Handling external data in MediaWiki

Yaron Koren · EMWCon Spring 2022
April 6, 2022 · Houston, TX
About me

- Based in New Jersey
- My extensions: Page Forms, Cargo, etc.
- My company: WikiWorks
- My book: *Working with MediaWiki*
- My podcast: *Between the Brackets*
External data (lowercase "d")

• The amount of data outside a system always dwarfs the amount of data inside the system (even for Wikidata)
• It’s best not to have redundant data (best is "single source of truth")
Two basic approaches to handling external data

1) Import the data into the wiki, abandoning the outside system
   - Various solutions for this, including bots and the Data Transfer extension
2) Keep the data in the outside system and query it in the wiki
   - Standard extension for this: External Data
   - Having the data in both systems, with some attempt at automated „syncing“, is rarely a good idea.
External Data (uppercase "D")

- A MediaWiki extension that lets you query and display outside data sources
- Created by me in 2009, improved and extended by many people
- Data retrieved can be either a single row of data (displayed via \texttt{#external\_value}), or a table (displayed via \texttt{#for\_external\_table}, \texttt{#display\_external\_table} or \texttt{#format\_external\_table})
- All functionality available via Scribunto/Lua as well
External Data: possible data sources

- Web pages (CSV, JSON, XML, plain text, HTML, INI, GFF)
- Web pages via SOAP (same format options)
- Local files (same format options)
- Databases
- LDAP servers
- Local executables
Examples pages

#get_web_data examples:
https://discoursedb.org/wiki/ED_get_web_data_examples

#get_program_data examples:
https://traditio.wiki/Традиция:Test/ED/exe
New features

In the last six months, the following features have been added (thanks to Alexander Mashin):

- `#get_program_data`
- `#get_external_data` (single catch-all function)
- Prepared SQL statements in `#get_db_data`
- INI format
- `#format_external_table` (uses Cargo to format results)
- ...and many others
External data in forms

- External data is also useful for setting possible values in forms.
- "values from external data=" – populates dropdowns, comboboxes, etc. with values retrieved by a call to \#get_web_data, \#get_db_data, etc.
Future complication: Parsoid

- Parsoid (a MediaWiki parser) parses different parts of a wiki page independently, not necessarily in order.
- Once Parsoid becomes the default parser, parser functions that rely on order (like `#get_web_data` + `#external_value`) will no longer necessarily work correctly.
- External Data is not the only extension with a problem.
PARSOID AS DEFAULT PARSER

VARIABLES, NUMERALPHA, EXTERNAL DATA AND SMW
What can be done?

- Variables and NumerAlpha may be screwed. :(  
- For External Data, we need a way to retrieve and display data at the same time.
One option: new parameters

{{#get_web_data:
url=https://imdb-api.com/en/API/Title/IMDB_KEY/tt0055630
|format=json
|data=title=title,year=year,directors=directors,
stars=stars
|display template=Display IMDb film}}

...this would handle data retrieval and display!
"Display IMDb film" template

Info for ID "...uh oh, what goes here?":

* Name: {{{title}}}
* Year: {{{year}}}
* Director(s): {{{directors}}}
* Actors: {{{stars}}}

...uh oh, what goes here?
We probably need another parameter

```json
{{#get_web_data:
url=https://imdb-api.com/en/API/Title/IMDB_KEY/tt0055630
|format=json
|data=title=title,year=year,directors=directors,
stars=stars
|display template=Display IMDb film
|additional values=film ID=tt0055630}}
```
Info for ID "{{film ID}}":
* Name: {{title}}
* Year: {{year}}
* Director(s): {{directors}}
* Actors: {{stars}}
Finally, a third new parameter

```yaml
{{#get_web_data:
url=https://imdb-api.com/en/API/Title/IMDB_KEY/tt0055630
|format=json
|data=title=title,year=year,directors=directors,
stars=stars
|display format=table}}
```

Just as with #format_external_table, uses Cargo to display values.
Questions/comments/concerns