

Plugin Options

- [url \(required\)](#)
- [verbose](#)
- [debug](#)
 - [debug.preview](#)
 - [debug.timeBuildSteps](#)
 - [debug.disableCompatibilityCheck](#)
 - [debug.throwRefetchErrors](#)
 - [debug.graphql](#)
 - [debug.graphql.showQueryVarsOnError](#)
 - [debug.graphql.showQueryOnError](#)
 - [debug.graphql.copyQueryOnError](#)
 - [debug.graphql.panicOnError](#)
 - [debug.graphql.onlyReportCriticalErrors](#)
 - [debug.graphql.copyNodeSourcingQueryAndExit](#)
 - [debug.graphql.writeQueriesToDisk](#)
 - [debug.graphql.printIntrospectionDiff](#)
- [production](#)
 - [production.hardCacheMediaFiles](#)
 - [production.allow404Images](#)
 - [production.allow401Images](#)
- [develop](#)
 - [develop.nodeUpdateInterval](#)
 - [develop.hardCacheMediaFiles](#)
 - [develop.hardCacheData](#)
- [auth](#)
 - [auth.htaccess](#)
 - [auth.htaccess.username](#)
 - [auth.htaccess.password](#)
- [schema](#)
 - [schema.queryDepth](#)
 - [schema.circularQueryLimit](#)
 - [schema.typePrefix](#)
 - [schema.timeout](#)
 - [schema.perPage](#)
 - [schema.requestConcurrency](#)
 - [schema.previewRequestConcurrency](#)
- [excludeFieldNames](#)
- [searchAndReplace](#)
 - [searchAndReplace\[\].search](#)
 - [searchAndReplace\[\].replace](#)
- [html](#)
 - [html.useGatsbyImage](#)
 - [html.gatsbyImageOptions](#)
 - [html.imageMaxWidth](#)
 - [html.fallbackImageMaxWidth](#)
 - [html.imageQuality](#)

- [html.createStaticFiles](#)
- [html.generateWebpImages](#)
- [html.placeholderType](#)
- [type](#)
 - [type._all](#)
 - [type._all.where](#)
 - [type._all.exclude](#)
 - [type._all.limit](#)
 - [type._all.excludeFieldNames](#)
 - [type._all.nodeInterface](#)
 - [type._all.beforeChangeNode](#)
 - [type.RootQuery](#)
 - [type.Medialtem](#)
 - [type.Medialtem.placeholderSizeName](#)
 - [type.Medialtem.createFileNodes](#)
 - [type.Medialtem.lazyNodes](#)
 - [type.Medialtem.localFile](#)
 - [type.Medialtem.localFile.excludeByMimeTypes](#)
 - [type.Medialtem.localFile.maxFileSizeBytes](#)
 - [type.Medialtem.localFile.requestConcurrency](#)
- [presets](#)
 - [presets\[\].presetName](#)
 - [presets\[\].useSelf](#)
 - [presets\[\].options](#)

url (required)

This is the only plugin option which is required for the plugin to work properly.

This should be the full url of your GraphQL endpoint.

Field type: `String`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    url: `https://yoursite.com/graphql`,
  },
}
```

verbose

Enables verbose logging in the terminal. Set to `false` to turn it off.

Field type: `Boolean`

Default value: `true`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    verbose: true,
  },
}
```

debug

An object which contains options related to debugging. See below for options.

Field type: Object

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      // Add your options here :)
    },
  },
}
```

debug.preview

When set to true, this option will display additional information in the terminal output about the running preview process.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      preview: true,
    },
  },
}
```

debug.timeBuildSteps

When set to true, this option will display how long each internal step took during the build process.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
```

