Neilos

Version 1.4

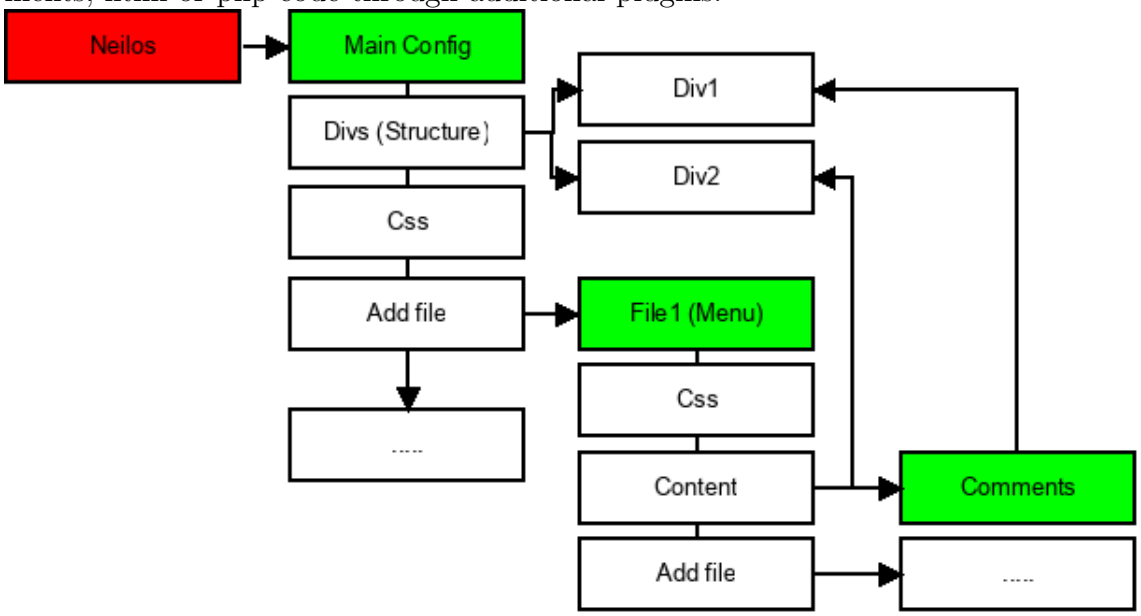
Contents

1 Introduction

Neilos is a content management system based almost completely on javascript. It is designed to be lightweight and fast and to use ajax for content loading. Furthermore, it is possible to use mysql, xml or both technologies to store content datas. Building a website with Neilos should be as simple as writing a basic xml file.

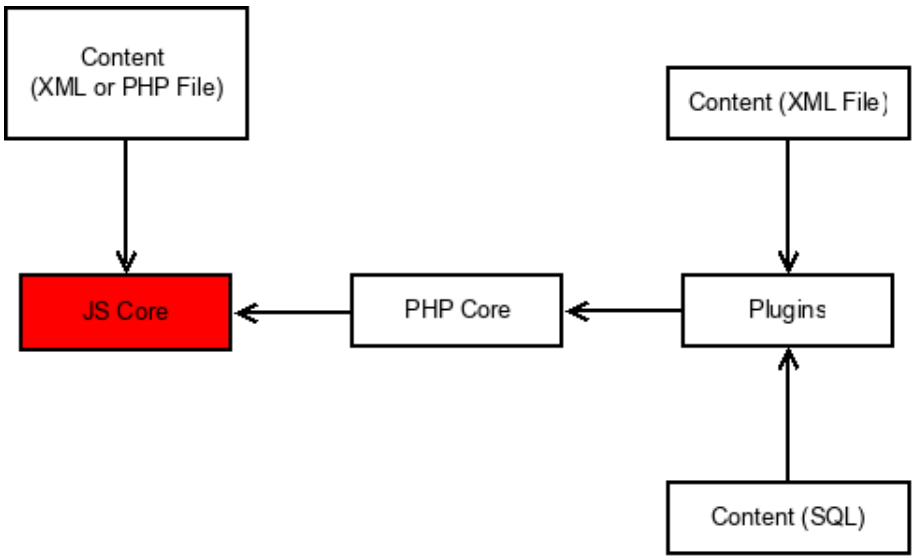
2 Workflow of Neilos

Neilos content and information consist and travel from the server to the client through entries. Entries can be stored in files or in a database, can have subentries or can load dinamically other entries.
The first thing Neilos does is loading resources/xml/config.xml, which is the main configuration of the site (it is an entry itself!). The main config usually (but not mandatory) loads the css styles, add the layout divs (see Structure section), and sets global variables, like default DOM target or default animation.
Finally, the main config loads other external entries, which can add articles, comments, html or php code through additional plugins.



