

## Node.js HTTP 통신 2



2016년도 2학기 2학년 방과후학교

## 정적 파일 요청

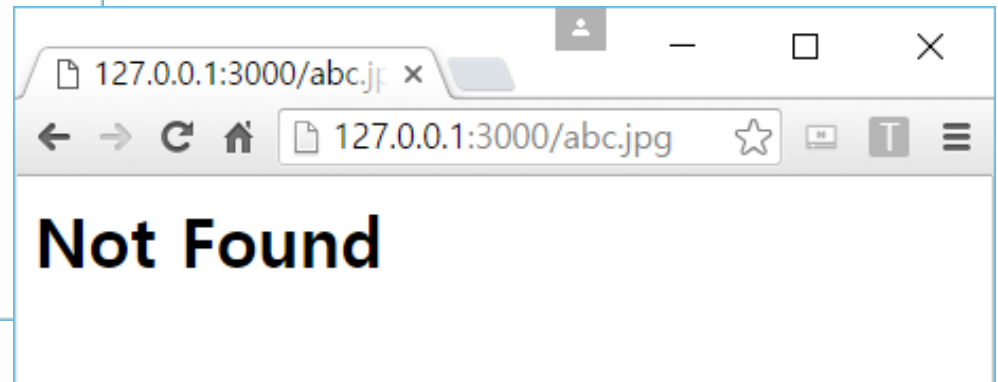
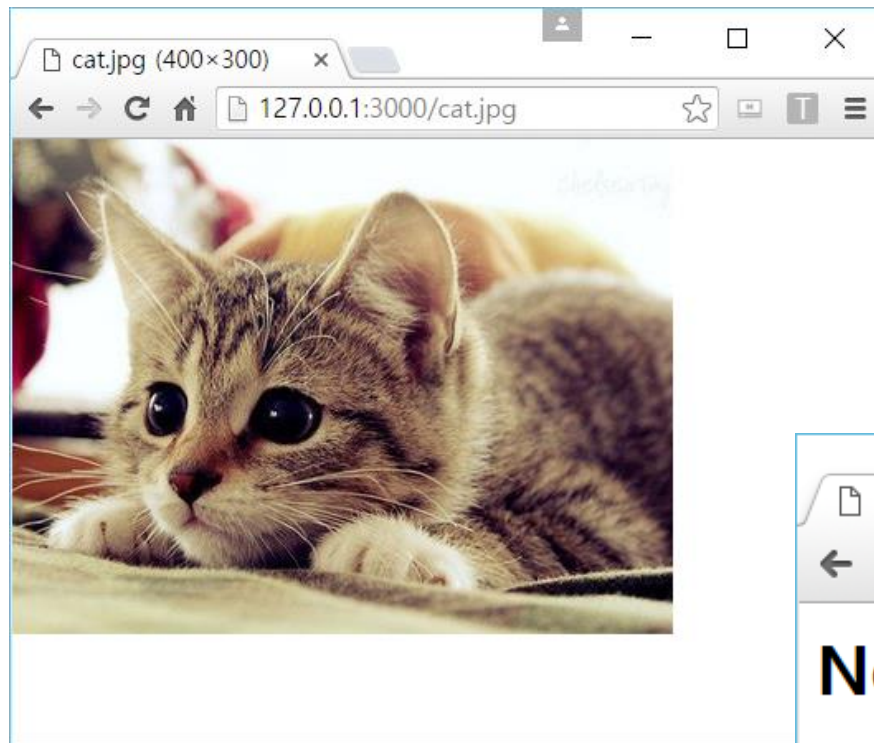
- 정적 파일(html, css, js, image, mp3 등)에 대해 요청하는 HTTP 서버를 만들어보자

예) <http://127.0.0.1:3000/images/cat.jpg>

- fs 모듈을 이용하여 정적 파일을 찾아, 응답 메시지에 파일 내용을 쓴다.
- 스트림의 pipe 기능을 이용하면 간단하게 작성할 수 있다.
  - ✓ `fs.createReadStream(path).pipe(res);`

## 정적 파일 요청

- <http://127.0.0.1:3000/cat.jpg> 요청 시 images/cat.jpg 파일을 읽어 응답데이터로 보내준다.



# POST 요청 처리

- POST 방식으로 요청된 데이터를 처리하기 위해 req의 data와 end 이벤트를 사용한다.
- data 이벤트 : 요청 메시지의 바디를 모두 읽어들이 때까지 계속 동작한다. 전달되는 데이터 조각들을 모두 모으는 일종의 버퍼링 동작을 수행한다.
- end 이벤트 : 요청 메시지의 바디를 모두 읽어왔을 때 발생한다. 즉 요청 메시지를 읽어오는 것이 끝났기 때문에 메시지 분석이 가능한 상태가 된다.

# 폼 (form) 구성

- HTML 화면에서 폼(form) 입력 : GET / POST 요청
- 폼(form) 구성 요소 : <input type='타입' name='이름' .. >

```
<form method='post' action='/upload'>
<ul>
  <li>date : <input type="date" name='date'></li>
  <li>email : <input type="email" name='email'></li>
  <li>file : <input type="file" name='file'></li>
  <li>radio : <input type="radio" name='radio' value='YES'>YES
  <input type="radio" name='radio' value='NO'>NO</li>
  <li>checkbox : <input type="checkbox" name='checkbox'></li>
  <li>text : <input type="text" name='text'></li>
  <li>password : <input type="password" name='password'></li>
  <li>number : <input type="number" name='number'></li>
  <li><textarea name='textarea'>TextArea</textarea></li>
  <li><input type="submit" value="Upload">
</ul>
</form>
```

• date : 2015. 07. 09.