```
options: {
  debug: {
    timeBuildSteps: true,
  },
},
}
```

debug.disableCompatibilityCheck

This option disables the compatibility API check against the remote WPGraphQL and WPGatsby plugin versions. Note that it's highly recommended to not disable this setting. If you disable this setting you will receive no support until it's re-enabled. It's also highly likely that you'll run into major bugs without initially realizing that this was the cause.

This option should only be used for debugging.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      disableCompatibilityCheck: true,
    },
  },
}
```

debug.throwRefetchErrors

When this is set to true, errors thrown while updating data in gatsby develop will fail the build process instead of automatically attempting to recover.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      throwRefetchErrors: true,
    },
  },
}
```

debug.graphql

An object which contains GraphQL debugging options. See below for options.

Field type: Object

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      graphql: {
        // Add your options here :)
      },
    },
  },
}
```

debug.graphql.showQueryVarsOnError

When a GraphQL error is returned and the process exits, this plugin option determines whether or not to log out the query vars that were used in the query that returned GraphQL errors.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      graphql: {
        showQueryVarsOnError: true,
      },
    },
  },
}
```

debug.graphql.showQueryOnError

If enabled, GraphQL queries will be printed to the terminal output when the query returned errors.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      graphql: {
        showQueryOnError: true,
      },
    },
  },
}
```

debug.graphql.copyQueryOnError

If enabled, GraphQL queries will be copied to your OS clipboard (if supported) when the query returned errors.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      graphql: {
        copyQueryOnError: true,
      },
    },
  },
}
```

debug.graphql.panicOnError

Determines whether or not to panic when any GraphQL error is returned.

Default is false because sometimes non-critical errors are returned alongside valid data.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      graphql: {
        panicOnError: false,
      },
    },
  },
}
```

debug.graphql.onlyReportCriticalErrors

Determines whether or not to log non-critical errors. A non-critical error is any error which is returned alongside valid data. In previous versions of WPGraphQL this was very noisy because trying to access an entity that was private returned errors.

Field type: Boolean

Default value: true

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      graphql: {
```

```
        onlyReportCriticalErrors: true,
      },
    },
  },
}
```

debug.graphql.copyNodeSourcingQueryAndExit

When a type name from the remote schema is entered here, the node sourcing query will be copied to the clipboard, and the process will exit.

Field type: String

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      graphql: {
        copyNodeSourcingQueryAndExit: true,
      },
    },
  },
}
```

debug.graphql.writeQueriesToDisk

When true, all internal GraphQL queries generated during node sourcing will be written out to `./WordPress/GraphQL/[TypeName]/*.graphql` for every type that is sourced. This is very useful for debugging GraphQL errors.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      graphql: {
        writeQueriesToDisk: true,
      },
    },
  },
}
```

debug.graphql.printIntrospectionDiff

When this is set to true it will print out the diff between types in the previous and new schema when the schema changes. This is enabled by default when `debug.preview` is enabled.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    debug: {
      graphql: {
        printIntrospectionDiff: true,
      },
    },
  },
}
```

production

Field type: Object

production.hardCacheMediaFiles

This option is experimental. When set to true, media files will be hard-cached outside the Gatsby cache at `./wordpress-cache/path/to/media/file.jpeg`. This is useful for preventing media files from being re-downloaded when the Gatsby cache automatically clears. When using this option, be sure to gitignore the `wordpress-cache` directory in the root of your project.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    production: {
      hardCacheMediaFiles: true,
    },
  },
}
```

production.allow404Images

This option allows images url's that return a 404 to not fail production builds.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    production: {
```



```
    allow404Images: true,
  },
},
}
```

production.allow401Images

This option allows images url's that return a 401 to not fail production builds. 401s are sometimes returned in place of 404's for protected content to hide whether the content exists.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    production: {
      allow401Images: true,
    },
  },
}
```

develop

Options related to the gatsby develop process.

Field type: Object

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    develop: {
      // options related to `gatsby develop`
    },
  },
}
```

develop.nodeUpdateInterval

Specifies in milliseconds how often Gatsby will ask WP if data has changed during development. If you want to see data update in near-realtime while you're developing, set this low. Your server may have trouble responding to too many requests over a long period of time and in that case, set this high. Setting it higher saves electricity too ⚡🌲.

