

BSc (Hons) in Information Technology
Specializing in Software Engineering
Year 3 - 2021
SE3040 – Application Frameworks
Lab 05 – Koa.js

1. Init a npm project.
2. Install Koa “koa”. Save it as a dev dependency.
3. Create a file named server.js and require “koa” dependency to it.
4. Create a Koa web server that returns ‘Hello World’ to all requests.

```
const app = new Koa();  
app.use(ctx => {  
  ctx.body = 'Hello World';  
});
```

```
app.listen(3000);
```

```
console.log('Application is running on port 3000');
```

5. Run your application. Perform HTTP requests to <http://localhost:3000> and verify the output. You should see “Hello World” plain text coming.
6. Install “@koa/router” to your project.
7. Create a new file named [home.router.js](#) inside the directory named routes. Import “@koa/router”.
8. Inside the new file, register following routes.

```
const router = new Router({  
  prefix: '/home'  
});
```

```
router.get('/', ctx => {  
  ctx.body = 'Hello World';  
});
```

```
router.post('/', ctx => {  
  ctx.body = 'Hello World';  
});
```

9. Export the “router” object you created in the above.
10. Import this file to your server.js file.
11. You need your server to understand JSON type that can be achieved by using body parser.
 - a. Install koa-bodyparser
 - b. Register this with your koa app.

```
app.use(bodyParser());
```
12. Register these routes with your Koa application. You need to allow HTTP methods while registering. This should be executed before listening to port 3000. You need to remove or put the below code before the previously created app.use in 4 (which returned Hello World to any request) or remove that part entirely. This is because that global listener will reply before your specific router capture the request. You should understand the router is imported as “HomeRoutes” in the below.

```
app.use(HomeRoutes.routes())  
  .use(HomeRoutes.allowedMethods());
```

BSc (Hons) in Information Technology
Specializing in Software Engineering
Year 3 - 2021
SE3040 – Application Frameworks
Lab 05 – Koa.js

13. Try out your newly created two routes prefix with <http://localhost:3000/home>
14. Next create a new file called “posts.api.js”. This will be your application logic. This should be under the directory “api”.
15. Install and import the dependency “uuid” to your newly created file.
16. Your “posts.api.js” should support below. Use your basic Node and JS knowledge to achieve this.
 - a. Create a global Map called “posts” to hold posts created by users.
 - b. Create a method “createPost” that accepts an object with properties (name, description).
 - c. This method should add new post passed to it to the Map by adding two new properties “id” and “postedDate”.
 - i. ID should be created using imported UUID library method “uuid.v4()”.
 - ii. Posted Date should be set using the JS Date class “new Date()”.
 - iii. Map has a method called set that can be used to add a new key and value to it. Use the created ID field as the key., value should be the complete post object.
 - d. Add another method “getPosts” that returns all posts created.
 - i. You should use the methos called “values()” in map.
 1. Try the spread syntax to expand map values as an array “[...map.values()]”.
 - e. Other method “getPost” should accept the argument “id”. This should return the post with that ID. Use the map method “get(key)” method to fetch the object by your key (ID).
 - f. Make sure you export all these methods.
17. Now, create a new file “post.routes.js”. Same as before import router and the 3 method you create in the previous exercise.

```
const router = new Router({
  prefix: '/posts'
});

router.get('/', ctx => {
  ctx.body = getPosts();
});

router.post('/', ctx => {
  let post = ctx.request.body;
  post = createPost(post);
  ctx.response.status = 201;
  ctx.body = post;
});

router.get('/:id', ctx => {
  const id = ctx.params.id;
  ctx.body = getPost(id);
});
```

18. Export the router and import that in the “server.js” as same before.
19. Try creating, get all and get one by ID routes.
 - a. <http://localhost:3000/posts> POST
 - b. <http://localhost:3000/posts> GET



BSc (Hons) in Information Technology
Specializing in Software Engineering
Year 3 - 2021
SE3040 – Application Frameworks
Lab 05 – Koa.js

- c. <http://localhost:3000/posts/{id}> GET