Building Websites with Jekyll

Rajesh Kumar
kr.rajesh.phy@gmail.com

January 1, 2024

Introduction

Jekyll is a powerful static site generator that simplifies the process of building and maintaining websites. It's a popular choice for developers who want a fast and efficient way to create static websites or blogs without the complexity of a traditional content management system.

In this article, we'll explore the basics of Jekyll and how you can use it to develop your website.

Getting Started with Jekyll

Installation

Before you start using Jekyll, you need to have Ruby installed on your system. Once Ruby is installed, you can install Jekyll using the following command:

```
gem install jekyll
```

Creating a New Jekyll Site

To create a new Jekyll site, navigate to the desired directory and run the following command:

```
jekyll new mywebsite
```

This will generate a new Jekyll site in the "mywebsite" directory.

Understanding the Jekyll Directory Structure

Jekyll follows a specific directory structure. Here are some key directories and files:

- **_layouts**: Contains templates for different pages.
- **_posts**: Stores your blog posts in Markdown format.
- **_config.yml**: Configuration file for your Jekyll site.
- **index.html**: The main page of your site.

Creating Content

To add a new blog post, create a new Markdown file in the _posts directory. Jekyll will automatically convert these files into HTML.

Customizing Your Site

Jekyll allows you to customize your site's appearance and functionality easily. You can modify the _config.yml file to change settings such as the site title, description, and more.

Building and Previewing Your Site

To build and preview your site locally, run the following commands:

```
cd mywebsite
jekyll serve
```

This will start a local server, and you can view your site by navigating to `http://localhost:4000` in your web browser.

Conclusion

Jekyll is a fantastic tool for website development, offering simplicity and flexibility. Whether you're creating a personal blog or a portfolio site, Jekyll can streamline the process and help you focus on creating great content.

In future posts, we'll explore advanced features of Jekyll and how to enhance your site further.

Happy coding!

Practical Session

How to install Jekyll in Linux

1. `sudo apt-get update`
2. `sudo apt-get install ruby-full build-essential zlib1g-dev`
3. `echo '# Install Ruby Gems to ~/gems' >> ~/.bashrc`
4. `echo 'export GEM_HOME="~/gems"' >> ~/.bashrc`
5. `echo 'export PATH="~/gems/bin:$PATH"' >> ~/.bashrc`
6. `source ~/.bashrc`
7. `gem install jekyll bundler`
8. `jekyll new my-awesome-site`

How to install Jekyll on macOS

1. First, check if gem is installed or not. Type the following command to check if gem is installed or not:

   ```
   sudo gem install jekyll
   ```

2. If gem is not installed, then install it using this command:

   ```
   sudo gem install jekyll
   ```

3. Ensure the gem directory is in the path. Add the following line to your shell profile file (e.g., /.bashrc or /.zshrc):

   ```
   export PATH="$PATH:$(ruby -e 'puts Gem.user_dir')/bin"
   ```

4. Install Jekyll and Bundler using the following command:

   ```
   gem install bundler
   ```

5. Install Jekyll Dependences using the following command:

   ```
   bundle install
   ```

6. Create a new Jekyll site at ./myblog:

   ```
   jekyll new myblog
   ```

How to install Jekyll on Windows

1. Install Ruby+Devkit 2.6.X (x64) from RubyInstaller Downloads. Choose the default options for installation.

2. Run the ridk install step from the last stage of the RubyInstaller installation. This is needed for installing gems with native extensions. From the command prompt:

   ```
   ridk install
   ```

3. Install Jekyll and Bundler using the following command:

   ```
   gem install jekyll bundler
   ```

4. Create a new Jekyll site at ./myblog:

   ```
   jekyll new myblog
   ```

5. Change into your new directory:

   ```
   cd myblog
   ```

6. Build the site and make it available on a local server:

   ```
   bundle exec jekyll serve
   ```

7. Browse to `http://localhost:4000`

8. Add the following line to your shell profile file (e.g., /.bashrc or /.zshrc):

   ```
   export PATH="$PATH:$(ruby -e 'puts Gem.user_dir')/bin"
   ```