Field type: Number

Default value: 5000

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
```

```
    develop: {
      nodeUpdateInterval: 300,
    },
  },
}
```

develop.hardCacheMediaFiles

This option is experimental. When set to true, media files will be hard-cached outside the Gatsby cache at `./wordpress-cache/path/to/media/file.jpeg`. This is useful for preventing media files from being re-downloaded when the Gatsby cache automatically clears. When using this option, be sure to gitignore the `wordpress-cache` directory in the root of your project.

Field type: `Boolean`

Default value: `false`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    develop: {
      hardCacheMediaFiles: true,
    },
  },
}
```

develop.hardCacheData

This option is experimental. When set to true, WordPress data will be hard-cached outside the Gatsby cache in `./wordpress-cache/caches`. This is useful for preventing the need to re-fetch all data when the Gatsby cache automatically clears. This hard cache will automatically clear itself when your remote WPGraphQL schema changes, or when you change your plugin options.

When using this option, be sure to gitignore the `wordpress-cache` directory in the root of your project.

Field type: `Boolean`

Default value: `false`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    develop: {
      hardCacheData: false,
    },
  },
}
```

auth

Options related to authentication. See below for options.

Field type: Object

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    auth: {
      // Add your options here :)
    },
  },
}
```

auth.htaccess

Options related to htaccess authentication.

Field type: Object

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    auth: {
      htaccess: {
        // Add your options here :)
      },
    },
  },
}
```

auth.htaccess.username

The username for your .htpassword protected site.

Field type: String

Default value: null

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    auth: {
      htaccess: {
        username: `admin`,
      },
    },
  },
}
```

auth.htaccess.password

The password for your .htpassword protected site.

Field type: String

Default value: `null`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    auth: {
      htaccess: {
        password: `1234strong_password`,
      },
    },
  },
}
```

schema

Options related to fetching and ingesting the remote schema.

Field type: `Object`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    schema: {
      // Add your options here :)
    },
  },
}
```

schema.queryDepth

The maximum field depth the remote schema will be queried to.

Field type: `Number`

Default value: `15`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    schema: {
      queryDepth: 15,
    },
  },
}
```

schema.circularQueryLimit

The maximum number times a type can appear as it's own descendant.

Field type: `Number`

Default value: 5

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    schema: {
      circularQueryLimit: 5,
    },
  },
}
```

schema.typePrefix

The prefix for all ingested types from the remote schema. For example Post becomes WpPost.

Field type: String

Default value: Wp

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    schema: {
      typePrefix: `Wp`,
    },
  },
}
```

schema.timeout

The amount of time in ms before GraphQL requests will time out.

Field type: Number

Default value: 30000

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    schema: {
      timeout: 30000,
    },
  },
}
```

schema.perPage

The number of nodes to fetch per page during node sourcing.

Field type: Number

Default value: 100

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    schema: {
      perPage: 100,
    },
  },
}
```

schema.requestConcurrency

The number of concurrent GraphQL requests to make at any time during node sourcing. Try lowering this if your WordPress server crashes while sourcing data.

Field type: Number

Default value: 15

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    schema: {
      requestConcurrency: 50,
    },
  },
}
```

schema.previewRequestConcurrency

The number of concurrent GraphQL requests to make at any time during preview sourcing. Try lowering this if your WordPress server crashes during previews. Normally this won't be needed and only comes into effect when multiple users are previewing simultaneously.

Field type: Number

Default value: 5

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    schema: {
      previewRequestConcurrency: 50,
    },
  },
}
```

excludeFieldNames

A list of field names to globally exclude from the ingested schema.

Field type: Array

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    excludeFieldNames: [`viewer`],
  },
}
```

searchAndReplace

An array of options to search and replace strings in nodes. See below for options.

Field type: Array

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    searchAndReplace: [
      {
        search: "https://some-url.com",
        replace: "https://some-new-url.com",
      },
    ],
  },
}
```

searchAndReplace[].search

The regex rule used to search when replacing strings in node data. It will search the stringified JSON of each node to capture strings at any nested depth.

Field type: String

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    searchAndReplace: [
      {
        search: "https://some-url.com",
      },
    ],
  },
}
```

searchAndReplace[].replace

The replacement string for each regex match.

Field type: String

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    searchAndReplace: [
      {
        replace: "https://some-new-url.com",
      },
    ],
  },
}
```

html

Options related to html field processing.

Field type: `Object`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    html: {
      // Add your options here :)
    },
  },
}
```

html.useGatsbyImage

Causes the source plugin to find/replace images in html with Gatsby images.

