sheffercool / **PinegrowDevelopersDocumentation**

forked from Pinegrow/PinegrowDevelopersDocumentation

Private site for creation of Pinegrow Web Editor API documentation

0 stars 1 fork

Star Watch

Code Pull requests Actions Projects Wiki Security Insights Settings

master

This branch is even with Pinegrow:master.

Pull request Compare

matjaztrontelj ... on 5 Mar

View code

README.md

API Introduction

The Pinegrow Web Editor is a tool for building responsive websites faster, with clean, non-opinionated code. The Pinegrow App can also be used as a CMS, allowing rapid templating of web pages that allow the end user to modify only selected content. Under the hood, Pinegrow is built with an SDK version of the NW.js App builder, around Node.js and the Chromium engine. This makes the Pinegrow App highly customizable and developer friendly. Plugins are created in JavaScript and programmers can take advantage of access to a number of Node.js modules and the Chromium Developer's Tools.

Extensions of the Pinegrow Web App are delivered through a main framework function. This framework exposes event hooks and functions that allow for manipulation of App content and detection of App state.

Content is primarily added to three areas:

1. The Library panel- this section of the App typically contains reusable HTML snippets that can be dragged to the main DOM page to visually build web pages. This library can

also contain other items, such as scripts.

2. The Properties panel - this section of the App contains controls that modify the on page code, such as adding a class or changing the src for an <image>.
3. The Actions panel - this section of the App is used both for modifying the on page code, but also for adding Pinegrow-specific code, such as defining locked areas on a CMS page.

In addition to adding content, the plugin API also allows for adding additional controls to Pinegrow, for example, a utility to minimize the active stylesheet on page save.

Common Use Cases

- Add components to the Library panel
- Add controls to the Properties panel
- Add templates and resources
- Add commands to the element context menu
- Add menus to the top menu bar
- Add commands to the project menu
- Add commands to the project file/folder context menu
- Add actions to the Actions panel
- Hook custom functions to events like saving a page or opening a project
- Create dynamic CMS editable areas

Next: Creating a plugin

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package