



CTOOLS PLUGINS AS ULTIMATE SOLUTION TO BUILD CUSTOM WEBSITES

Presentation - <http://bit.ly/1Mp1pJA>

Repo - https://github.com/netlooker/dc_leuven_ctools_demo



Author: Robert Czarny



AMPLEXOR

Pure Content Management

About me

Robert Czarny

(netlooker)

- Currently Drupalist freelancer
- 8,5 year in love with Drupal
- Specialisation:
 - Solutions for Government sector
 - Internal portals, websites and apps
 - Document management systems

czarny@data.pl

<https://plus.google.com/u/0/+RobertCzarny>



Agenda



- 1. How to deal with custom requirements**
 - Coder approach
 - Context & Display Suite approach
 - Ctools / Page Manager approach
- 2. Backend data processing**
 - concept which keeps your urls clean and secure
- 3. Short introduction to CTools Plugins**
 - plugin types
 - what are they for
- 4. Examples of how to use ctools plugins in real life**
 - demo for developers
 - showcasing described topics
- 5. Integration with other contrib modules**
- 6. Writing D7 CTools plugins which are ready for D8**

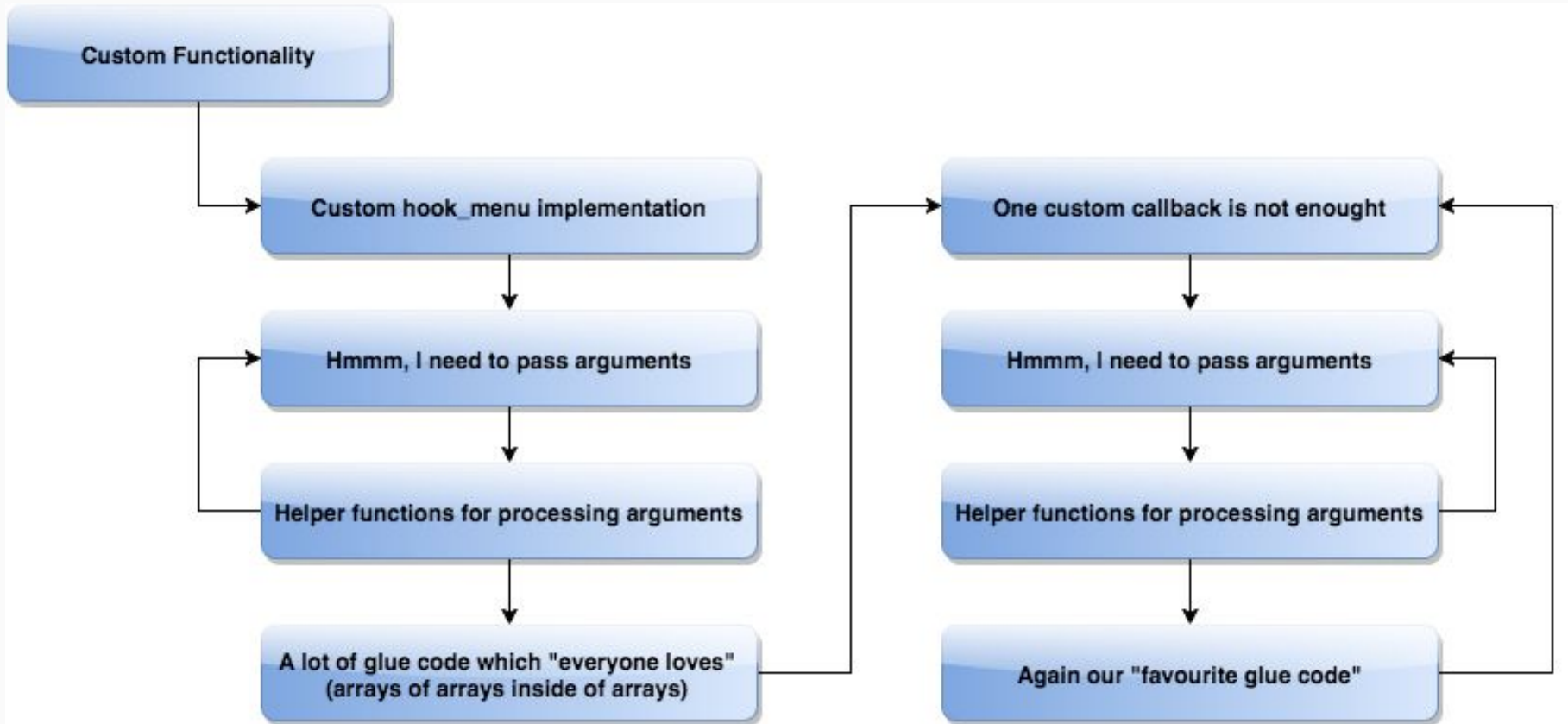
So you need to build pretty custom site

At some time you will reach point when you will have a strong belief that you are able to build anything with Drupal.

Be careful **not all of things and projects should be builded on Drupal** (but in real life there are not many of them ;)



I'm coder, I need to code - is it valid approach for solving custom requirements?



Yeah, it is always better to code than to use all of that “monkey clicking stuff”

Yes you are smart programmer but seems that you aren't a **lazy Drupal developer ;)**

Most probably you will end with:

- **complex urls**
- **many combinations of arguments**
- **a lot of glue code**
- **a lot of hardcoded stuff**



Wait I'm a lazy Drupal developer but I've chosen Context and contrib modules for it

Yes, seems that you are a **lazy Drupal developer** but did you had those problems:

- **again complex urls** full of arguments and parameters