Field type: `Boolean`

Default value: `true`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    html: {
      useGatsbyImage: true,
    },
  },
}
```

html.gatsbyImageOptions

Set custom options for your Gatsby Images.

Field type: `Object`


```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    html: {
      gatsbyImageOptions: {
        [your - option - key]: "your-option-value",
        [your - option - key - 2]: "your-option-value-2",
      },
    },
  },
}
```

html.imageMaxWidth

Adds a limit to the max width an image can be. If the image size selected in WP is smaller or the image file width is smaller than this those values will be used instead.

Field type: Number

Default value: null

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    html: {
      imageMaxWidth: 1024,
    },
  },
}
```

html.fallbackImageMaxWidth

If a max width can't be inferred from html this value will be passed to Sharp. If the image is smaller than this, the image file's width will be used instead.

Field type: Number

Default value: 1024

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    html: {
      fallbackImageMaxWidth: 800,
    },
  },
}
```

html.imageQuality

Determines the image quality that Sharp will use when generating inline html image thumbnails.

Field type: Number

Default value: 90

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    html: {
      imageQuality: 90,
    },
  },
}
```

html.createStaticFiles

When this is true, any url's which are wrapped in "", ", or () and which contain /wp-content/uploads will be transformed into static files and the url's will be rewritten. This adds support for video, audio, and anchor tags which point at WP media item uploads as well as inline-html css like background-image: url().

Field type: Boolean

Default value: true

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    html: {
      createStaticFiles: true,
    },
  },
}
```

html.generateWebpImages

When this is true, .webp images will be generated for images in html fields in addition to the images gatsby-image normally generates.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    html: {
      generateWebpImages: false,
    },
  },
}
```

html.placeholderType

This can be either "blurred" or "dominantColor". This is the type of placeholder image to be used in Gatsby Images in HTML fields.

Field type: String

Default value: blurred

type

Options related to specific types in the remote schema.

Field type: Object

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
      // Add your options here :)
    },
  },
}
```

type.__all

A special type setting which is applied to all types in the ingested schema.

Field type: Object

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
      __all: {
        limit: 10,
      },
    },
  },
}
```

type.__all.where

This string is passed as the WPGraphQL "where" arguments in the GraphQL queries that are made while initially sourcing all data from WPGraphQL into Gatsby during an uncached build. A common use-case for this is only fetching posts of a specific language. It's often used in conjunction with the beforeChangeNode type option as "where" only affects the initial data sync from WP to Gatsby while beforeChangeNode will also run when syncing individual updates from WP to Gatsby.

Field type: String

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
```

```

    type: {
      Page: {
        where: `language: ${process.env.GATSBY_ACTIVE_LANGUAGE}`,
      },
    },
  },
}

```

type.__all.exclude

Completely excludes a type from node sourcing and from the ingested schema.

Field type: Boolean

```

{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
      Page: {
        exclude: true,
      },
    },
  },
}

```

type.__all.limit

The maximum amount of objects of this type to fetch from WordPress.

Field type: Number

type.__all.excludeFieldNames

Excludes fields on a type by field name.

Field type: Array

```

{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
      Page: {
        excludeFieldNames: [`dateGmt`, `parent`],
      },
    },
  },
}

```

type.__all.nodeInterface

Determines whether or not this type will be treated as an interface comprised entirely of other Gatsby node types.

Field type: Boolean

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
      Page: {
        nodeInterface: true,
      },
    },
  },
}
```

type.__all.beforeChangeNode

A function which is invoked before a node is created, updated, or deleted. This is a hook in point to modify the node or perform side-effects related to it. This option should be a path to a JS file where the default export is the `beforeChangeNode` function. The path can be relative to your `gatsby-node.js` or absolute. Currently you can inline a function by writing it out directly in this option but starting from Gatsby v4 only a path to a function file will work.

Field type: `String | Function`

type.RootQuery

A special type which is applied to any non-node root fields that are ingested and stored under the root `wp` field. It accepts the same options as other types.

