



Theme Documentation - Basics


 **Dillon** included in  documentation
06.03.2020 4694 words 23 minutes



Discover what the Hugo - **Lovelt** theme is all about and the core-concepts behind it.

1 Requirements



Thanks to the simplicity of Hugo, **Hugo** is the only dependency of this theme.

Just install latest version of  **Hugo** (> 0.62.0) for your OS (**Windows**, **Linux**, **macOS**).



Why not support earlier versions of Hugo?

Since **Markdown Render Hooks** was introduced in the **Hugo Christmas Edition**, this theme only supports Hugo versions above **0.62.0**.

Hugo extended version is recommended

Since ~~some features of this theme depend on processes~~  **SCSS** to  **CSS**, it is recommended to use Hugo **extended** version above **0.62.0**.

Hugo extended version is recommended

Since some features of this theme need to processes  **SCSS** to  **CSS**, it is recommended to use Hugo **extended** version for better experience.

2 Installation

The following steps are here to help you initialize your new website. If you don't know Hugo at all, we strongly suggest you learn more about it by following this [great documentation for beginners](#).


| 2.1 Create Your Project

Hugo provides a **new** command to create a new website:

```
Bash  
1 hugo new site my_website  
2 cd my_website
```

| 2.2 Install the Theme

The **Lovelt** theme's repository is: <https://github.com/dillonzq/Lovelt>.

You can download the [latest release](#)  **zip** file of the theme and extract it in the **themes** directory.

Alternatively, clone this repository to the **themes** directory:

```
Bash  
1 git clone https://github.com/dillonzq/Lovelt.git themes/Lovelt
```

Or, create an empty git repository and make this repository a submodule of your site directory:

```
Bash  
1 git init  
2 git submodule add https://github.com/dillonzq/Lovelt.git themes/Lovelt
```

| 2.3 Basic Configuration

The following is a basic configuration for the Lovelt theme:

TOML

Note

When building the website, you can set a theme by using **--theme** option. However, we suggest you modify the configuration file (**config.toml**) and set the theme as the default.

| 2.4 Create Your First Post

Here is the way to create your first post:

```
Bash  
1 hugo new posts/first_post.md
```

Feel free to edit the post file by adding some sample content and replacing the title value in the beginning of the file.

Note

By default all posts and pages are created as a draft. If you want to render these pages, remove the property **draft: true** from the metadata, set the property **draft: false** or add **-D / --buildDrafts** parameter to **hugo** command.

| 2.5 Launching the Website Locally

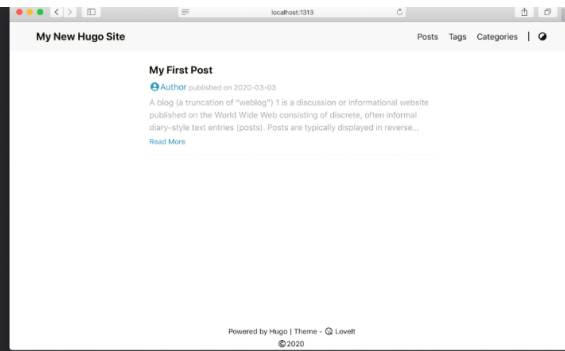
Launch by using the following command:

```
Bash  
1 hugo serve
```

Go to <http://localhost:1313>.

CONTENTS

- | 1 Requirements
- | 2 Installation
 - | 2.1 Create Your Project
 - | 2.2 Install the Theme
 - | 2.3 Basic Configuration
 - | 2.4 Create Your First Post
 - | 2.5 Launching the Website Locally
 - | 2.6 Build the Website
- | 3 Configuration
 - | 3.1 Site Configuration
 - | 3.2 Favicons, Browserconfig, Manifest
 - | 3.3 Style Customization
- | 4 Multilingual and i18n
 - | 4.1 Compatibility
 - | 4.2 Basic Configuration
 - | 4.3 Overwrite Translation Strings
- | 5 Search
 - | 5.1 Output Configuration
 - | 5.2 Search Configuration



Basic configuration preview

Tip

When you run `hugo serve`, when the contents of the files change, the page automatically refreshes with the changes.

Note

Since the theme use `.scratch` in Hugo to implement some features, it is highly recommended that you add `--disableFastRender` parameter to `hugo server` command for the live preview of the page you are editing.

Bash

```
hugo serve --disableFastRender
```

2.6 Build the Website

When your site is ready to deploy, run the following command:

Bash

```
hugo
```

A `public` folder will be generated, containing all static content and assets for your website. It can now be deployed on any web server.

Tip

The website can be automatically published and hosted with [Netlify](#) (Read more about [Automated HUGO deployments with Netlify](#)). Alternatively, you can use [AWS Amplify](#), [Github pages](#), [Render](#) and more...

3 Configuration

3.1 Site Configuration

In addition to [Hugo global configuration](#) and [menu configuration](#), **Lovell** lets you define the following parameters in your site configuration (here is a `config.toml`, whose values are default).

Please open the code block below to view the complete sample configuration 📄:

TOML

Note

Note that some of these parameters are explained in details in other sections of this documentation.

Hugo environments

Default environments are `development` with `hugo serve` and `production` with `hugo`.

Due to limitations in the local `development` environment, the **comment system**, **CDN** and **fingerprint** will not be enabled in the `development` environment.

You could enable these features with `hugo serve -e production`.

Tips about CDN Configuration

CHANGED 0.2.7

TOML

```
[params.cdn]
1 # CDN data file name, disabled by default
2
3 # ("jsdelivr.yml")
4
data = ""
```

The default CDN data file is located in `themes/LoveIt/assets/data/cdn/` directory. You can store your own data file in the same path under your project: `assets/data/cdn/`.

