## Chapter 3
## Express.js Request Object

Express.js Request and Response objects are the parameters of the callback function which is used in Express applications.

The express.js request object represents the HTTP request and has properties for the request query string, parameters, body, HTTP headers, and so on.

Syntax:

```
app.get('/', function (req, res) {
  // --
})
```

//……………………………………………………………………………………………………

Express.js Request Object Properties

The following table specifies some of the properties associated with request object.

1. req.app:- This is used to hold a reference to the instance of the express application that is using the middleware.

2. req.baseurl:- It specifies the URL path on which a router instance was mounted.

3. req.body:- It contains key-value pairs of data submitted in the request body. By default, it is undefined, and is populated when you use body-parsing middleware such as body-parser.

4. req.cookies:- When we use cookie-parser middleware, this property is an object that contains cookies sent by the request.

5. req.fresh:- It specifies that the request is "fresh." it is the opposite of req.stale.

6. req.hostname:- It contains the hostname from the "host" http header.

7. req.ip:- It specifies the remote IP address of the request.

8. req.ips:- When the trust proxy setting is true, this property contains an array of IP addresses specified in the ?x-forwarded-for? request header.

9. req.originalurl:- This property is much like req.url; however, it retains the original request URL, allowing you to rewrite req.url freely for internal routing purposes.

10. req.params:- An object containing properties mapped to the named route ?parameters?. For example, if you have the route /user/:name, then the "name" property is available as req.params.name. This object defaults to {}.

11. req.path:- It contains the path part of the request URL.

12. req.protocol:- The request protocol string, "http" or "https" when requested with TLS.

13. req.query:- An object containing a property for each query string parameter in the route

14. req.route:- The currently-matched route, a string.

15. req.secure:- A Boolean that is true if a TLS connection is established.

16. req.signedcookies:- When using cookie-parser middleware, this property contains signed cookies sent by the request, unsigned and ready for use.

17. req.stale:- It indicates whether the request is "stale," and is the opposite of req.fresh.

18. req.subdomains:- It represents an array of subdomains in the domain name of the request.

19. req.xhr:- A Boolean value that is true if the request's "x-requested-with" header field is "xmlhttprequest", indicating that the request was issued by a client library such as jQuery

//......................................................................................................................................

Request Object Methods

Following is a list of some generally used request object methods:

req.accepts (types)

This method is used to check whether the specified content types are acceptable, based on the request's Accept HTTP header field.

Examples:

```
req.accepts('html');
//=>?html?
req.accepts('text/html');
// => ?text/html?
```

//..........................................................................................................................................

req.get(field)

This method returns the specified HTTP request header field.

Examples:

```
req.get('Content-Type');
// => "text/plain"
req.get('content-type');
// => "text/plain"
req.get('Something');
// => undefined
```

//..........................................................................................................................................

req.is(type)

This method returns true if the incoming request's "Content-Type" HTTP header field matches the MIME type specified by the type parameter.

Examples:

```
// With Content-Type: text/html; charset=utf-8
req.is('html');
req.is('text/html');
req.is('text/*');
// => true
```

req.param(name [, defaultValue])

This method is used to fetch the value of param name when present.

Examples:

```
// ?name=sasha
req.param('name')
// => "sasha"
// POST name=sasha
req.param('name')
// => "sasha"
// /user/sasha for /user/:name
req.param('name')
// => "sasha"
```