Field type: `Object`

Default value: `{ excludeFieldNames: ['viewer', 'node', 'schemaMd5'], },`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    RootQuery: {
      excludeFieldNames: ['viewer'],
    },
  },
}
```

type.MedialItem

Field type: `Object`

type.MedialItem.placeholderSizeName

This option allows you to choose the placeholder size used in the new Gatsby image service (currently in ALPHA/BETA) for the small placeholder image. Please make this image size very small for better performance. 20px or smaller width is recommended. To use, create a new image size in WP and name it "gatsby-image-placeholder" (or the name that you pass to this option) and that new size will be used automatically for placeholder images in the Gatsby build.

Field type: `String`

Default value: `gatsby-image-placeholder`

type.Medialtem.createFileNodes

This option controls whether or not a File node will be automatically created for each Medialtem node (available on Medialtem.localFile). Set this to false if you don't want Gatsby to download the corresponding file for each media item.

Field type: Boolean

Default value: true

type.Medialtem.lazyNodes

Enables a different media item sourcing strategy. Instead of fetching Media Items that are referenced by other nodes, Media Items will be fetched in connection resolvers from other nodes. This may be desirable if you're not using all of the connected images in your WP instance. This is not currently recommended because it messes up cli output and can be slow due to query running concurrency.

This option no longer works starting in Gatsby v4+. If you want to prevent this plugin from creating File nodes for each Medialtem node, set the type.Medialtem.createFileNodes option to false instead.

Field type: Boolean

Default value: false

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
      MediaItem: {
        lazyNodes: true,
      },
    },
  },
}
```

type.Medialtem.localFile

Options related to File nodes that are attached to Medialtem nodes.

Field type: Object

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
      MediaItem: {
        localFile: {
          // Add your options here :)
        },
      },
    },
  },
}
```

type.Medialtem.localFile.excludeByMimeTypes

Allows preventing the download of files associated with Medialtem nodes by their mime types.

Field type: Array

Default value: []

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
      MediaItem: {
        localFile: {
          excludeByMimeTypes: [`video/mp4`],
        },
      },
    },
  },
}
```

type.Medialtem.localFile.maxFileSizeBytes

Allows preventing the download of files that are above a certain file size (in bytes). Default is 15mb.

Field type: Number

Default value: 15728640

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
      MediaItem: {
        localFile: {
          maxFileSizeBytes: 10485760, // 10Mb
        },
      },
    },
  },
}
```

type.Medialtem.localFile.requestConcurrency

Amount of images to download concurrently. Try lowering this if wordpress server crashes on import.

Field type: Number

Default value: 100

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    type: {
```

```

MediaItem: {
  localFile: {
    requestConcurrency: 50,
  },
},
},
},
},
}

```

presets

An array of plugin options presets that are applied if the `useIf` function on each returns true. The default includes an optimization for when in Gatsby Preview mode.

Field type: `Array`

Default value:

```

;[
  {
    presetName: `PREVIEW_OPTIMIZATION`,
    useIf: (): boolean =>
      (process.env.NODE_ENV === `development` &&
        !!process.env.ENABLE_GATSBY_REFRESH_ENDPOINT) ||
        process.env.RUNNER_TYPE === `PREVIEW`,
    options: {
      html: {
        useGatsbyImage: false,
        createStaticFiles: false,
      },
      type: {
        __all: {
          limit: 50,
        },
        Comment: {
          limit: 0,
        },
        Menu: {
          limit: null,
        },
        MenuItem: {
          limit: null,
        },
        User: {
          limit: null,
        },
      },
    },
  },
],

```


presets[].presetName

The name of the plugin options preset.

Field type: String

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    presets: [
      {
        presetName: `DEVELOP`,
      },
    ],
  },
}
```

presets[].useIf

A function used to determine whether or not to apply this plugin options preset. It should return a boolean value. True will cause the preset to apply, false will disclude it.

Field type: Function

Default value: `() => false`

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    presets: [
      {
        useIf: () => process.env.NODE_ENV === `development`,
      },
    ],
  },
}
```

presets[].options

Any valid options except for `url` and `presets`.

Field type: Object

```
{
  resolve: `gatsby-source-wordpress`,
  options: {
    presets: [
      {
        name: `DEVELOP`,
        useIf: () => process.env.NODE_ENV === `development`,
        options: {
```

```
    type: {
      __all: {
        limit: 1,
      },
    },
  },
},
],
},
}
```

Up Next :point_right:

- :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](#)