

## Problems with Gatsby Source WordPress v3 :broken\_heart:

`gatsby-source-wordpress@v3`, the last major version of this plugin, utilized the WP REST API for sourcing WP data into Gatsby. This worked but was problematic for a few reasons.

- The inflexibility and lack of an enforced schema for this API made proper node caching technically possible but functionally impossible. It also made it impossible to know beforehand the shape and type of data to expect which made it impossible to programmatically determine how to use the returned data.
- The REST API is largely untyped, and REST does not enforce typing even if it was.
  - This means Gatsby needed to use inference to automatically build GraphQL types for the data that was returned from WP. The big issue with this is: if all data of a type is removed from WP, your Gatsby site's GraphQL queries will break because no data means Gatsby can no longer infer the data structure. This led to a lot of odd and counterintuitive issues and a poor developer/content creator experience.
  - If we were to manually write out types for the entire core WP REST API so that we no longer needed to use inference, any custom WP REST plugins would not work and would require a lot of additional tedious and error prone work to support these plugins changing their data structure over time.
- REST does not have a popularly used standard way of implementing connections between nodes
  - For example if you fetch 100 posts, you will also be fetching repeated data if each post includes data about the posts author including their name and email, etc. Perhaps all the posts had the same author and so we fetched 100x more data than we needed to.
  - Even if WP REST API core implemented a standard way to retrieve connections to prevent overfetching connected data, any WP REST API extensions, community or otherwise, may or may not adhere to this standard, causing a large problem and stunting the ability for the community to easily extend the integration between WP and Gatsby. This would also mean a Gatsby plugin would need to be written for every single WP REST API extension, ballooning the amount of support needed for the WP/Gatsby integration to flourish.
- Being that the WP REST API is part of WordPress core, developments, bug fixes, and improvements to it move very slowly. There has been an open issue about broken media items for 4 years with no resolution in sight. This bug occasionally breaks Gatsby builds due to improperly missing media items. This leads to poor DX and cryptic error messages. Fixing this bug in a WP instance requires opening an SQL client and manually

patching your DB each time this bug occurs.

- The WP REST API does not support many basic types of WP data such as menus, previews, plugins, themes.

All of the above points are fixed by using this plugin and WPGraphQL :smile\_cat:

## Up Next :point\_right:

- :runner: Installation & Getting started
- :school: Tutorials
- :feet: Features
- :electric\_plug: Plugin options
- :boat: Migrating from other WP source plugins
- :house: Hosting WordPress
- :athletic\_shoe: Themes, Starters, and Examples
- :medal\_sports: Usage with popular WPGraphQL extensions
- :hammer\_and\_wrench: Debugging and troubleshooting
- :national\_park: Community and Support
- :point\_left: Back to README.md