Default Types and Fields

- Scalars
  - Standard GraphQL Scalars
  - Custom Scalars
- Interfaces
- Object Types
- Enums
- Inputs
- Unions
- Connections
- Mutations

On this page you will find information about the default Types and Fields that are generated by WPGraphQL.

The Types registered by WPGraphQL can be used by 3rd party plugins that are registering GraphQL fields, connections, arguments and inputs.

Types must be unique in the Schema, so any Type already declared by WPGraphQL should not be re-declared by plugins extending WPGraphQL.

De-registering or removing Types from the Schema is not supported.

Scalars

WPGraphQL registers the following Scalars for use throughout the Schema.

Standard GraphQL Scalars

- **Boolean**
- **Float**
- **ID**

- **Integer**
- **String**

Custom Scalars

For more information about custom scalars and how to create your own, see the Custom Scalars documentation.

Interfaces

Next, WPGraphQL registers Interfaces. Interfaces are special Types that declare common fields that Object Types have in common. The following Interfaces are registered by default:

- **Node**
- **Commenter**
- **ContentNode**
- **ContentTemplate**
- **DatabaseIdentifier**
- **EnqueuedAsset**
- **HierarchicalTermNode**
- **HierarchicalContentNode**
- **MenuItemLinkable**
- **NodeWithAuthor**
- **NodeWithComments**
- **NodeWithContentEditor**
- **NodeWithExcerpt**
- **NodeWithFeaturedImage**
- **NodeWithTitle**
- **NodeWithTrackbacks**
- **NodeWithPageAttributes**
- **TermNode**
- **UniformResourceIdentifiable**

Object Types

Next, Object Types are registered to represent common Types of data managed in WordPress.

- **RootQuery**
- **PostObjects**: the GraphQL Type for each post type set to "show_in_graphql"

- **TermObjects**: the GraphQL Type for each Taxonomy set to "show_in_graphql"
- **RootMutation**
- **Avatar**
- **Comment**
- **CommentAuthor**
- **EnqueuedStylesheet**
- **EnqueuedScript**
- **MediaDetails**
- **MediaItemMeta**
- **MediaSize**
- **Menu**
- **MenuItem**
- **PageInfo**
- **Plugin**
- **ContentType**
- **PostTypeLabelDetails**
- **Settings**
- **Taxonomy**
- **Theme**
- **User**
- **UserRole**
- **Templates:**
  - An Object Type is registered for each Page Template based on it's Template Name.

Enums

- **AvatarRatingEnum**
- **CommentsConnectionOrderbyEnum**
- **ContentNodeIdTypeEnum**
- **ContentTypeEnum**
- **ContentTypeIdTypeEnum**
- **MediaItemSizeEnum**
- **MediaItemStatusEnum**
- **MenuLocationEnum**
- **MenuItemNodeIdTypeEnum**
- **MenuNodeIdTypeEnum**
- **MimeTypeEnum**
- **OrderEnum**
- **PostObjectFieldFormatEnum**
- **PostObjectsConnectionDateColumnEnum**
- **PostObjectsConnectionOrderbyEnum**
- **PostStatusEnum**
- **RelationEnum**

- **TaxonomyEnum**
- **TaxonomyIdTypeEnum**
- **TermNodeIdTypeEnum**
- **TermObjectsConnectionOrderbyEnum**
- **TimezoneEnum**
- **UserNodeIdTypeEnum**
- **UserRoleEnum**
- **UsersConnectionOrderbyEnum**
- **UsersConnectionSearchColumnEnum**

Inputs

- **DateInput**
- **DateQueryInput**
- **MenuItemsConnectionWhereArgs**
- **PostObjectsConnectionOrderbyInput**
- **UsersConnectionOrderbyInput**

Unions

- **ContentRevisionUnion**
- **MenuItemObjectUnion**
- **PostObjectUnion**
- **TermObjectUnion**

Connections

Connections are composed of many Types. Each connection registered creates a Type in the Schema named after the FromType and ToType, and creates additional Types for "Where" args to filter and sort the connections.

Connections are registered from Connected Types to the following.

For example, Connections are registered from the RootQuery to Comments, from Users to Comments, from Posts to Comments, etc.

- **Comments**
- **Commenter**
- **EnqueuedScripts**

- **EnqueuedStylesheets**
- **MediaItems**
- **Menus**
- **MenuItemLinkable**
- **MenuItems**
- **Plugins**
- **PostObjects**
- **Revisions**
- **Taxonomies**
- **TermObjects**
- **Themes**
- **Users**
- **UserRoles**

Mutations

WPGraphQL generates the following Mutations:

- **Comments**: `createComment`, `updateComment`, `deleteComment`, `restoreComment`
- **MediaItem**: `createMediaItem`, `updateMediaItem`, `deleteMediaItem`
- **User**: `createUser`, `deleteUser`, `updateUser`, `registerUser`, `resetUserPassword`, `sendPasswordResetEmail`
- **Settings:** `updateSettings`
- **Post Types:**
  - create, update and delete mutations are registered for each post type that's set to show in GraphQL.
- **Taxonomies:**
  - create, update and delete mutations are registered for each post type that's set to show in GraphQL

□□□ #1
□ □□□ □□□ 2026-08-29 23:36:26 UTC
□ □□□ □□□ 2026-08-29 23:36:26 UTC