Making a MediaWiki Skin
I am Jon Robson

- Work as a developer at WMF.
- Frontend engineer who’s had to learn PHP.
- Previously built Minerva
- Now working on Vector 2022 skin
Session goals

- (We only have 30 minutes, not 45!)
- Get more Wikimedians involved in skin development
- Summarize some of the changes that have been made to skins and why
- Create a skin
- Identify skin related projects for the hackathon
- Arrange follow up sessions over the weekend for people who want to build skins
- NOT to talk about new Vector skin. See session tomorrow for that.
91 working skins in the MediaWiki universe

MediaWiki can be used in 91 skins. MediaWiki development began in 2002. 20 years later...

How can we grow the skin building community?

How can we get more?
Making skins was complicated

- Required PHP knowledge
- JS and CSS knowledge
- Ability to navigate MediaWiki core’s documentation
- Skins had to spend lots of time tweaking themselves to work with different extensions
- Skins would often break with changes in core and extensions
Desktop improvements

- WMF project.
- Meant that the developers maintaining mobile site for the first time were also maintaining the desktop site.
- Seized opportunity to start standardizing all the good ideas across all the deployed skins (Timeless, Minerva, Monobook, Vector, CologneBlue, Modern).
- Connected with skin community and started to think about skin architecture.
Model <-> View

- Skins should be dumb. They are a visual representation of data.
- Re-architected skins to share a data layer
- Used Mustache as the view layer
- Converted Vector, Modern, CologneBlue, MonoBook to Mustache
- Not possible without help from volunteer (waves to Ammarpad, Mainframe, Ashley, Alistair, Isarra, Lens and all the people I forgot!)
Reduced skin to 3 files + optional learn later config

- Skin.mustache
- Skin.css
- Skin.js
- Configuration
  - skin.json
  - i18n
Making skins now easier!

- No PHP knowledge necessary
- Standardized hooks for extensions to integrate with skins
- Various tools to help you get started
Mustache basics

mustache basics

{#is-talk-page}<!-- if block ->
This is a talk page.
{#/is-talk-page}

<h1>{{ title }}</h1> <!-- two braces: escape! -->
{{ html }} <!-- three braces = HTML:-->
Let’s make a skin!
Skin builder

https://skins.wmflabs.org/#/add
Command line

```
npm install mediawiki-skins-cli
npx create-mw-skin SkinName
cd SkinName
```

Update your LocalSettings.php:

```php
wfLoadSkin('Foo');
```
**Explore data**

https://github.com/jdlrobson/mediawiki-skins-skinjson/

```json
<table>
<thead>
<tr>
<th>JSON</th>
<th>Raw Data</th>
<th>Headers</th>
</tr>
</thead>
<tbody>
<tr>
<td>html-title-heading:</td>
<td>&quot;&lt;li id=&quot;firstHeading&quot;&gt;&lt;/li&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>html-title:</td>
<td>&quot;Selenium category test&quot;</td>
<td></td>
</tr>
<tr>
<td>is-title-blank:</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>is-anon:</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>is-article:</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>is-mainpage:</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>is-special-page:</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>data-logos:</td>
<td>(...)</td>
<td></td>
</tr>
<tr>
<td>data-toc:</td>
<td>[]</td>
<td></td>
</tr>
<tr>
<td>data-search-box:</td>
<td>(...)</td>
<td></td>
</tr>
<tr>
<td>data-portlets:</td>
<td>(...)</td>
<td></td>
</tr>
<tr>
<td>data-portlets-sidebar:</td>
<td>(...)</td>
<td></td>
</tr>
<tr>
<td>data-root:</td>
<td>(...)</td>
<td></td>
</tr>
<tr>
<td>array-indices:</td>
<td>[]</td>
<td></td>
</tr>
<tr>
<td>html-site-notice:</td>
<td>&quot;&lt;div id=&quot;localNotice&quot;&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>html-user-message:</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>html-subtitle:</td>
<td>&quot;&lt;div id=&quot;mw-content-text&quot;&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>html-body-content:</td>
<td>&quot;&lt;div id=&quot;catlinks&quot;&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>html-categories:</td>
<td>&quot;&lt;div id=&quot;catlinks&quot;&gt;&quot;</td>
<td></td>
</tr>
</tbody>
</table>
| html-after-content: | "<div class="skin-js-hook-validation-element skin-js-validation-element" data-hook="SkinAfterContent"></div>
| html-undele-link:   | null                               |         |
| html-user-language-attributes: | null |         |
| link-mainpage:      | "wiki/Home"                        |         |
```
Hackathon projects
Make a new skin

New skins help us identify better architecture

- Follow guide on https://www.mediawiki.org/wiki/Manual:How_to_make_a_MediaWiki_skin
- Publish to github/ gerrit
- Add your skin to mediawiki.org!
- Blog about your experience / open Phabricator tickets
Port an existing (Wordpress) skin

I have been experimenting with converting Wordpress skins into MediaWiki skins.

- Find a Wordpress theme you like
- Try porting it with the experimental converter tool
- Help me make it better :-)

[[Category:Skins_using_deprecated_features]] <!-- Use of BaseTemplate::getFor... -->
Fork an old skin

Many skins that used to work were archived.

How to help:
- Find a skin in https://www.mediawiki.org/wiki/Category:Archived_skins that is worth saving.
- Resurrect it
Fix an existing skin

Many skins are using deprecated code and will break in a future release.

How to help:
- Go to mw.org category
- View source of the skin for details
- Submit pull request
Port an existing skin

Many skins are written in PHP. Might be a lot simpler if written using SkinMustache

- Read
- Change
- Update docs
Improve architecture

Help the skins of the future by improving all skins.

How to help:
● Go to our Phabricator board
● Pick a task!
Let’s talk more

- Around on IRC today (#wikimedia-hackathon) and Telegram group to talk more and help you get started
- We can arrange a follow up session to help anyone stuck.