gatsby-plugin-utils

Usage

```
npm install gatsby-plugin-utils
```

`validateOptionsSchema`

The `validateOptionsSchema` function verifies that the proper data types of options were passed into a plugin from the `gatsby-config.js` file. It is called internally by Gatsby to validate each plugin's options when a site is started.

Example

```
import { validateOptionsSchema } from "gatsby-plugin-utils"

await validateOptionsSchema(pluginName, pluginSchema, pluginOptions)
```

`testPluginOptionsSchema`

Utility to validate and test plugin options schemas. An example of a plugin options schema implementation can be found in the `gatsby-node.js` file of `gatsby-plugin-google-analytics` .

Example

```
// This is an example using Jest (https://jestjs.io/)
import { testPluginOptionsSchema } from "gatsby-plugin-utils"

it(`should partially validate one value of a schema`, async () => {
  const pluginSchema = ({ Joi }) =>
    Joi.object({
      someOtherValue: Joi.string(),
      toVerify: Joi.boolean(),
    })

  // Only the "toVerify" key of the schema will be verified in this test
  const { isValid, errors } = await testPluginOptionsSchema(pluginSchema, {
    toVerify: `abcd`,
  })

  expect(isValid).toBe(false)
  expect(errors).toEqual([`"toVerify" must be a boolean`])
})
```

`isGatsbyNodeLifecycleSupported`

Utility to be used by plugins to do runtime check against `gatsby` core package checking wether particular `gatsby-node` lifecycle API is supported. Useful for plugins to be able to support multiple `gatsby` core versions.

Example

```
const { isGatsbyNodeLifecycleSupported } = require(`gatsby-plugin-utils`)

// only use createSchemaCustomization lifecycle only when it's available.
if (isGatsbyNodeLifecycleSupported(`createSchemaCustomization`)) {
  exports.createSchemaCustomization = function createSchemaCustomization({
    actions,
  }) {
    // customize schema
  }
}
```

`hasFeature`

Feature detection is now part of Gatsby. As a plugin author you don't know what version of Gatsby a user is using. `hasFeature` allows you to check if the current version of Gatsby has a certain feature.

Here's a list of features: // TODO

Example

```
const { hasFeature } = require(`gatsby-plugin-utils`)

if (!hasFeature(`image-cdn`)) {
  // You can polyfill image-cdn here so older versions have support as well
}
```

Add ImageCDN support

Our new ImageCDN allows source plugins to lazily download and process images. if you're a plugin author please use this polyfill to add support for all Gatsby V4 versions.

For more information (see here)[https://gatsby.dev/img]

Example

```
const {
  addRemoteFilePolyfillInterface,
  polyfillImageServiceDevRoutes,
} = require(`gatsby-plugin-utils/pollyfill-remote-file`)

exports.createSchemaCustomization ({ actions, schema }) => {
  actions.createTypes([
    addRemoteFilePolyfillInterface(
      schema.buildObjectType({
        name: `PrefixAsset`,
        fields: {
          // your fields
        },
        interfaces: [`Node`, 'RemoteFile'],
      }),
      {
        schema,
        actions,
      }
    )
  ]);
}

/** @type {import('gatsby').onCreateDevServer} */
exports.onCreateDevServer = ({ app }) => {
  polyfillImageServiceDevRoutes(app)
}
```