



# Experience Builder (XB) in Drupal



# What is Experience Builder?

Experience Builder provides an end-to-end solution that is easier to use than Layout Builder and Paragraphs.

## Key User Experience Improvements:

- Visual, browser-based tool for building complex pages.
- Built with ease of use for **Site Builders, Junior Developers, and Content Creators** in mind.
- Predefined templates and components for faster page creation.

**Why it Stands Out:** Combines the best of Layout Builder and Paragraphs but improves on the user interface and ease of use.



# Why Do We Need Experience Builder?

- **Challenges with Existing Solutions:** Layout Builder, Paragraphs, and Gutenberg have significant usability issues.
- **Importance of Page Building:** Critical for content creators, who often need to build pages quickly, without relying on developers.
- **Key Takeaways:**
  - People are moving away from Drupal due to page-building challenges.
  - Many organizations require a more intuitive, visual tool.
  - Builders expect to deliver pages on aggressive timelines without developer intervention.



# Findings from Research on Existing Solutions

**Layout Builder:** Difficult to use, requires many patches, and is hard for non-technical users to understand.

**Paragraphs:** Powerful but disjointed from visual design, and has scalability issues with larger sites.

**Gutenberg:** Fast setup but hard to customize and integrate with Drupal.

## Key Challenges:

- **Dependency on Developers:** Even basic page creation often requires developers.
- **Complexity:** Setting up and maintaining Layout Builder requires too many modules.
- **UI Limitations:** None of the tools provide a truly intuitive user interface for content creators.

Reference - [Page Builder Research 2024](#)



# The Overall Approach of Experience Builder

- **Single Directory Components (SDC):** Components with all files (Twig, YAML, SCSS, JS) in one directory.
- **Reusable Components:** Create once, use everywhere.
- **Seamless User Experience:** Drag-and-drop interface for layout creation.
- **Flexibility for Developers:** Easy to create and register custom components.



# What's Latest in the Experience Builder Initiative?

- **Component-based architecture** using SDC.
- **Improved flexibility** for grid systems, rows, columns.
- **Enhanced preview and live editing** capabilities.
- **Integration** with Media Library.



# How to Get Started with Experience Builder (XB)

- [https://git.drupalcode.org/project/experience\\_builder/-/blob/0.x/CONTRIBUTING.md?ref\\_type=heads](https://git.drupalcode.org/project/experience_builder/-/blob/0.x/CONTRIBUTING.md?ref_type=heads)
- <https://www.drupal.org/community-initiatives/starshot-demo-design-system/sdds-development-guide>
- <https://github.com/TravisCarden/ddev-drupal-xb-dev>



## Next Steps

[Milestone 0.1.0: Experience Builder Demo](#) - Hosted Demo at DrupalCon Barcelona 2024

[\[META\] Support component types other than SDC](#) - Support for Blocks, Views, Menus.

[Milestone 0.2.0: Experience Builder-rendered nodes](#) - Save content created using the Experience Builder





# Running Cypress Tests

## Running Unit/Component Tests

- `npm run cy:component`

## Running E2E Tests

- In your `.env` file, set `BASE_URL` and `DB_URL`. See `.env.example` for an example.
- The e2e tests use the application file in `/ui/dist`, which is only updated by running `npm run build`. Be sure to do this before running e2e tests.
- Then, *either*:
  - Use `npm run cy:open` to run e2e with the (very helpful) Cypress GUI test runner (do that in its own terminal). This runs the test in a visible browser.
  - Use `npm run cy:run` to run the same e2e tests in the terminal (this is also the command used by Gitlab CI). This runs the test in a "headless" browser.



# Demo (XB + SDDS)



# Q&A