBRepoSettings['conceptBaseUri'] = $wgServer .  
$wgScriptPath . '/index.php/Special:EntityData/';
```



# Formatter URL

- L'elemento di un programma Netflix contiene la stringa 81254224 nella proprietà Netflix ID
- La proprietà Netflix ID contiene a sua volta una proprietà Formatter URL che contiene il modello `https://www.netflix.com/title/$1`
- Il risultato, nella pagina del programma, è automaticamente generato come <https://www.netflix.com/title/81254224>



# Configurare il Server – Base URI

- Questo parametro definisce il percorso per mappare un **concetto** secondo RDF, come `http://server/entity/Q2` all'indirizzo reale della entità
- Un vantaggio collaterale è che è possibile indirizzare <http://server/entity/Q2.json> verso la speciale <https://server/wiki/Special:EntityData/Q2.json>

```
$wgWBRepoSettings['siteLinkGroups'] = [ 'wikipedia',  
    'wikigroup', 'wikia', 'special' ];  
  
$wgWBRepoSettings['specialSiteLinkGroups'] = array(  
    'commons', 'wikidata', 'gamepedia' );  
  
$wgWBRepoSettings['formatterUrlProperty'] = 'P5';  
  
$wgWBRepoSettings['conceptBaseUri'] = $wgServer .  
$wgScriptPath . '/index.php/Special:EntityData/';
```

# Sezioni

- È possibile **raggruppare** degli *statements* secondo vari criteri
- Un esempio tipico è aggregare gli identificativi **esterni** e separarli dal resto degli *statements* nella pagina
  - WikiData usa questa convenzione dato che fa uso **estensivo** di questi dati
  - Vediamo come **replicare** questa impostazione

The screenshot displays a Wikidata page with several data entries grouped into sections. Each entry includes a label, a value, and an edit icon. Some entries also show a reference count and an 'add value' button.

- surface gravity**: 9.807 metre per second squared. 1 reference. + add value
- minimum temperature record**: -89.2 degree Celsius. 1 reference. Includes sub-properties: point in time (21 July 1983) and location (Vostok Station). + add value, + add statement
- Identifiers** section:
  - VIAF ID**: 6270149919445006650001. 1 reference. + add value
  - Bibliothèque nationale de France ID**: 11975911n. 0 references. + add reference, + add value
  - GND ID**: 1135962553. edit

# Sezioni

1 reference  
+ add value

surface gravity 9.807 metre per second squared edit  
1 reference  
+ add value

minimum temperature record -89.2 degree Celsius edit  
point in time 21 July 1983  
location Vostok Station  
1 reference  
+ add value  
+ add statement

Identifiers

VIAF ID 6270149919445006650001 edit  
1 reference  
+ add value

Bibliothèque nationale de France ID 11975911n edit  
0 references  
+ add reference  
+ add value

GND ID 1135962553 edit

```
$wgWBRepoSettings['statementSections'] = [  
  'item' => [  
    'identifiers' => [  
      'type' => 'dataType',  
      'dataTypes' => [ 'external-id' ],  
    ],  
  ],  
];
```

# Carenza seria

- Impossibile fare query **inverse**, ovvero non è possibile chiamare con una singola funzione un elenco di elementi che fanno riferimento a una proprietà
  - Esempio pratico: avete un insieme di elementi che rappresentano episodi di una serie e, in ogni episodio, un elenco di attori
  - Non è possibile ottenere con una singola chiamata LUA l'elenco degli episodi in cui appare un determinato attore.
- C'è una proposta per creare la funzione `mw.wikibase.entity:getBacklinks` discussa qui <https://phabricator.wikimedia.org/T185313>
  - Ma è ancora molto lontana dall'essere realizzata, se mai lo sarà
- Per risolvere questo problema, ho dovuto affrontarlo **cambiando** completamente prospettiva

# Entra *Semantic MediaWiki*

- Si tratta di una estensione aggiuntiva, *separata* da WikiBase
  - Spesso vengono viste come **alternative**, perché vengono entrambe usate per memorizzare dati strutturati
  - Ma in realtà sono diverse: una consente di strutturare dati nelle **pagine in** maniera semantica, l'altra è più simile a un **database**
- SMW consente di **risolvere** velocemente il problema delle *query inverse* come appena discusso
- Dopo aver considerato se usare una al posto dell'altra
  - Mi sono reso conto di poterle fare lavorare **insieme** con un impegno moderato

# Mettere WB e SMW al lavoro insieme

- In due parole vi spiego che ho fatto in modo di:
  - Inserire i dati in una voce della wiki non come *semplice* testo, ma come testo accompagnato dai relativi tag di SMW secondo il modello `[[Property name::property value]]`
  - Così leggo da WB non solo il dato, ma anche la sua **caratterizzazione** in SMW
  - In questa maniera io presento il dato all'utente, in più lo rendo *anche* **semantico** e posso lavorarci sopra tramite codice
  - Senza carico aggiuntivo per l'utente che contribuisce
- Per chi è interessato ad approfondire l'argomento SMW, terrò una breve sessione alla *SMWCon 2020* [https://www.semantic-mediawiki.org/wiki/SMWCon\\_Fall\\_2020](https://www.semantic-mediawiki.org/wiki/SMWCon_Fall_2020)

# Pensieri vari

- Il WikiBase **Registry**
  - <https://wikibase-registry.wmflabs.org/>
- WikiBase **Local Media**
  - una estensione sviluppata da terze parti, diffusa con GPL che permette di inserire media senza necessariamente usare *Commons*. Si trova a <https://github.com/ProfessionalWiki/WikibaseLocalMedia>
- WikiBase **API**
  - Funziona esattamente come la API di MediaWiki, è possibile usare i tools dedicati a MW. Io, per esempio, uso *DotNetWikiBot* e sto scrivendo una estensione per WB. La documentazione, anche se un po' carente, si trova a <https://www.mediawiki.org/wiki/Wikibase/API>



# Pensieri vari

- Universal **Language** Selector

- È di importanza fondamentale per visualizzare le lingue alternative nel pannello in alto di ogni elemento, dove si trovano le etichette e le descrizioni. Senza, si vede solo la lingua della Wiki. Si installa come una normale estensione e necessita poca o nessuna configurazione

<https://www.mediawiki.org/wiki/Extension:UniversalLanguageSelector>

- **Federazione**

- Nelle intenzioni, *tutte* le istanze di WB pubblicate su internet dovrebbero conoscersi e poter essere usate insieme. Dovrebbe essere possibile prendere dati da una istanza e *aggiungere* informazioni senza dovere reinventare la ruota ogni

volta. [https://www.wikidata.org/wiki/Wikidata:Federation\\_input](https://www.wikidata.org/wiki/Wikidata:Federation_input) può essere un buon punto di partenza per indagare su questo futuro sviluppo

# Grazie, tocca a voi

