Helping functions and methods

Method	Return	Description
ResursBank\Module\Data::getPrefix($extra)	A prefix	Returns the default prefix that is used by the plugin internally. Adding an extra variable, will return "the_prefix_$extra". For example, all configuration data are using "admin" as an extra parameter, so currently the returned data is **trbwc_admin**. Getting configuration data in this way, for example if you want to determine if the plugin is enabled the complete option key in the database looks like **trbwc_admin_enabled**.
ResursBank\Module\Data::getImage($imageName)	URL	$imageName is the short name (i.e. *"logotype"*) of an image that usually resides in /images in the plugin structure. *Loading will occur with an autodetected file extension which limits the risks of loading something not allowed.* Example: **getImage('author-photo')** Returns: **https://url.com/plugin/url/author-photo.[jpg|png|gif]**
ResursBank\Module\Data::getGatewayPath($subDirectory)	Absolute path to plugin structure.	By entering a subdirectory name it als returns */full/path/to/subDirectory*. Example: **getGatewayPath('images')** Returns **/full/path/to/plugin/images**
ResursBank\Module\Data::applyFilters($filterName, $value[, $args])	Whatever that is applied.	This is actually a standard apply_filters, but with a helper that always adds a proper prefix to every filters applied. Example: **applyFilters('checkout')** Will apply: **rbwc_checkout** **Note: Compare to applyFiltersDeprecated that instead generates "resurs_bank_checkout", to comply with prior plugin releases.**
ResursBank\Module\Data::getVersionByComposer		Always returns version number that lies within composer.json.
ResursBank\Module\Data::getCurrentVersion		Always returns version number that lies within the initializer.
ResursBank\Module\Data::getValidatedVersion	boolean	Returns true if both composer-version and internal version is the same. If not, the plugin should warn inside wp-admin that someone forgot to update this data.

ResursBank\Module\Data::getGenericClass	Generic::class	Returns a Generic::class with properly configured template path pointing too plugin-directory/templates - adding templates to that directory in the extension .html makes it possible to call the class like this:
		
		`$content .= self::getGenericClass()->getTemplate(`
		`            'plugin_information',`
		`            [`
		`                'required_drivers' => self::getSpecialString('required_drivers'),`
		`                'render' => $renderData,`
		`            ]`
		`        );`
		
		The variables added to the second argument in the getTemplate-method could then be used like this:
		
		`Required drivers: <?php echo $required_drivers ?>`

ResursBank\Module\Data::getResursOption	A value	Using getResursOption directly will give you the opportunity to just send in a saved option key in the method. The default parameter keys are prefixed trbwc_admin and is added automatically. It also checks values against "boolean-like" valued strings like yes/true/no/false (via getTruth below). As the prefix COULD possibly change, it is recommended to always use those internals.
ResursBank\Module\Data::getTruth($value)	boolean or null	If you are using get_options directly, you can use this function to determine if the returned option should be considered a boolean or string (on the values yes/true and false/no that is the default boolean value returned from a woocommerce setup). If the returned value is null, then you can use it as a string. This is entirely handled by Data::getResursOption if you decide to use that instead.
ResursBank\Module\Data::getFormFields($section)	$formFieldArray	Form fields for admin based on section.

ResursBank\Module\Data::setLogInternal($type, $message)	-	To send logs to Woocommerce natural logger, setLogInternal can be used. The datatypes in the first argument can be found in the Data class and are defined as below. This first introduced an action, but was later converted to a static method. In this plugin, the types are just used to mark severity, but uses respecitvely method in WC_Logger anyway:

Constant Type/Name	Description	WC_Logger()	Suggested when
Data::LOG_DEBUG	Debug information.	debug()	Developer notes.
Data::LOG_NOTICE	Notice only (default).	notice()	Notices to merchant and less important data.
Data::LOG_CRITICAL	Critical errors.	critical()	Critical errors, I.E. when API is down.
Data::LOG_ERROR	Not critical but an error.	debug()	
Data::LOG_WARNING	Not an error. Not critical. A warning.	debug()	A headsup to merchants/developers. I.E. when the plugin mismatches the required woocommerce version.

Example on how to log data:

```
Data::setLogInternal(
    Data::LOG_NOTICE,
    sprintf('Customer %s has an address located outside country.', $customer)
);
```