- **famous excuse to enable "few additional"** contrib modules
- **exporting settings to Features**
- **again many combinations of passed arguments and parameters**
- well, you still need to use blocks
- still hard to reuse in another projects
- yes true, but together with Display Suite I can do more ...
- ... (flame war begins)



What instead?

Let's take a
look at CTools
plugins



What is CTools?

Acronym for Chaos Tools

This suite is primarily a set of APIs and tools to improve the developer experience. It also contains a module called the Page Manager whose job is to manage pages. **All these enhancements have grown far more than just Panels.**

<https://www.drupal.org/project/ctools>



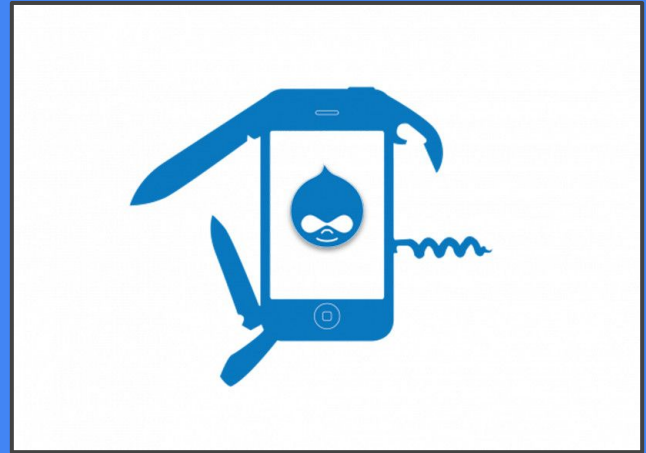
Ok, show me why this heavy and complex Panels/CTools are better?

- **it is not about Panels** but about Ctools plugins in Panels ecosystem
- **plugins concept** - allowing you to create more generic solutions
- Some similarities with **MVC like model** - (little bit of fantasy and interpretation)
- **Separation of logic, content and presentation layers**
- if you hate Panels you don't need to use them (small problem with rendering part which you need to do by yourself)
- **easy way of building plugins** (for most of them you need to have definition array and one main custom callback function)
- possibility to overwrite default Drupal endpoints (like node/%node)
- **powerful integration with Views** -
- **easy way of controlling access** via access plugins + one entry point concept

Ok, enough of talking. Time for Sitebuilder DEMO



**Is it enough ?
No, of course
not :)**



Custom CTools plugins = Ultimate powers

You can:

- build your own reusable plugins
- extend existing plugins
- check implementations in contrib modules
- check provided examples

Let's talk about types of those plugins



Argument plugin

Get a value from the URL, and use it to create a context (data model)



Context Plugin (do not confuse with Context module)

Provide custom data to other elements on the page.

