



To develop a **custom Lottie Files block in WordPress (Gutenberg)** using clean code, you'll need to:

- Register the block with a custom icon.
- Allow users to input a Lottie JSON URL or upload a file.
- Render the Lottie animation in both editor and frontend.
- Use minimal and readable code structure.

---

## 1. Project Structure (in plugin form)

```
bashCopyEdit/lottie-block/
├── lottie-block.php
├── block.json
├── /build/
│   ├── index.js
│   └── style.css
└── /src/
    └── index.js
```

---

## 2. lottie-block.php - Plugin Loader

```
phpCopyEdit<?php
/**
 * Plugin Name: Lottie Files Block
 * Description: A Gutenberg block to insert Lottie animations.
 * Version: 1.0
 * Author: Your Name
 */

function lottie_block_register() {
    register_block_type(__DIR__);
}
add_action('init', 'lottie_block_register');
```



### 3. block.json - Block Registration

```
jsonCopyEdit{
  "apiVersion": 2,
  "name": "custom/lottie-block",
  "title": "Lottie Animation",
  "category": "media",
  "icon": {
    "src": "smiley",
    "foreground": "#2b6cb0"
  },
  "description": "Embed Lottie animations via URL or upload.",
  "keywords": ["lottie", "animation", "json"],
  "attributes": {
    "src": {
      "type": "string",
      "default": ""
    }
  },
  "editorScript": "file:./build/index.js",
  "style": "file:./build/style.css",
  "render": "file:./render.php"
}
```

---

### 4. src/index.js - Block Logic (React/JSX)

```
jsCopyEditimport { registerBlockType } from '@wordpress/blocks';
import { MediaUpload, InspectorControls } from '@wordpress/block-editor';
import { PanelBody, TextControl, Button } from '@wordpress/components';
import { __ } from '@wordpress/i18n';
import { useEffect } from '@wordpress/element';
```



```
registerBlockType('custom/lottie-block', {
  edit: ({ attributes, setAttributes }) => {
    const { src } = attributes;

    return (
      <>
        <InspectorControls>
          <PanelBody title="Lottie Animation Settings">
            <TextControl
              label="Lottie JSON URL"
              value={src}
              onChange={(value) => setAttributes({ src:
value }}}
              help="Paste a Lottie JSON file URL or
upload a file."
            />
          </PanelBody>
        </InspectorControls>
        <div className="lottie-preview">
          {src ? (
            <lottie-player
              autoplay
              loop
              mode="normal"
              src={src}
              style={{ width: '100%', height: '300px' }}
            />
          ) : (
            <p>Please provide a Lottie JSON URL.</p>
          )}
        </div>
      </>
    );
  },
});
```

---



## 5. render.php - Frontend Output

```
phpCopyEdit<?php
$src = esc_url($attributes['src'] ?? '');

if ($src) {
    echo '<lottie-player autoplay loop src="' . $src . '"
style="width:100%;height:300px;"></lottie-player>';
} else {
    echo '<p>No Lottie animation selected.</p>';
}
```

---

## 6. style.css - Optional Styles

```
cssCopyEdit.lottie-preview {
    border: 1px dashed #ccc;
    padding: 10px;
    text-align: center;
}
```

---

## 7. Include Lottie Player

To support <lottie-player>, enqueue the official Lottie player:

```
phpCopyEditfunction lottie_enqueue_script() {
    wp_enqueue_script(
        'lottie-player',
        'https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottie-playe
r.js',
        [],
        null,
        true
    );
}
```



```
add_action('wp_enqueue_scripts', 'lottie_enqueue_script');
add_action('enqueue_block_editor_assets', 'lottie_enqueue_script');
```

---

## 8. Build with WP Scripts

Add to your `package.json`:

```
jsonCopyEdit"scripts": {
  "build": "wp-scripts build",
  "start": "wp-scripts start"
}
```

Install dependencies:

```
bashCopyEditnpm install @wordpress/scripts --save-dev
```

Then run:

```
bashCopyEditnpm run build
```

---

## Custom Icon

You can replace "smiley" with a custom Dashicon or SVG inline icon in the `block.json`. For example:

```
jsonCopyEdit"icon": {
  "src": "<svg>...</svg>",
  "foreground": "#ff6600"
}
```

To convert the Lottie Files block from a plugin to a theme-based block, we simply move the block code inside the theme directory – typically under `wp-content/themes/your-theme/blocks/lottie-block/` – and register it from the theme's `functions.php`.



