

JavaScript



```
// PEHLE aisa hoga:
// router.get('/reports', (req, res) => { ... });

// ISKO CHANGE KARKE AISHA KAREIN:
router.get('/reports', isAdmin, (req, res) => {
  // Aapka report download/view logic
  const filePath = path.join(__dirname, '../public/reports/conference_report.pdf')
  res.download(filePath);
});
```

Haan, `data` yahan par ek normal (custom) JavaScript variable hi hai.

Aap is variable ka naam apni marzi se kuch bhi rakh sakte hain (jaise `confData` , `yearDetails` , etc.).

Is code me `data` do alag-alag jagah par use ho raha hai:

2. `res.render()` ke andar (`{ data }`)

Jab aap Express me `res.render('conference', { data })` likhte hain, toh yeh **ES6 Object Shorthand Syntax** hota hai.

Iska poora matlab yeh hota hai:

JavaScript



```
res.render('conference', { data: data });
```

JavaScript



```
res.render('conference', { data: data });
```

- **Left side wala data** : Yeh Key Name hai jo EJS file ko milega. Isiliye EJS file me aap `<%= data.description %>` ya `data.agenda.day1` karke use karte hain.
- **Right side wala data** : Yeh aapka Custom Variable hai jisme backend ka data store tha.

`conferenceData[year]` ka Logic:

Jab URL me koi user `/conference/2026` par jayega:

1. `req.params.year` se `year` variable ki value `"2026"` milegi.
2. `conferenceData[year]` actual me `conferenceData["2026"]` ban jayega.
3. Isse `data` variable me sirf **2026 ka object**
`{ title: "2026 Conference", location: "Goa" }` assign ho jayega.

Aap bilkul sahi pakde hain! Data actual me `data` key ke roop me hi EJS ko mil raha hai, lekin iska karan Express ka ek shorthand syntax hai.

Code me ho kya raha hai, use step-by-step samajhiye:

JavaScript ES6 Shorthand Syntax

Aapke code me last line yeh likhi hui hai:

JavaScript



```
res.render('conference', { data });
```

JavaScript ES6 me jab **Object Key Name** aur **Variable Name** dono same hote hain, toh hume `{ data: data }` pura likhne ki zaroorat nahi padti. Hum simply `{ data }` likhte hain.

Backend is line ko internally aise hi treat karta hai:

JavaScript



```
res.render('conference', { data: data });  
//                ^^^^  ^^^^  
//                Key   Variable Value
```

Agar hum Key Name badal dein:

Agar aap backend me key name `conferenceInfo` kar dein:

JavaScript



```
// Backend Route  
res.render('conference', { conferenceInfo: data });
```

Tab EJS file me `data` kaam nahi karega. Phir aapko EJS file me key name ke hisaab se likhna padega:

Code snippet



```
<!-- EJS Template -->  
<h1><%= conferenceInfo.title %></h1>  
<p><%= conferenceInfo.description %></p>
```

Summary

- `data` **Key Hai:** EJS ko hamesha `{ key: value }` ki **Key** milti hai.
- **Shorthand Magic:** `{ data }` likhne ki wajah se key ka naam bhi `data` hi ban gaya hai.



```
1
2
3      <ul>
4        <% data.agenda.day1.forEach(function(item) { %>
5          <li><%= item %></li>
6        <% }); %>
7      </ul>
8
```

data.agenda.day1 --- yaha value kaise pass kar rahe hai

```
1
2
3
4 router.get('/:year', (req, res) => {
5   const year = req.params.year;
6   let data = conferenceData[year];
7
8   if (!data) {
9     return res.status(404).send('Conference year not found');
10  }
11
12  if (year === '2026') {
13    data = {
14      ...data,
15      agenda: getAgendaData(),
16      photos: getGalleryPhotos()
17    };
18  }
19
20  res.render('conference', { data });
21 });
22
```

```
1
2
3
4 router.get('/:year', (req, res) => {
5   const year = req.params.year;
6   let kamlu = conferenceData[year];
7
8   if (!kamlu) {
9     return res.status(404).send('Conference year not found');
10  }
11
12  if (year === '2026') {
13    kamlu = {
14      ...kamlu,
15      agenda: getAgendaData(),
16      photos: getGalleryPhotos()
17    };
18  }
19
20  res.render('conference', { data: kamlu });
21 });
22
```