Meta Tag Generator Angular App Documentation

Overview

The Meta Tag Generator Angular App allows users to easily generate meta tags for their web pages by providing a few simple inputs. This app automates the process of creating essential meta tags, such as title, description, keywords, and Open Graph tags, to improve SEO and social media sharing.

Features

- User-friendly interface to input metadata.
- Supports standard and Open Graph meta tags.
- Angular signals for seamless data sharing and management.

Installation

- Node.js and npm installed
    - Go to the Node.js download page.
    - Download the installer for your operating system (Windows, macOS, or Linux).
    - Run the installer and follow the prompts. The installer will also install npm (Node Package Manager).
    -
- Angular CLI installed
    - npm install -g @angular/cli
- Angular CDK installed
    - npm install @angular/cdk

Usage

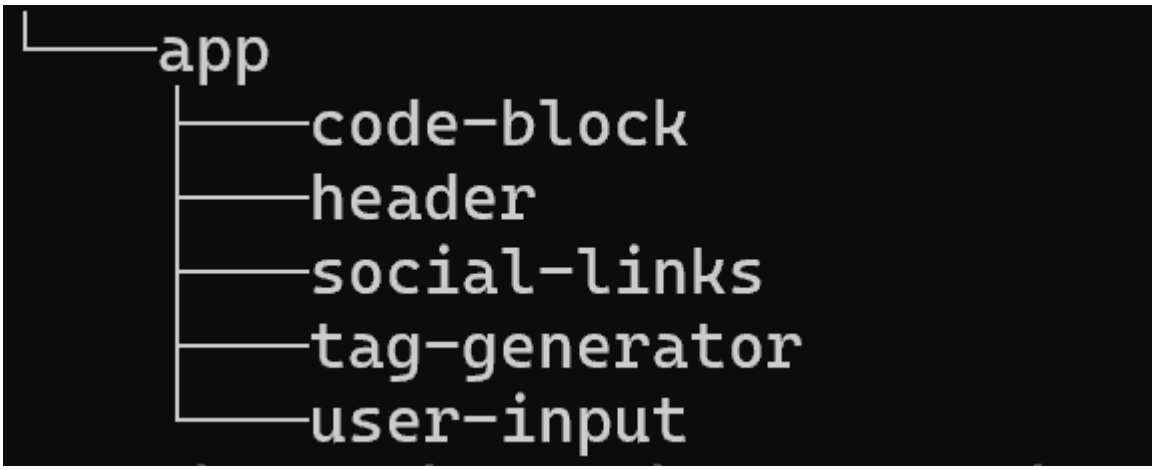
1. **Enter Metadata:**
    - Fill out the input fields for the page title, description, keywords, and other optional metadata (e.g., image URL for Open Graph).
2. **Press Generate Button:**
    - A list of generated meta tags is displayed.
3. **Copy Meta Tags:**
    - Use the "Copy" button to copy the generated meta tags for use in your project.

Architecture

Component Structure

- `AppComponent`: Root component managing overall structure.
- `UserInputComponent`: Central hub for user input and state management.
- `TagGeneratorComponent`: Handles meta tag creation.
- `SocialLinksComponent`: Manages social platform-specific metadata.
- `CodeBlockComponent`: Handles the display and copy functionality.

Folder Structure



Key Functionalities

Dynamic Meta Tag Generation

The app dynamically generates meta tags based on user input and social platform selection. This is managed through Angular's `@Input` properties, signals, and computed properties.

- **Key Methodology:**
  - `TagGeneratorComponent` listens for form data and social platform configurations passed as inputs.
  - Generates basic and social media-specific meta tags dynamically.
  - Uses `computed` to create a real-time view of the meta tags.

```
generatedTags = computed(() => {
  const tags = [];
  const formData = this.formData();
  const socialLinks = this.socialLinks();

  // Basic meta tags
  tags.push(`<title>${formData.title}</title>`);
  tags.push(`<meta name="viewport" content="width=device-width, initial-scale=1.0">`);
  tags.push(`<meta charset="${formData.charset}">`);
  tags.push(`<meta name="author" content="${formData.author}">`);
  tags.push(`<meta name="description" content="${formData.description}">`);
  tags.push(`<meta name="keywords" content="${formData.keywords}">`);

  // Social platform-specific meta tags
  socialLinks.forEach(platform => {
    if (platform.name === 'Facebook') {
      tags.push(`<meta property="og:title" content="${platform.fields.title}">`);
      tags.push(`<meta property="og:description" content="${platform.fields.description}">`);
      tags.push(`<meta property="og:image" content="${platform.fields.image}">`);
    } else {
      tags.push(`<meta name="${platform.name.toLowerCase()}:title" content="${platform.fields.title}">`);
      tags.push(`<meta name="${platform.name.toLowerCase()}:description" content="${platform.fields.description}">`);
      tags.push(`<meta name="${platform.name.toLowerCase()}:image" content="${platform.fields.image}">`);
    }
  });

  return tags.join('\n');
});
```

Real-Time Input Management

User input and social link configurations are handled dynamically using Angular signals and event bindings.

- **Form Input Management (`UserInputComponent`):**
  - Stores metadata in a signal.
  - Updates values dynamically using the `updateFormField` and `handleInput` methods.

```
updateFormField(field: string, value: string): void {
  this.formData.set({
    ...this.formData(),
    [field]: value
  });
}
```

Social Platform Management (`SocialLinksComponent`):

- Toggles platform selection and updates metadata for each selected platform.
- Emits updated data via `@Output` to the parent component.

```
togglePlatform(platformName: string): void {
  const updatedPlatforms = this.socialPlatforms().map(platform =>
    platform.name === platformName
      ? { ...platform, selected: !platform.selected }
      : platform
  );
  this.socialPlatforms.set(updatedPlatforms);
```

Meta Tag Preview and Copy

- The `CodeBlockComponent` displays the generated meta tags and allows users to copy them to the clipboard.
- **Features:**
  - Auto-updates to show the latest generated meta tags.
  - Provides a "Copy to Clipboard" feature.

```
copyCode() {
  this.clipboard.copy(this.code);
}
```

Component Communication

- `UserInputComponent` handles the top-level state of form data and selected social links.
- Passes data to:
    - `TagGeneratorComponent` for meta tag generation.
    - `SocialLinksComponent` for social platform configurations.
- Communicates changes using signals and `@Output`.

Related Links

- GitHub repository
- Finished Application(coming soon)