## Final Folder Structure (Inside Your Theme)

pgsql

Copy

Edit

/your-theme/

```
|
|— functions.php
|— style.css
|— blocks/
|   |— lottie-block/
|       |— block.json
|       |— render.php
|       |— build/
|           |— index.js
|           |— style.css
```

1. functions.php – Register the Block from the Theme

Add this to your theme's functions.php:

php

Copy

Edit

```
function theme_register_lottie_block() {
    register_block_type( get_template_directory() . '/blocks/lottie-block'
    );
}

add_action('init', 'theme_register_lottie_block');

function theme_enqueue_lottie_script() {
    wp_enqueue_script(
        'lottie-player',
        'https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottie-playe
        r.js',
        [],
        null,
        true
    );
}

add_action('enqueue_block_assets', 'theme_enqueue_lottie_script');
```



## 2. blocks/lottie-block/block.json

Same as before, but now relative to the theme path:

json

Copy

Edit

```
{
  "apiVersion": 2,
  "name": "custom/lottie-block",
  "title": "Lottie Animation",
  "category": "media",
  "icon": {
    "src": "format-video",
    "foreground": "#ff6f61"
  },
  "description": "Embed Lottie animations via URL or file upload.",
  "keywords": ["lottie", "animation", "json"],
  "attributes": {
    "src": {
      "type": "string",
      "default": ""
    }
  },
  "editorScript": "file:./build/index.js",
  "style": "file:./build/style.css",
  "render": "file:./render.php"
}
```

## 3. render.php

Unchanged from plugin version:

php

Copy

Edit

<?php

```
$src = esc_url($attributes['src'] ?? '');
```

```
if ($src) {
  echo '<lottie-player autoplay loop src="' . $src . '"
```



```
style="width:100%;height:300px;"></lottie-player>';  
} else {  
echo '<p>No Lottie animation selected.</p>';  
}
```

4. build/index.js

Same JSX logic:

js

Copy

Edit

```
import { registerBlockType } from '@wordpress/blocks';  
import { InspectorControls } from '@wordpress/block-editor';  
import { PanelBody, TextControl } from '@wordpress/components';  
import { __ } from '@wordpress/i18n';
```

```
registerBlockType('custom/lottie-block', {  
edit: ({ attributes, setAttributes }) => {  
const { src } = attributes;
```

```
return (  
<>  
<InspectorControls>  
<PanelBody title="Lottie Animation Settings">  
<TextControl  
label="Lottie JSON URL"  
value={src}  
onChange={(value) => setAttributes({ src: value })}  
help="Paste a Lottie JSON file URL or upload a file."  
>  
</PanelBody>  
</InspectorControls>  
<div className="lottie-preview">  
{src ? (  
<lottie-player  
autoplay  
loop  
mode="normal"  
src={src}
```



```
style={{ width: '100%', height: '300px' }}
/>
) : (
<p>Please provide a Lottie JSON URL.</p>
)}
</div>
</>
);
}
});
```

5. build/style.css

css

Copy

Edit

```
.lottie-preview {
border: 1px dashed #ccc;
padding: 10px;
text-align: center;
}
```

6. Build Your Block

If using npm with @wordpress/scripts:

bash

Copy

Edit

```
npm install @wordpress/scripts --save-dev
```

```
npm run wp-scripts build
```

Or you can manually place the built files in /build/.

## Share this:

- [Click to share on Facebook \(Opens in new window\) Facebook](#)
- [Click to print \(Opens in new window\) Print](#)
- [Click to email a link to a friend \(Opens in new window\) Email](#)
- [Click to share on LinkedIn \(Opens in new window\) LinkedIn](#)



- [Click to share on Tumblr \(Opens in new window\) Tumblr](#)
- [Click to share on Pinterest \(Opens in new window\) Pinterest](#)
- [Click to share on Telegram \(Opens in new window\) Telegram](#)
- [Click to share on WhatsApp \(Opens in new window\) WhatsApp](#)