

# Exercise



**Mosh Hamedani**  
[programmingwithmosh.com](https://programmingwithmosh.com)



100

JS solution.js ✕

JS exercise is 12 Exercise/async-demo

## > 01 Synchro

## > 02. Patterns For Dealing With Asynchron

## > 05. Named Functions

## 07 Replacing Cal

## > 08. Consuming Promises

## > 10. Running Parallel Pro

- ▼ async-demo

```
{} package.json
```

① README.md

```

27 function getCustomer(id) {
28   return new Promise((resolve, reject) => {
29     setTimeout(() => {
30       resolve({
31         id: 1,
32         name: 'Mosh Hamedani',
33         isGold: true,
34         email: 'email'
35       });
36     }, 4000);
37   });
38 }
39

```

```

40 function getTopMovies() {
41   return new Promise((resolve, reject) => {
42     setTimeout(() => {
43       resolve(['movie1', 'movie2']);
44     }, 4000);
45   });
46 }
47

```

```
Complexity is 4 Everything is cool!
```

```
48 function sendEmail(email, movies) {
49     return new Promise((resolve, reject) => {
50         setTimeout(() => {
51             resolve();
52         }, 4000);
53     });
54 }
```

```
dandy@My-Ubuntu:~/Keep/GitHub Repositories/Node.js/06. Asynchronous JavaScript/Applications/12. Exercise/async-demo$ node solution.js
Customer: { id: 1, name: 'Mosh Hamedani', isGold: true, email: 'email' }
Top movies: [ 'movie1', 'movie2' ]
Email sent...
```

Ln 54, Col 4 Spaces: 4 UTF-8 LF Babel JavaScript JavaScript Standard Style  