Developer tooling

This section describes some tooling to make a WebHare developer's live easier

Users in a support role may also find some of the tips here useful, eg the Find and Debug bookmarklets

The 'dev' module is an open-source module that ships with various tooling to help develop in WebHare. You can get it by executing:

```
1    wh module get https://gitlab.com/webhare/dev.git
```

If you have a gitlab.com account and would rather checkout using your ssh key:

```
1    wh module get git@gitlab.com:webhare/dev.git
```

wh dev::rewrite <file> cleans up XML for supported filetypes (eg siteprofiles) and can even clear up some deprecated constructs.

wh dev:cleanup executes a cleanup on a folder or an entire module

wh dev:rewrite can also rewite siteprofiles to the 'simpler' syntax enabled by WebHare 4.28 (eg simplified filetypes with kind= and embedded members). To opt-in to this rewrite your module must explicitly declare that it won't support older WebHare versions, eg by setting webhareversion=">= 4.28.0-dev" on its packaging node:

```
1    <packaging download="true" webhareversion=">= 4.28.0-dev">
2    </packaging>
```

Bookmarklets

These bookmarklets can make life a bit easier when using or developing on WebHare. To use these bookmarks, enable your browser's bookmark and drag the hyperlinks directly to the bar.

Find - open the current output page in WebHare

Debug - modify debug settings for the current page

On an iOS device, bookmark this page and name it Debug, Find or something like that. Then edit the bookmark and update the link to:

```
javascript:(function(){window.open(location.origin+'/.publisher/common/
```

Frontend debugmode

You should generally have 'output debugmode' enabled (see above) so changes to JS/CSS will take effect earlier.
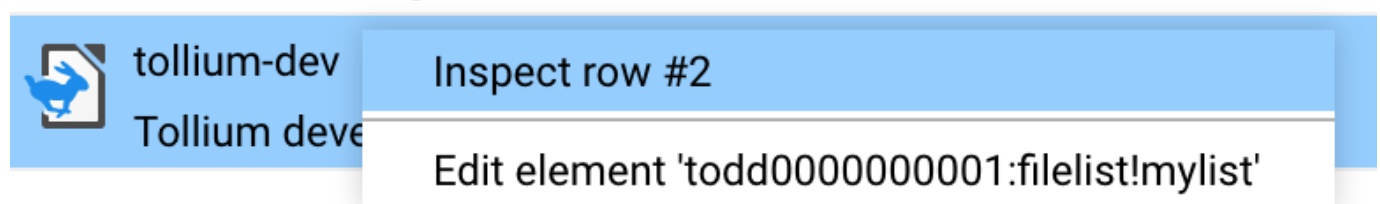
When developing static pages you can often use the live preview (ctrl+shift+b) in the Publisher to get a dynamic and easily refreshable page.

If you have the dev agent set up, this Connect bookmarklet allows you to connect it to your current WebHare installation

Tollium development

Inspect mode

You can alt-shift-rightclick any Tollium component to open a special context menu that offers you some actions that are useful for development. For example, a list row may show:

'Inspect row' will show you the raw data of that list entry. 'Edit element' will open the Tollium screens definition for the selected component in your editor (if the Dev agent is properly configured).

Alt-shift-rightclick doesn't open this menu if the WebHare server you're using is not in development mode. You can still access this menu by first activating Tollium's inspect mode by clicking three times on the small black bar at the top of the brower window (above the WebHare logo and application tabs).

The black bar will flash blue to indicate that inspect mode has been activated. Some actions (such as Inspect row) will still require you to be a sysop to access on a non-development server as this feature may show sensitive data (Tollium normally won't send unused list row data to the client)

Sublime

Installing the Sublime WebHare package

1. Open the Command Palette (Cmd+Shift+P on Mac)
2. Select "Package Control: Install Package"
3. Install the package named "WebHare"

Some useful commands in Sublime's WebHare Package are:

- **Documentation search**

  Opens the local WebHare installation's documentation site (/docs/). Make sure the docs site is enabled (application "Modules and configuration" => Menu => "Tollium backend" => tab "Server" => enable option "Index source code").

- **Get Last Error Stack Trace**

  Scrolls through (the errors from) the last stack trace. Jumps to the source code after selecting one of the stack trace items.

- **Codegrep**

  Search code from the local modules, both WebHare's internal modules as well as your own modules. Double click on a result to jump to the code (requires the 'mousemap' settings).

- **Symbol search**

  After selecting a (WebHare) function, macro or object, jumps to the source code.

- **Validate file**

  Compiles/validates the current file. Supports whscr, whlib, xml and siteprl.

- **Source switch**

  This command can be used to switch between a screen XML file and the accompanying WHLIB file.

To install WebHare's recommended packages, select Preferences => Package Settings => WebHare => Recommended Packages => Install all recommended packages.

Open Tollium component in Sublime

Use the following combination on a (Tollium) component to open the component's XML specification in Sublime:

Option + shift + right click

Other useful Sublime Packages

- TabsExtra

Mouse integration

To support mouse-related functions, you'll need to add extra settings. This will make the following work:

- Double click codegrep results
- Cmd + right click on a function to trigger a popup with documentation
- Cmd + shift + right click to trigger symbol definitions

To get this to work on OSX, add a file named "Default (OSX).sublime-mousemap" in the Sublime Package/User folder (*/Users/<username>/Library/Application Support/Sublime Text 3/Packages/User*) with the following contents:

```
1
```

Open Source

The WebHare Platform is available under the MPL 2.0 open source license. You can find the source code on GitLab and prebuilt images on Docker Hub. Managed hosting and commercial support for WebHare is offered by WebHare bv (and others).

See contributing for more information.