

Node.js: Routers

Example: Simple GET Request

```
// in app.js
app.get( '/myFirstRoute', function(request, response) {
    response.send( '<body><h1>hello everyone!</h1></body>' );
});
```

Example: GET Request with query parameters

- query parameters are accessed from the `request.query` object

Exercise

- Create an Express application that has two routes.
- The first route should respond to GET requests at '/'. Respond with a greeting.
- The second route should respond to GET requests at: /messages.
- It should look for the message_1 and message_2 query parameters, then repeat the messages back to the user in the response.
- Bonus: Change the route to accept any number of messages.

Example: POST Request

- Middleware is needed to parse the post data.

Example: POST Request with JSON

Request Object

- `request.body`: POST request data (remember to use a body parser)
- `request.query`: URL query parameters
- `request.params`: URL parameters

Exercise

- Add a route to your previous Express application that accepts a POST request at '/messages'.
- Respond the same way: repeat the messages back. Test your route with cURL.
- Bonus: extract the code that responds into a function, then use that function in both the GET messages and POST messages routes.

Response Object: redirect

- `response.redirect`: hit specified route on server

```
response.redirect( ' /path/to/route ' );
```

Example: Static Content

Response Object (review)

- `response.send`: send function contents to client
- `response.render`: send function contents to view engine, render Jade template, send result to client

Dynamic Routes

- the expected route is: `localhost:3000/pickle/<param1>/<param2>`

RESTful Route Example (revisited)

