

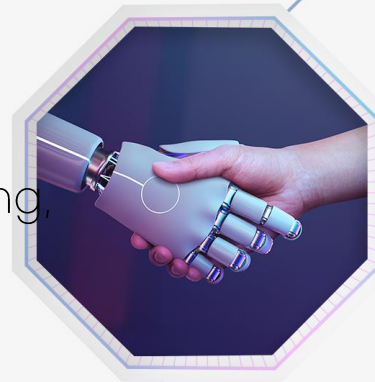


Express.js

Web Programming

Sung-Dong Kim,

■ School of Computer Engineering
Hansung University





◆ 미들웨어 사용하기

◆ 라우터 사용하기

미들웨어 사용하기



- ◆ 개발자가 다양한 기능을 사용할 수 있도록 여러 가지 미들웨어 제공

◆ static

- » 설치 `npm install serve-static --save`

- » 특정 폴더의 파일들을 특정 path로 접근할 수 있게 함

- » [public] 폴더의 모든 파일을 웹 서버의 root path ("/")로 접근하게 함

```
const static = require('serve-static');  
app.use('/', static(path.join(__dirname, 'public')));
```

- » 1st 파라미터 = 요청 패스 : 2nd 파라미터 = 특정 폴더 지정 → 요청 패스와 폴더가 mapping됨



body-parser

- » 설치 `npm install body-parser --save`
- » POST로 요청했을 때 요청 파라미터를 확인할 수 있게 함
- » 본문 영역에 있는 요청 파라미터를 파싱 → `req.body` 에 넣어줌

```
app.use(bodyParser.urlencoded({ extended: false }));  
app.use(bodyParser.json());
```



login1.html

```
<form action="" method="POST">
  <table>
    <tr>
      <td><label for="">아이디</label></td>
      <td><input type="text" name="id"></td>
    </tr>
    <tr>
      <td><label for="">비밀번호</label></td>
      <td><input type="password" name="password"></td>
    </tr>
  </table>
  <input type="submit" value="전송">
</form>
```





app8.js

...

```
const bodyParser = require('body-parser');  
app.use(bodyParser.urlencoded({ extended: false }));  
app.use(bodyParser.json());
```

```
app.use(express.json());  
app.use(express.urlencoded({ extended: true }));
```

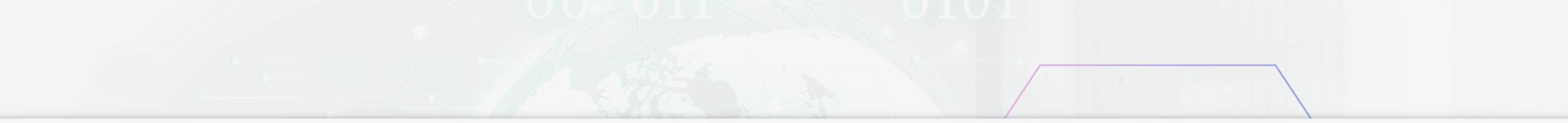
```
const path = require('path');  
const static = require('serve-static');  
app.use('/', static(path.join(__dirname, 'public')));
```

```
app.use(  
  express.static(path.join(__dirname, 'public'));
```



```
app.use(function(req, res, next) {  
  const paramId = req.body.id || req.query.id;  
  const paramPassword = req.body.password || req.query.password;  
  
  res.status(200).send(  
    `<h1>서버에서 응답한 결과</h1>  
    <div><p>Param id : ${paramId}</p></div>  
    <div><p>Param password : ${paramPassword}</p></div>  
    <br><br><a href="/login1.html"> 로그인 페이지로 돌아가기</a>  
    `);  
});
```





라우터 사용하기



◆ 라우터 미들웨어: 요청 url을 일일이 확인하는 문제 해결

◆ 사용자가 요청한 기능을 path 기준으로 구별

◆ 라우터 객체 참조

```
const router = express.Router();
```

◆ 라우팅 함수 등록

```
router.route('/process/login').get(...);  
router.route('/process/login').post(...);
```

```
router.route(요청 path).get(실행될 함수);  
router.route(요청 path).post(실행될 함수);
```

