

■ Express.js Request Parameters vs Request Body

In Express.js, data can be extracted from different parts of an HTTP request. The most common ways are via **URL parameters** (`req.params`), **wildcards** (`*`), and **request body** (`req.body`). This note explains the difference clearly.

1. Using `req.params` and Wildcards (`*`) - Express route parameters are defined in the URL path.
 - The ``*`` acts as a wildcard, and its value is available in **`req.params[0]`**.

Example:

```
app.get('/name/*', (req, res) => {  
  console.log(req.params[0]); // value after /name/  
  res.send("You requested: " + req.params[0]);  
});
```

Test URLs:

- `/name/raj` → `req.params[0] = "raj"`
- `/name/raj/chaudhary` → `req.params[0] = "raj/chaudhary"`

2. Named Parameters (`:param`) Instead of using `*`, you can define named parameters.

Example:

```
app.get('/name/:username', (req, res) => {  
  console.log(req.params.username);  
  res.send("Hello " + req.params.username);  
});
```

- `/name/rajendra` → `req.params.username = "rajendra"`

3. Using `req.body` - `req.body` is used to extract data sent in the body of a POST/PUT request.

- It requires middleware:

```
app.use(express.urlencoded({ extended: true })); // For form data  
app.use(express.json()); // For JSON data
```

Example:

```
app.post('/name', (req, res) => {  
  console.log(req.body.username);  
  res.send("Hello " + req.body.username);  
});
```

HTML Form:

```
<form action="/name" method="POST">  
  <input type="text" name="username" placeholder="Enter name">  
  <button type="submit">Submit</button>  
</form>
```

4. Comparison
Method: `req.params[0]` (URL wildcard `*`) vs `req.params[0]` (named param `:param`) vs `req.params.username` (named param `:username`) vs `req.body.username` (request body)
Source: `req.params[0]` (URL wildcard `*`) vs `req.params[0]` (named param `:param`) vs `req.params.username` (named param `:username`) vs `req.body.username` (request body)
Access: `req.params[0]` (URL wildcard `*`) vs `req.params[0]` (named param `:param`) vs `req.params.username` (named param `:username`) vs `req.body.username` (request body)
Example: `req.params[0]` (URL wildcard `*`) vs `req.params[0]` (named param `:param`) vs `req.params.username` (named param `:username`) vs `req.body.username` (request body)
Request: `req.params[0]` (URL wildcard `*`) vs `req.params[0]` (named param `:param`) vs `req.params.username` (named param `:username`) vs `req.body.username` (request body)
Body: `req.params[0]` (URL wildcard `*`) vs `req.params[0]` (named param `:param`) vs `req.params.username` (named param `:username`) vs `req.body.username` (request body)
Type: `req.params[0]` (URL wildcard `*`) vs `req.params[0]` (named param `:param`) vs `req.params.username` (named param `:username`) vs `req.body.username` (request body)
Format: `req.params[0]` (URL wildcard `*`) vs `req.params[0]` (named param `:param`) vs `req.params.username` (named param `:username`) vs `req.body.username` (request body)
JSON: `req.params[0]` (URL wildcard `*`) vs `req.params[0]` (named param `:param`) vs `req.params.username` (named param `:username`) vs `req.body.username` (request body)

■ Summary:

- Use **req.params** for values in the URL path.
- Use **wildcard (*)** when you don't know the depth of the path.
- Use **req.body** for data sent in POST/PUT requests (form or JSON).