Middleware and Helpers

Gatsby Functions provides an Express-like architecture that simplifies building Node.js APIs. We include a number of middlewares to parse common request data as well as response helpers.

Data formats

We parse commonly used data types. You can parse more by adding custom middleware. Data available by default on the `req` object:

- Cookies at `req.cookies`
- URL Queries (e.g. `api/foo?query=foo`) at `req.query`
- Form parameters and data at `req.body`
- JSON POST bodies at `req.body`
- Files uploaded from forms at `req.files`

Response helpers

- `res.send(body)` — returns the response. The `body` can be a `string`, `object`, or `buffer`
- `res.json(body)` — returns a JSON response. The `body` can be any value that can be serialized with `JSON.stringify()`
- `res.status(statusCode)` — set the HTTP status for the response. Defaults to 200.
- `res.redirect([statusCode], url)` — Returns a redirect to a URL. Optionally set the statusCode which defaults to 302.

Custom middleware

Custom Connect/Express middleware are supported.

An example of how to add CORS support to a Function:

```
import Cors from "cors"

const cors = Cors()

export default async function corsHandler(req, res) {
```

```
  // Run Cors middleware and handle errors.
  await new Promise((resolve, reject) => {
    cors(req, res, result => {
      if (result instanceof Error) {
        reject(result)
      }

      resolve(result)
    })
  })

  res.json(`Hi from Gatsby Functions`)
}
```

Custom body parsing

Generally useful for file upload support.

This is not yet supported. Add a comment in the discussion if this is an important use case for you.