◆ app 객체에 등록

```
app.use('/', router);
```

요청 패스를 라우터 객체에 등록할 때 사용하는 메소드

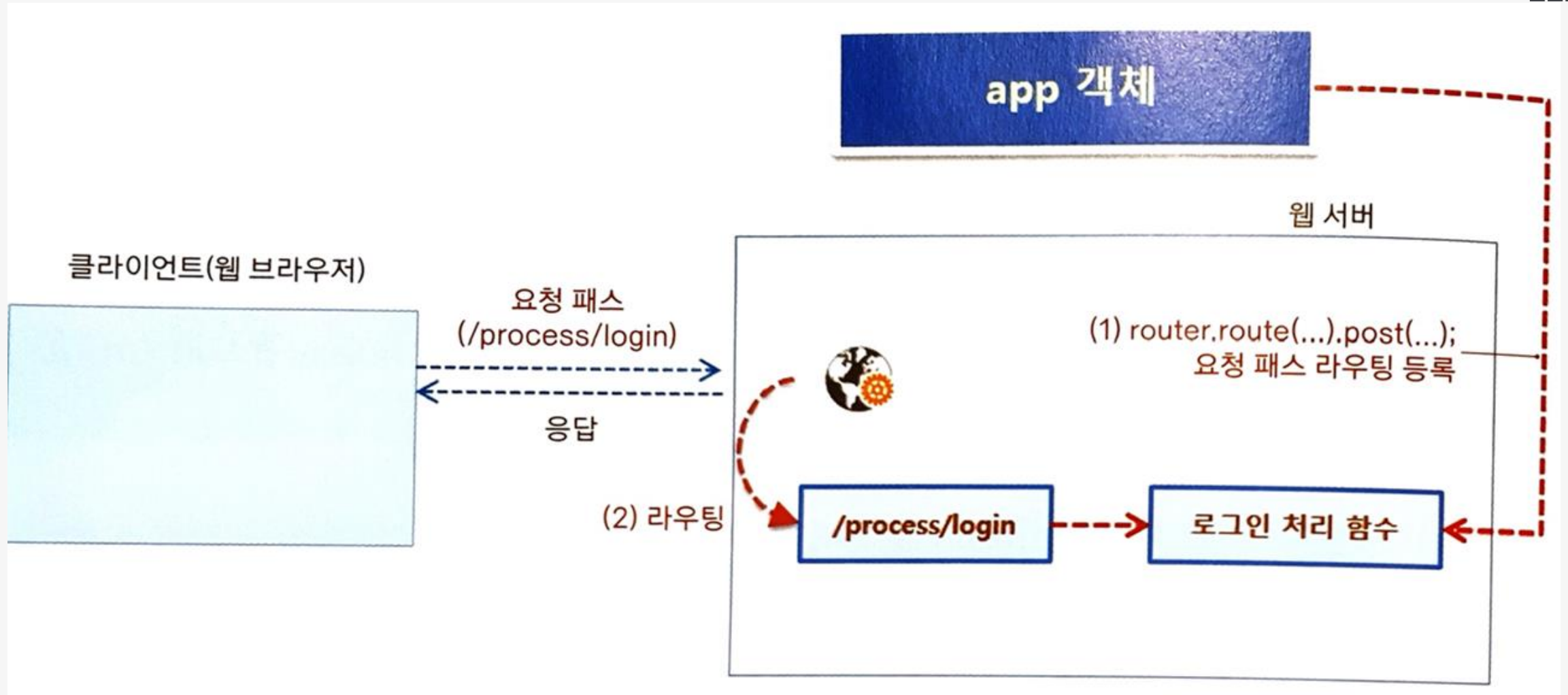
method	설명
get(callback)	GET 방식으로 특정 패스 요청 발생 시 사용할 callback 함수 지정
post(callback)	POST 방식으로 특정 패스 요청 발생 시 사용할 callback 함수 지정
put(callback)	PUT 방식으로 특정 패스 요청 발생 시 사용할 callback 함수 지정
delete(callback)	DELETE 방식으로 특정 패스 요청 발생 시 사용할 callback 함수 지정
all(callback)	모든 요청 방식 처리

login2.html: `<form method="POST" action="/process/login">`

```
const router = express.Router();
```

app9.js

```
router.route('/process/login').post(function(req, res) {  
  console.log('/process/login 처리');  
  ...  
  res.write('<br><br><a href="/login2.html">login 페이지로 돌아가기</a>');  
  res.end();  
});  
app.use('/', router);
```



◆ URL parameter: URL 주소의 일부로 포함됨

» login3.html: `<form method="POST" action="/process/login/:name"`

```
router.route('/process/login/:name').post(function(req, res) {  
  console.log('/process/login/:name 처리');  
  const paramName = req.params.name;  
  ...  
  res.write('<div><p>Param Name : ${paramName}</a></div>');  
  ...  
  res.write('<br><br><a href="/login3.html">login 페이지로 돌아가기</a>');  
  res.end();  
});
```

app10.js

학습 정리



- ◆ 미들웨어 사용하기: static, body-parser
- ◆ 라우터 사용하기