

NODE.JS - RESPONSE OBJECT

http://www.tutorialspoint.com/nodejs/nodejs_response_object.htm

Copyright © tutorialspoint.com

The **res** object represents the HTTP response that an Express app sends when it gets an HTTP request.

Response Object Properties

Following is the list of few properties associated with response object.

SN	Properties & Description
1	res.app This property holds a reference to the instance of the express application that is using the middleware.
2	res.headersSent Boolean property that indicates if the app sent HTTP headers for the response.
3	res.locals An object that contains response local variables scoped to the request

Response Object Methods

res.append(*field* [, *value*])

```
res.append(field [, value])
```

This method appends the specified value to the HTTP response header field. Following are few examples:

```
res.append('Link', ['<http://localhost/>', '<http://localhost:3000/>']);  
res.append('Set-Cookie', 'foo=bar; Path=/; HttpOnly');  
res.append('Warning', '199 Miscellaneous warning');
```

res.attachment(*filename*)

```
res.attachment([filename])
```

This method is used to send a file as an attachment in the HTTP response. Following are few examples:

```
res.attachment('path/to/logo.png');
```

res.cookie(*name*, *value* [, *options*])

```
res.cookie(name, value [, options])
```

This method is used to set cookie name to value. The value parameter may be a string or object converted to JSON. Following are few examples:

```
res.cookie('name', 'tobi', { domain: '.example.com', path: '/admin', secure: true });  
res.cookie('cart', { items: [1,2,3] });  
res.cookie('cart', { items: [1,2,3] }, { maxAge: 900000 });
```

res.clearCookie*name[, options]*

```
res.clearCookie(name [, options])
```

This method is used to clear the cookie specified by name. Following are few examples:

```
res.cookie('name', 'tobi', { path: '/admin' });  
res.clearCookie('name', { path: '/admin' });
```

res.download*path[, filename][, fn]*

```
res.download(path [, filename] [, fn])
```

This method is used to transfer the file at path as an "attachment". Typically, browsers will prompt the user for download. Following are few examples:

```
res.download('/report-12345.pdf');  
res.download('/report-12345.pdf', 'report.pdf');  
res.download('/report-12345.pdf', 'report.pdf', function(err){  
  });
```

res.end*[data][, encoding]*

```
res.end([data] [, encoding])
```

This method is used to end the response process. Following are few examples:

```
res.end();  
res.status(404).end();
```

res.format*object*

```
res.format(object)
```

This method is used to performs content-negotiation on the Accept HTTP header on the request object, when present. Following are few examples:

```
res.format({  
  'text/plain': function(){  
    res.send('hey');  
  },  
  
  'text/html': function(){  
    res.send('hey')  
  },  
  
  'application/json': function(){  
    res.send({ message: 'hey' });  
  },  
  
  'default': function() {  
    // log the request and respond with 406  
    res.status(406).send('Not Acceptable');  
  }  
})
```

```
});
```

res.get*field*

```
res.get(field)
```

This method is used to return the HTTP response header specified by field. Following are few examples:

```
res.get('Content-Type');
```

res.json*[body]*

```
res.json([body])
```

This method is used to send a JSON response. Following are few examples:

```
res.json(null)
res.json({ user: 'tobi' })
res.status(500).json({ error: 'message' })
```

res.jsonp*[body]*

```
res.jsonp([body])
```

This method is used to send a JSON response with JSONP support. Following are few examples:

```
res.jsonp(null)
res.jsonp({ user: 'tobi' })
res.status(500).jsonp({ error: 'message' })
```

res.links*links*

```
res.links(links)
```

This method is used to join the links provided as properties of the parameter to populate the response's Link HTTP header field. Following are few examples:

```
res.links({
  next: 'http://api.example.com/users?page=2',
  last: 'http://api.example.com/users?page=5'
});
```

res.location*path*

```
res.location(path)
```

This method is used to set the response Location HTTP header field based on the specified path parameter. Following are few examples:

```
res.location('/foo/bar');
res.location('foo/bar');
res.location('http://example.com');
```

res.redirect*[status,]path*

```
res.redirect([status, ] path)
```

This method is used to redirect to the URL derived from the specified path, with specified HTTP status code status. Following are few examples:

```
res.redirect('/foo/bar');
res.redirect('http://example.com');
res.redirect(301, 'http://example.com');
```

res.renderview[, locals][, callback]

```
res.render(view [, locals] [, callback])
```

This method is used to render a view and sends the rendered HTML string to the client. Following are few examples:

```
// send the rendered view to the client
res.render('index');

// pass a local variable to the view
res.render('user', { name: 'Tobi' }, function(err, html) {
  // ...
});
```

res.send[body]

```
res.send([body])
```

This method is used to send the HTTP response. Following are few examples:

```
res.send(new Buffer('whoop'));
res.send({ some: 'json' });
res.send('

some html

');
```

res.sendFilepath[, options][, fn]

```
res.sendFile(path [, options] [, fn])
```

This method is used to transfer the file at the given path. Sets the Content-Type response HTTP header field based on the filename's extension. Following are few examples:

```
res.sendFile(fileName, options, function (err) {
  // ...
});
```

res.sendStatusstatusCode

```
res.sendStatus(statusCode)
```

This method is used to set the response HTTP status code to statusCode and send its string representation as the response body. Following are few examples:

```
res.sendStatus(200); // equivalent to res.status(200).send('OK')
res.sendStatus(403); // equivalent to res.status(403).send('Forbidden')
res.sendStatus(404); // equivalent to res.status(404).send('Not Found')
res.sendStatus(500); // equivalent to res.status(500).send('Internal Server Error')
```

res.setfield[, value]

```
res.set(field [, value])
```

This method is used to set the response's HTTP header field to value. Following are few examples:

```
res.set('Content-Type', 'text/plain');

res.set({
  'Content-Type': 'text/plain',
  'Content-Length': '123',
  'ETag': '12345'
})
```

res.statusCode

```
res.status(code)
```

This method is used to set the HTTP status for the response. Following are few examples:

```
res.status(403).end();
res.status(400).send('Bad Request');
res.status(404).sendFile('/absolute/path/to/404.png');
```

res.type

```
res.type(type)
```

This method is used to set the Content-Type HTTP header to the MIME type. Following are few examples:

```
res.type('.html');           // => 'text/html'
res.type('html');           // => 'text/html'
res.type('json');           // => 'application/json'
res.type('application/json'); // => 'application/json'
res.type('png');            // => image/png:
```

```
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```