

ExpressJs

```
const express = require('express');
const app = express();
const path = require('path');
const hbs = require('hbs'); //handlebars
const { request } = require('http');
const requests = require('requests');
const staticpath = path.join(__dirname, '../public');
const templatepath = path.join(__dirname, '../template');
const partialpath = path.join(__dirname, '../src/views/partials');
```

```
// built in middleware(this is imp for css if the page is dynamic)
app.use(express.static(staticpath))
// console.log(path.join(__dirname, '../src/views/partials'));
```

```
// to set view engine
app.set("view engine", "hbs");
```

```
// partials creation Decalration of path
hbs.registerPartials(partialpath);
```

```
// renaming views folder
// app.set('views', templatepath);
```

```
// templete engine route
app.get("/", (req, res) => {
  res.render("index.hbs");
});
```

```
// This is about us page with hbs
app.get("/about", (req, res) => {
  res.render("about.hbs");
});
```

```
// this is dynamic data plz comment out line 17,18,19 to see the result.
(Passing values to dynamic page can be done in this way)
```

```
app.get("/", (req, res) => {
  res.render("index.hbs", {
    name: 'sumeet'
  });
});
```

```
app.get('/', (req, res) => {
  res.send('hello ')
})
app.get('/watch', (req, res) => {
```

```

    res.status(200);
    res.send('<h1>this is watch page</h1>')
  })
  app.get('/about', (req, res) => {
    res.status(200);
    res.send('this is about us page')
  })
  app.get('/contact', (req, res) => {
    res.status(200);
    res.send('this is contact us page')
  })
  app.get('/city', (req, res) => {
    res.status(200);
    requests(`http://api.openweathermap.org/data/2.5/weather?q=${
    {req.query.name}
    &units=metric&appid=a581b5a9f443f697a406c557726430ba`)
    .on("data", (chunk) => {
      const objdata = JSON.parse(chunk);
      const arrData = [objdata];
      console.log(`city name is ${arrData[0].name} and the temp is : $
      {arrData[0].main.temp}`);
      res.write(arrData[0].name);

    })
    .on("end", (err) => {
      if (err) return console.log("connection closed due to errors", err);
      res.end();
    });
  })
  // This is Error page
  app.get('*', (req, res) => {
    res.status(404);
    res.render("error");
  })

  // app port
  app.listen(3000, () => {
    console.log('listening to port 3000')
  })

```

// This will be output of the temp in console just comment out /city url and type
(<http://localhost:3000/city?name=delhi>) in url to get this result in console

```
// [nodemon] restarting due to changes...  
// [nodemon] starting `node app.js`  
// listening to port 3000  
// city name is Delhi and the temp is : 15.05
```