

# **Taking advantage of new Config features for extensions**

MediaWiki Developer Summit 2015

# What? Why?

- We want to move configuration out of \$GLOBALS
- And other extension registration stuff.

# Configuration settings

- Config interface provides accessors to accessing settings (1.23+)
- ConfigFactory stores individual Config objects
- GlobalVarConfig is an implementation of Config that just reads from \$GLOBALS

# Config (continued)

- Dropping the “wg” prefix, no more global namespace
- Manual-ish dependency injection
- Is your thing actually a config setting or more of an extension registration type thing?

# This is awesome! How do I use it?

```
$wgConfigRegistry[ 'ext-name' ] = 'GlobalVarConfig::
newInstance'
$config = ConfigFactory::getDefaultInstance() ->makeConfig
( 'ext-name' );
$config->get( 'FooBar' ); // no "wg" prefix
RequestContext::getMain() ->getConfig();
ConfigFactory::getDefaultInstance() ->makeConfig( 'main' );
```

[https://www.mediawiki.org/wiki/Manual:Configuration\\_for\\_developers](https://www.mediawiki.org/wiki/Manual:Configuration_for_developers)

# Creating extension.json

- 1.25 includes a new system called extension registration.
- `convertExtensionToRegistration.php` script
- Need to keep the PHP entry point for backwards-compatibility (and at least until jenkins and WMF production is updated)
- `validateRegistrationFile.php` to check against the schema (need to have schema library via composer)
- [https://www.mediawiki.org/wiki/Extension\\_registration](https://www.mediawiki.org/wiki/Extension_registration)

# Try it!

- Pick a small-ish extension to start with for Config
- Registration conversion script might throw errors, usually exposing weird patterns