Term **context** refers to a kind of wrapper around any meaningful object. The following contexts are embedded in the Panel module by default:

Comment, File, Node, Node add form, Node edit form, String, Taxonomy term, Taxonomy vocabulary, Token, User, User edit form.



Access plugin / Selection rules

Get **TRUE** or **FALSE** based on given contexts and configuration



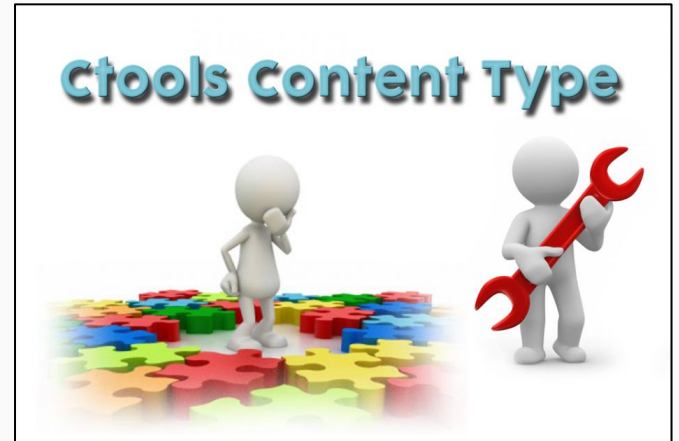
Relationship Plugin

Get another context
based on the other one.



Content type (yes, another kind of type)

Returns some markup which can be based on context data



Time for Developers Demo



Ctools pluggable caching mechanism

Panels can be cached as a whole, meaning the entire output of the panel can be cached, or individual content panes that are heavy can be cached.

There are many solutions (dozen contrib modules and build in options) to speed up a thing which is already faster than regular Drupal block system.

<https://docs.acquia.com/articles/panels-caching>

Ctools custom layout plugin

[Creating a custom Ctools layout plugin](#) for Drupal is actually quite easy. You define your layout in a .info file, create the html structure in a template (*.tpl.php), style it in a stylesheet (*.css or *.scss) and provide a thumbnail image so the site administrator has an idea of what it looks like. Each of these files follows similar but slightly different naming conventions.

Ctools with Fences module

Using Fences, we can gain complete control over fields. Using the Theme's templates, we can strip out any unwanted HTML inserted by Drupal core. Page Manager allows us to define layouts, and what content gets loaded gives us complete control over the page. Using this system, we can:

- completely separate design from content and layout
- easily create variations
- configure customizations rapidly

CTools in Drupal 8

- CTools Object cache - became foundation for TempStore
- CTools Export - became foundation for D8 CMI
- CTools Export UI - became foundation for Config Entities
- CTools Plugins - became foundation for D8 plugins
- CTools Ajax & Modal - in D8 Core
- CTools Content Plugins - became D8 Core blocks
- CTools Access Plugins - became D8 Core Conditions
- [Possibility to write D7 plugins in D8 oo style](#)



Resources:

- <http://www.slideshare.net/agentolivia/ctools-styles>
- <http://www.sitepoint.com/building-custom-ctools-plugins-in-drupal-7>
- <http://jimhir.ch/blog/using-fences-page-manager-optimize-html-markup-drupal-7>
- <http://2015.pnwdrupalsummit.org/2015/sessions/ctools-plugins-and-you>
- <https://www.previousnext.com.au/blog/drupal-8-now-object-oriented-plugins-drupal-7>
- <http://atendesigngroup.com/blog/creating-custom-ctools-layout-plugins-yeoman>
- <http://www.stevector.com/plugin-haikus/#1>
- <http://internetdevels.com/blog/ctools-context-plugin>
- <http://morpht.com/posts/how-create-ctools-content-types-drupal-7>
- <https://drupalize.me/videos/overview-page-manager-series?p=1147>
- <https://drupalize.me/videos/introduction-ctools-style-plugins?p=1691>



Questions ?

Tell me your story.



The end

Thanks everyone !



AMPLEXOR

Pure Content Management

