

Views in MVC

Views define the UI and the data of a website that will be displayed on the screen to the user.

index.js

```
const express = require('express');
const path = require('path');
const port = 8000;
const app = express();

app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

app.get('/', function(req, res){
    return res.render('home');
});

app.listen(port, function (err) {
    if (err) {
        console.log("Oops, error! ", err);
        return;
    }
    console.log("Server running on the port, ", port);
});
```

app.set() is used to set a property and its value of the app object.

- `app.set('view engine', 'ejs')` sets a property 'view engine' with a value 'ejs' to the app object. It tells the server that the template engine that you'll be using is ejs.
- `app.set('views', path.join(__dirname), 'views')` sets a property 'views' with a value equal to a path which is `path.join(__dirname, 'views')`. It tells the server to look into the 'views' folder in the current directory to render any required web document.

So when `res.render('home')` is called, the server directly looks into the

'views' folder for the 'home' page. That is why we **don't** have to specify the full path as `res.render('./views/home')`.

`path.join(__dirname, 'views')`

If you don't specify `app.set('views', path.join(__dirname, 'views'))`; then the server by **default** tries to find if there is any 'views' folder available in the current directory.

- If the 'views' folder is available then it'll automatically look for the required web document in it.
- Else it'll look for `app.set('views', path.join(__dirname, 'views_folder_name'))`;
Here, the 'views' property of the app is set to the specified path of the folder 'views_folder_name'. This folder name can be anything and it'll contain your web documents i.e. views.
This folder will be searched for any required views.

This is why `app.set('views', path.join(__dirname, 'views_folder_name'))` is important to specify. It helps the server know where it has to look in for the web documents.

Notes:

- We should always specify `app.set('views', path.join(__dirname, 'views'))` as a best practice and not rely on the default folder search.
- Although you can keep the name of the views folder as you want, it's preferable to keep a meaningful name for example 'views'.