2.1 Neilos Structure

Nailos consists of 2 main blocks, the client-side javascript core (neilos.js) and the server-side php core (load.php).
Only the former is strictly necessary, the php core can be ignored in the simplest neilos configuration.



3 XML Entry Anatomy

Every XML file is loaded with an *entryid* parameter. Neilos searches for the entry with the id supplied.
An entry consists of 2 main sections: config and content. There can be also subentries, which are listed inside the parent entry and outside its config and content sections.
Content should consists of pure html, since it will be dumped to the DOM.

```
<entry id='entryid'>
  <config>
    ...
  </config>
  <content>
    ...
  </content>

  <!-- Subentries -->
  <entry id='..'></entry>
  <entry id='..'></entry>
</entry>
```

3.1 DOM

By default, every entry is added to the DOM with a main div (*#id_entry*) and 3 subdivs: .title, .content and .comments.
This behaviour can eventually be changed with some options (see below). If both title and content do not exist, the entry is not added to the DOM.

3.2 Config

The config tag is useful to load additionals css files, entries or to set useful settings. Almost all options are inheritable to subentries. Available options are:

- <pagetitle>

Change the title of the page

- `<title>`

Set the title of the entry

- `<css id='css_id'>`

Load an external css file. The id is used to identify the css in order to remove or replace it afterwards.

- `<author>m3l7</author>`
  `<date>2011-09-08</date>`

Specify the date and author of the entry. They will be added to the DOM in the .title before everything else.

- `<structure_div overwrite='false' parent="#parent">div_name</structure_div>`

Add a div to the DOM. (See Structure section)

- `<target>target</target>`

specify where the entry should be added in the DOM (jquery selector).

- `<animation speed='fast' speedshow='normal' speedhide='slow'`
  `type='fade/slide/hide'>enabled</animation>`

Select the type and speed of the animation. If speedshow/speedhide is set, it will be used instead of speed for showing/hiding the entry. The speed can have the format $\#n$, where n is the animation time in milliseconds.

- `<home>#home.xml</home>`

Set the default home page. See links section for informations about links meaning.

- `<display entries="showfirst/show/hide"></display>`

Choose if content (and comments) should be visible or not. If showfirst is set, the entry is visible only if it's the first child.
Default: show

- `<load_file mode='loadwhenshown'`
  `entryid="menu">resources/xml/menu.xml</load_file>`

Load an external file. Neilos will add the content inside the jquery selector tag.
If mode='loadwhenshown' is set, the file is loaded only when the entry is toggled.

- `<clear>false</clear>`

If true, the target will be cleared before adding the entry.

- `<visibiity>true</visibility>`

If true, the entry will not be shown.

- `<skipsubentries>true</skipsubentries>`

If true, subentries will be ignored.

- `<type>notitle</type>`

If set, title will be ignored.

- `<default_extension>xml</default_extension>`

Set the default extension of content files. Can be xml or php and can be set only in the main config (config.xml)
Neilos will cycle throigh the other extensions automatically if the file is not found.

- `<class>optionalclass</class>`

Add an additional class to the entry.

- `<click prevent_default='true' updateurl='true' href='filename' remove='true'></c`

Open an internal link (specified by href) when the entry is clicked. If prevent_default is set, the normal toggle behaviour is avoided.If updateurl is set true, the url bar will be updated to the href link.
If remove is set, the entry is removed after being clicked.

- `<alias source='page' file='filename.php'></alias>`

Open filename.php when #page link is requested

- `<tag>tag1,tag2</tag>`

Attach a tag to the entry. Tags are useful to load dinamically only certain "categories" of entries. Currently, tags are only supported when using the php core.

- `<plugin></plugin>`

Add a plugin to the entry. See the proper section.

- `<updateurl>#!url</updateurl>`

Update the url bar to the specified url.

- `<delete id='id'></delete>`

Delete the specified entry

- `<checkform selector='#form' href='#!link'/>`

  Check the specified form, and pass its input boxes with check='true' property to href as an additional parameter ($\&param1 =' input'$).
  This is only a proof of concept. Should be extended in the future.

