**Query Parameter**

* Query parameters are the key-value pairs that appear after the "?" in the URL.
* Each query parameters are separated with “&”
* Query parameters are accessed via the **req.query** object.

**URL:** <http://localhost:3000/search?pname=samsung&price=2000>

req.query

{

pname: 'samsung',

price: '2000'

}

Example#1

const express = require('express');

const app = express();

app.get('/search', (req, res)=> {

    var pname = req.query.pname;

    var price = req.query.price;

    res.send('product name : ' + pname+'\nproduct price : ' + price);

});

app.listen('3000', ()=> {

  console.log('Server Started');

  console.log(`Running at http://localhost:3000`);

});

Example#2

const express = require('express');

const app = express();

app.get('/search', (req, res)=> {

    const { pname, price } = req.query;

    res.send('product name : ' + pname+'\nproduct price : ' + price);

});

app.listen('3000', ()=> {

  console.log('Server Started');

  console.log(`Running at http://localhost:3000`);

});

**Path Parameter**

* Path parameters are part of the URL's path.
* Path parameters are specified within the route's URL path by prefixing with a colon “:”.
* Path parameters are accessed via the **req.params** object.

**URL:** <http://localhost:3000/search/samsung/20000>

Example#3

const express = require('express');

const app = express();

app.get('/search/:pname/:price', (req, res)=> {

    const pname = req.params.pname;

    const price = req.params.price;

    res.send(`product name : ${pname} \nprice : ${price}`);

});

app.listen('3000', ()=> {

  console.log('Server Started');

  console.log(`Running at http://localhost:3000`);

});

Example#4

const express = require('express');

const app = express();

app.get('/search/:pname/:price', (req, res)=> {

    const { pname, price } = req.params;

    res.send(`product name : ${pname} \nprice : ${price}`);

});

app.listen('3000', ()=> {

  console.log('Server Started');

  console.log(`Running at http://localhost:3000`);

});

**Sending Data in Post Request**

* Middleware is a function that has access to the request and response object.
* express.urlencoded() middleware function is used to parse form data from incoming requests.
* express.json() middleware function is used to parse JSON data from incoming requests.

Example#5

const express = require('express');

const app = express();

app.use(express.urlencoded({ extended: true }));

app.post('/insert', (req, res)=> {

    const pname = req.body.pname;

    const price = req.body.price;

    res.send(`product name : ${pname} \nprice : ${price}`);

});

app.listen('3000', ()=> {

  console.log('Server Started');

  console.log(`Running at http://localhost:3000`);

});

Example#6

const express = require('express');

const app = express();

app.use(express.json());

app.post('/insert', (req, res)=> {

    const pname = req.body.pname;

    const price = req.body.price;

    res.send(`product name : ${pname} \nprice : ${price}`);

});

app.listen('3000', ()=> {

  console.log('Server Started');

  console.log(`Running at http://localhost:3000`);

});