Main Class

- Class Structure
- Single Instance
- Constructor

We can move the different components into a single class - our main class. It should check basic requirements and load the files required to activate the addon's functionality, and then load the various components.

Class Structure

A good example of an addon class is one that hosts the following functionality:

```
final class Plugin {

	private static $_instance = null;
	public static function instance() {}

	public function __construct() {}
	public function is_compatible(): bool {}
	public function init(): void {}

}
```

Single Instance

The main class should use a singleton design pattern to make sure it loads only once:

```
final class Plugin {

	/**
	 * Instance
	 *
	 * @since 1.0.0
	 * @access private
	 * @static
	 * @var \Elementor_Test_Addon\Plugin The single instance of the class.
	 */
	private static $_instance = null;

	/**
	 * Instance
	 *
	 * Ensures only one instance of the class is loaded or can be loaded.
	 *
	 * @since 1.0.0
	 * @access public
	 * @static
	 * @return \Elementor_Test_Addon\Plugin An instance of the class.
	 */
	public static function instance() {

		if ( is_null( self::$_instance ) ) {
			self::$_instance = new self();
		}
		return self::$_instance;

	}

}

\Plugin::instance();
```

Constructor

The constructor should perform compatibility checks to make sure the basic requirements are met. If all checks pass, the main functionality should be initialized.

```
final class Plugin {

	/**
	 * Constructor
	 *
	 * Perform some compatibility checks to make sure basic requirements are meet.
	 * If all compatibility checks pass, initialize the functionality.
	 *
	 * @since 1.0.0
	 * @access public
	 */
	public function __construct() {

		if ( $this->is_compatible() ) {
			add_action( 'elementor/init', [ $this, 'init' ] );
		}

	}

	/**
	 * Compatibility Checks
	 *
	 * Checks whether the site meets the addon requirement.
	 *
	 * @since 1.0.0
	 * @access public
	 */
	public function is_compatible(): bool {

		// Compatibility checks here...

	}

	/**
	 * Initialize
	 *
	 * Load the addons functionality only after Elementor is initialized.
```

```
 *
 * Fired by `elementor/init` action hook.
 *
 * @since 1.0.0
 * @access public
 */
public function init(): void {

// Addon functionality here...

}

}
```

Now let's examine a few examples of addon compatibility checks:

□□□ #2
□ □□□ □□□ 2026-03-27 01:08:44 UTC
□ □□□ □□□ 2026-03-27 01:08:44 UTC