Cloud Computing

CS571 - SFBU

Technology: (Cloud Computing course)

Javascript and Node.js:

1. The problem was to create a node.js server and to return a object with the requested format.
2. The ppt with the response and code is attached.

Solution:

I added a new condition to check if the param in the route is of currenttime. If so, the current time is returned and formatted into a JSON object.

Design:

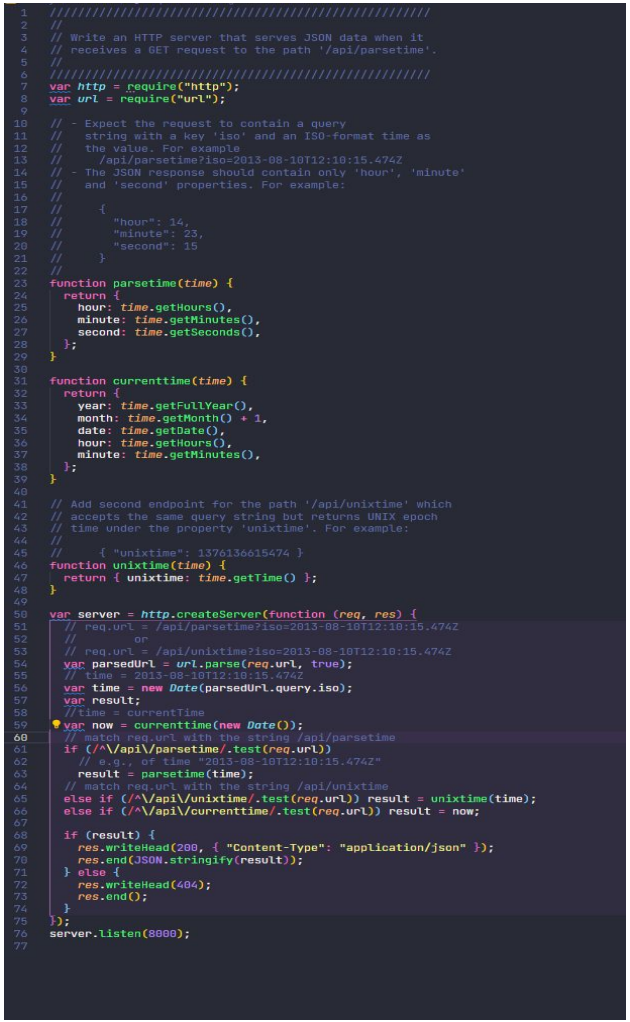
A new function of currenttime was also added to format to the requested format.

Modification:

The server is now listening to the port 8000, without needing to pass as CLI parameter.

Ownership:

The code developed was a improvement and modification of a given code. It was not developed 100% by me.



```
////////////////////////////////////////////////////
//
// Write an HTTP server that serves JSON data when it
// receives a GET request to the path '/api/parsetime'.
//
////////////////////////////////////////////////////
var http = require("http");
var url = require("url");

// - Expect the request to contain a query
//   string with a key 'iso' and an ISO-format time as
//   the value. For example
//     /api/parsetime?iso=2013-08-10T12:10:15.474Z
// - The JSON response should contain only 'hour', 'minute'
//   and 'second' properties. For example:
//
//      {
//        "hour": 14,
//        "minute": 23,
//        "second": 15
//      }
//
function parsetime(time) {
  return {
    hour: time.getHours(),
    minute: time.getMinutes(),
    second: time.getSeconds(),
  };
}

function currenttime(time) {
  return {
    year: time.getFullYear(),
    month: time.getMonth() + 1,
    date: time.getDate(),
    hour: time.getHours(),
    minute: time.getMinutes(),
  };
}

// Add second endpoint for the path '/api/unixtime' which
// accepts the same query string but returns UNIX epoch
// time under the property 'unixtime'. For example:
//
//      { "unixtime": 1376136615474 }
function unixtime(time) {
  return { unixtime: time.getTime() };
}

var server = http.createServer(function (req, res) {
  // req.url = /api/parsetime?iso=2013-08-10T12:10:15.474Z
  //                or
  // req.url = /api/unixtime?iso=2013-08-10T12:10:15.474Z
  var parsedUrl = url.parse(req.url, true);
  // time = 2013-08-10T12:10:15.474Z
  var time = new Date(parsedUrl.query.iso);
  var result;
  //time = currentTime
  var now = currenttime(new Date());
  // match req.url with the string /api/parsetime
  if (/^\/api\/parsetime/.test(req.url))
    // e.g., of time "2013-08-10T12:10:15.474Z"
    result = parsetime(time);
  // match req.url with the string /api/unixtime
  else if (/^\/api\/unixtime/.test(req.url)) result = unixtime(time);
  else if (/^\/api\/currenttime/.test(req.url)) result = now;

  if (result) {
    res.writeHead(200, { "Content-Type": "application/json" });
    res.end(JSON.stringify(result));
  } else {
    res.writeHead(404);
    res.end();
  }
});
server.listen(8000);
```

The solution I developed was a modification of the given code and implemented in a way that it is now able to check and return the current time.

The solution was improved and not developed from zero.