3.3 Link to entries

It is possible to specify just the name of the file in which the entry is stored. In this case the php core will not be used.

```
<load_file entryid="menu">
resources/xml/menu.xml</load_file>
```

To use the php core, use this syntax:

```
<load_file entryid="menu"> resources/load.php?data="filename"
&id="menu"&tag="tag"&sql="true"&file="false"</load_file>
```

The data tag is the XML filename where the entry is stored. If a database is used for storage, data will be ignored or will be used as id if id parameter is not set.
sql and file parameters are used to override the default behaviour

3.4 Menu links

A typical menu link points to index.php#!page.xml .Neilos will load automatically page.xml and will add it to the site with ajax, using the file name as id. The file must be located in resources/content/.
If the file is not found and no extension is provided, Neilos will try to add .xml and .php extensions to the file name.
When using the php core, it is possible to use these simplified syntax:

```
index.php#!load.php?parameters
```

instead of

```
index.php#!resources/load.php?parameters
```

and

```
index.php#!load/home?parameters
```

instead of

```
index.php#!resources/load.php?data="home"&parameters
```

4 Variables

4.1 Server (PHP Core) variables

Server variables are stored in vars.php

- `$_SITENAME = 'n';`

  Name of the site. Mandatory.

- ```
  $_ANALYTICS = FALSE;
  $_ANALYTICS_ID = 'xxx';
  $_STATCOUNTER = FALSE;
  $_STATCOUNTER_PROJ = 'xxx';
  $_STATCOUNTER_SEC = 'xxx';
  ```

  Stat counter and Google Analytics parameters.

- `$_SITENAME = 'n';`

- ```
  $_DEFAULT_STORAGE = 'sql';
  $_DEFAULT_STORAGE = 'file';
  ```

  Select the default storage.

- ```
  $_SQL_HOST = "xxx";
  $_SQL_USER = "xxx";
  $_SQL_PASS = "xxx";
  $_SQL_DB = "xxx";
  ```

  MySQL parameters.

4.2 Client (JS Core) variables

Neilos has an internal parser which substitutes some special strings inside content entries. At this stage, it's very limited.

- `$_version`

  show Neilos version

5 Structure

Structure objects can be created using the following methods:

5.1 js methods

- Neilos.Structure.new_div: this will create a simple div. Additional classes can be added.
- Neilos.Structure.new_tab: this will create an entry (**TODO** Rename to new_entry?). An entry is a div with 2 or 3 subdivs: [title], content, comments. Each subdivs has 2 subdivs: _text and _text_right
  ```

5.2 xml methods

- `<structure_div overwrite='false' parent="#parent">`

  create a div and a subdiv _content. It is created at the end of the current DOM (or inside parent, if specified), so it should be used only in the main config. It must be placed inside <config> tag.

- `<structure_tab parent="#parent">`

  create a standard container for an entry

6 SEO

Starting from version 1.2.2, Neilos can be indicized by Google, using the standard _escaped_fragment method.
A *fallback_load.php* and a modified index.php are provided, which can load the site using php only code. At the moment, *fallback_load.php* loads all the content, but generally not in the correct order and with the correct containers; for this readon it is not suitable for every day use as a replacement for the default js code.

7 Plugins

Starting from version 1.3, Neilos supports a server side plugin architecture.
To load a plugin, just add the "plugin" tag to the config section, i.e.

```
<plugin>
        <load plugin="More">
            <param name="entries_per_page">1</param>
            <param name="show_more">1</param>
        </load>
</plugin>
```

Dev Version supports a basic client side plugin architecture. To load a js (client side) plugin, use

```
<plugin type='js'>name</plugin>
```

7.1 List of plugins

7.1.1 More (Server)

More plugin prevents neilos to load all subentries of a given entry if they are too many.
It is possible to specify how many subentries have to be shown at the same time, and choose to show a more button for showing the next page.
Parameters:
-entries_per_page: number of entries to display in a single 'page'.
-show_more: if set to 1, a button which loads the next page will be visible.

7.1.2 Loader (Server)

Load entries and add them as subentries using the php core. This is different from the standard $load\_file$ situation where the entries are loaded separately by the js core.
This is useful when additional entries have to be loaded as subentries, i.e. to be postprocessed by other plugins.
Parameters:
Same as the standard $load\_file$ GET parameters of the php core. (i.e. data,tag,sql,file)

8 Admininstration

Starting from 1.4, Neilos has a very poor admin interface. At the moment, it's only possible to convert all the content xml files to the database and back, and to clean the database. There is no support for login or multiuser, so it's not safe to publish the admin/ folder.