Tips about social Configuration

NEW 0.2.0

You can directly set your ID to get a default social link and its icon:

TOML

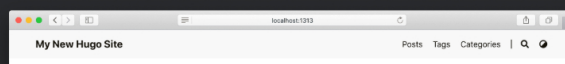
```
[params.social]
1 Mastodon = "@xxxx"
```

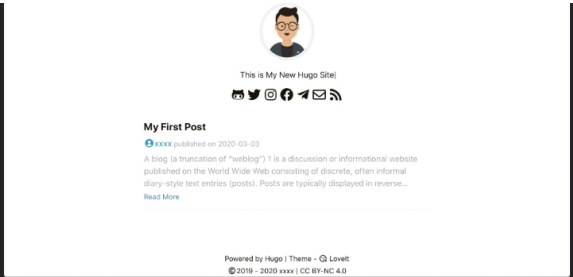
The social link generated is `https://mastodon.technology/@xxxx`.

Or You can set more options through a dict:

TOML

The default data of all supported social links is located in `themes/LoveIt/assets/data/social.yml`, which is you can refer to.





Complete configuration preview

3.2 Favicons, Browserconfig, Manifest

It is recommended to put your own favicons:


- apple-touch-icon.png (180x180)
- favicon-32x32.png (32x32)
- favicon-16x16.png (16x16)
- mstile-150x150.png (150x150)
- android-chrome-192x192.png (192x192)
- android-chrome-512x512.png (512x512)

into `/static`. They're easily created via <https://realfavicongenerator.net/>.

Customize `browserconfig.xml` and `site.webmanifest` to set theme-color and background-color.

3.3 Style Customization

CHANGED 0.2.8

 Note

Hugo **extended** version is necessary for the style customization.

Lovett theme has been built to be as configurable as possible by defining custom `.scss` style files.

The directory including the custom `.scss` style files is `assets/css` relative to **your project root directory**.

In `assets/css/_override.scss`, you can override the variables in `themes/Lovett/assets/css/_variables.scss` to customize the style.

Here is an example:

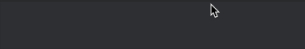
```
1 @import url('https://fonts.googleapis.com/css?family=Fira+Mono:400,700&display=swap&subset=');
2 $code-font-family: Fira Mono, Source Code Pro, Menlo, Consolas, Monaco, monospace;
```

In `assets/css/_custom.scss`, you can add some css style code to customize the style.

4 Multilingual and i18n

Lovett theme is fully compatible with Hugo multilingual mode, which provides in-browser language switching.

Docs About English > 🔍



Language Switch

4.1 Compatibility

CHANGED 0.2.10

Language	Hugo Code	HTML Lang Attribute	Theme Docs	Lunr.js Support
English	en	en	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Simplified Chinese	zh-cn	zh-CN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
French	fr	fr	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Polish	pl	pl	<input type="checkbox"/>	<input type="checkbox"/>
Brazilian Portuguese	pt-br	pt-BR	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Italian	it	it	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Spanish	es	es	<input type="checkbox"/>	<input checked="" type="checkbox"/>
German	de	de	<input type="checkbox"/>	<input checked="" type="checkbox"/>
German	de	de	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Serbian	sr	sr	<input type="checkbox"/>	<input type="checkbox"/>
Russian	ru	ru	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Romanian	ro	ro	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Vietnamese	vi	vi	<input type="checkbox"/>	<input checked="" type="checkbox"/>

4.2 Basic Configuration

After learning [how Hugo handle multilingual websites](#), define your languages in your [site configuration](#).


For example with English, Chinese and French website:

> TOML


Then, for each new page, append the language code to the file name.

Single file `my-page.md` is split in three files:

- in English: `my-page.en.md`
- in Chinese: `my-page.zh-cn.md`
- in French: `my-page.fr.md`

 Note

Be aware that only translated pages are displayed in menu. It's not replaced with default language content.

 Tip

Use Front Matter parameter to translate urls too.

4.3 Overwrite Translation Strings

Translations strings are used for common default values used in the theme. Translations are available in [some languages](#), but you may use another language or want to override default values.

To override these values, create a new file in your local i18n folder `i18n/<LanguageCode>.toml` and inspire yourself from `themes/LoveIt/i18n/en.toml`.

By the way, as these translations could be used by other people, please take the time to propose a translation by [making a PR](#) to the theme!

5 Search

NEW 0.2.0

Based on Lunr.js or algolia, searching is supported in **Lovelt** theme.

5.1 Output Configuration

In order to generate `index.json` for searching, add `JSON` output file type to the `home` of the `outputs` part in your [site configuration](#).

```
1 [outputs]
2   home = ["HTML", "RSS", "JSON"]
```

5.2 Search Configuration

Based on `index.json` generated by Hugo, you could activate searching.

Here is the search configuration in your [site configuration](#):

TOML

```
1 [outputs]
2   home = ["HTML", "RSS", "JSON"]
```

How to choose search engine?

The following is a comparison of two search engines:

- lunr** : simple, no need to synchronize `index.json`, no limit for `contentLength`, but high bandwidth and low performance (Especially for Chinese which needs a large segmentit library)
- algolia** : high performance and low bandwidth, but need to synchronize `index.json` and limit for `contentLength`

NEW 0.2.0 The content of the post is separated by `h2` and `h3` HTML tag to improve query performance and basically implement full-text search. `contentLength` is used to limit the max index length of the part starting with `h2` and `h3` HTML tag.

Tips about algolia

You need to upload `index.json` files to algolia to activate searching. You could upload the `index.json` files by browsers but a CLI tool may be better. **Algolia Atomic** is a good choice. To be compatible with Hugo multilingual mode, you need to upload different `index.json` for each language to the different index of algolia, such as `zh-cn/index.json` or `fr/index.json` ...