divider

Installation

It takes just a few minutes to get up and running with Shokunin Static Site Gene rator (SSG).

To install Shokunin Static Site Generator (SSG), you need to have the Rust toolc hain installed on your machine. You can install the Rust toolchain by following the instructions on the Rust website.

Once you have the Rust toolchain installed, you can install Shokunin Static Site Generator (SSG) using the following command:

cargo install ssgFor simplicity, we have given Shokunin Static Site Generator (S SG) a simple alias

which can stand for

or

.

You can then run the help command to see the available options and commands:

ssg --help Requirements

The minimum supported Rust toolchain version is currently Rust1.72.0 or later (s table). It is recommended that you install the

latest stable version of Rust.

Platform support

Shokunin Static Site Generator (SSG) is supported and tested on the following platforms and architectures as part of our CI/CD pipeline.

This list is based on the Rust Platform Support list.

The GitHub Actions shows the platforms in which the Shokunin Static Site Generat or (SSG) library tests are run.

Documentation

Info:You can find our documentation on docs.rs, lib.rs and crates.io.

Usage

Command Line Interface (CLI)

The Shokunin Static Site Generator (SSG) library runs in a Terminal window and c
an be used to generate a static website.

Here’s the first command you can enter in your Terminal window to run Shokunin S
tatic Site Generator (SSG):

ssg --new=docs --content=content --template=template --output=output --serve=pu
blicor

ssg -n=docs -c=content -t=template -o=output -s=publicThis command will create
a new website with the name

in the current directory. It will use the

directory to gather the website content and the

directory to generate the website files. It will serve the website directly from

the

directory.

Arguments

,

The name of the new website. (required),

The directory containing the website content. (required),

The directory containing the website templates. (required),

The directory where the generated website files will be saved temporarily. (requ
ired),

Run the development server. (optional). The directory from which the website wil

l be served. In your project

To use the Shokunin Static Site Generator (SSG) library in your project, add the following to your

file:

[dependencies]

shokunin = "0.0.26"Add the following to your

file:

extern crate ssg;

use ssg::*;then you can use the Shokunin Static Site Generator (SSG) functions i n your application code.

Examples

To get started with Shokunin Static Site Generator (SSG), you can use the exampl es provided in the

directory of the project.

To run the examples, clone the repository and run the following command in your terminal from the project root directory.

cargo run --example exampleThe command will generate a static website based on t he configuration details in the

directory.

use ssg::compiler::compile;

use std::path::Path;

fn main() -> Result<(), Box<dyn std::error::Error>> {

// Define the paths to the build, site, content and template directories.

let build_path = Path::new("examples/example.com/build");

let content_path = Path::new("examples/example.com/content");

```
let site_path = Path::new("examples/example.com/public");
let template_path = Path::new("examples/example.com/template");
compile(build_path, content_path, site_path, template_path)?;
Ok(())
}
```

The main() function in this code compiles a website from the
directory, using the
directory to generate the website files. The compiled website is saved in the
directory and served directly from the
directory.

Args

The path to the directory where the compiled website will be saved. The path to the directory containing the website content. The path to the directory where th e generated website files will be served from. The path to the directory contain ing the website templates. Quick Start

You are now ready to use Shokunin and create amazing websites!