• email : abc@gmail.com

• file : 파일 선택 Plank.jpg

• radio : ☒ YES ☐ NO

• checkbox : ☒

• text : textInput

• password : .....

• number : -4

• TextArea

Upload

# 폼 (form) 데이터 전송

- urlencoded 방식 : 이름=값 방식으로 전송
  - ✓ Content-Type : application/x-www-form-urlencoded
  - ✓ input 태그의 name과 value가 key1=value1&key2=value2로 전송됨
  - ✓ form의 enctype 속성을 작성하지 않으면 urlencoded 방식으로 전송
- 멀티 파트 방식 : 파일, 글자 등 여러 데이터 전송
  - ✓ Content-Type : multipart/form-data; boundary=asdfqwer

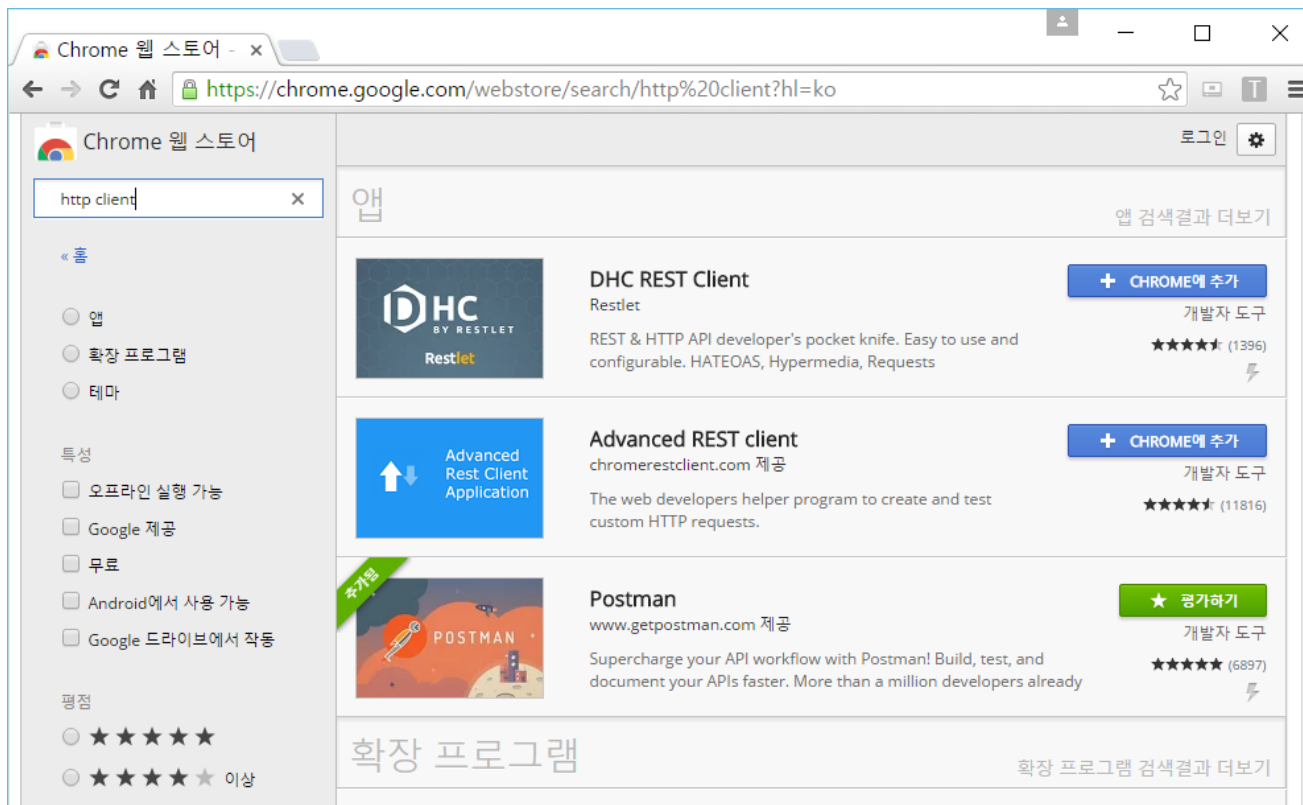
```
<form method='post' action='/upload' enctype='multipart/form-data'>  
...  
</form>
```

## 재전송을 막자

- 중복 Post 요청 방지를 위해 처리 후 redirect를 한다.
- PRG (Post-Redirect-Get) 패턴 방식
- 상태코드 : 302, Location : redirect 주소

# POST 요청 처리

- POST 요청 처리를 위해 HTML 페이지를 만드는 것은 시간 낭비!!
- HTTP Client 프로그램인 postman을 사용하자
- 도구 더보기 > 확장 프로그램 > 더 많은 확장 프로그램 다운로드





# POST 요청 처리

- POST 방식으로 urlencoded 전송

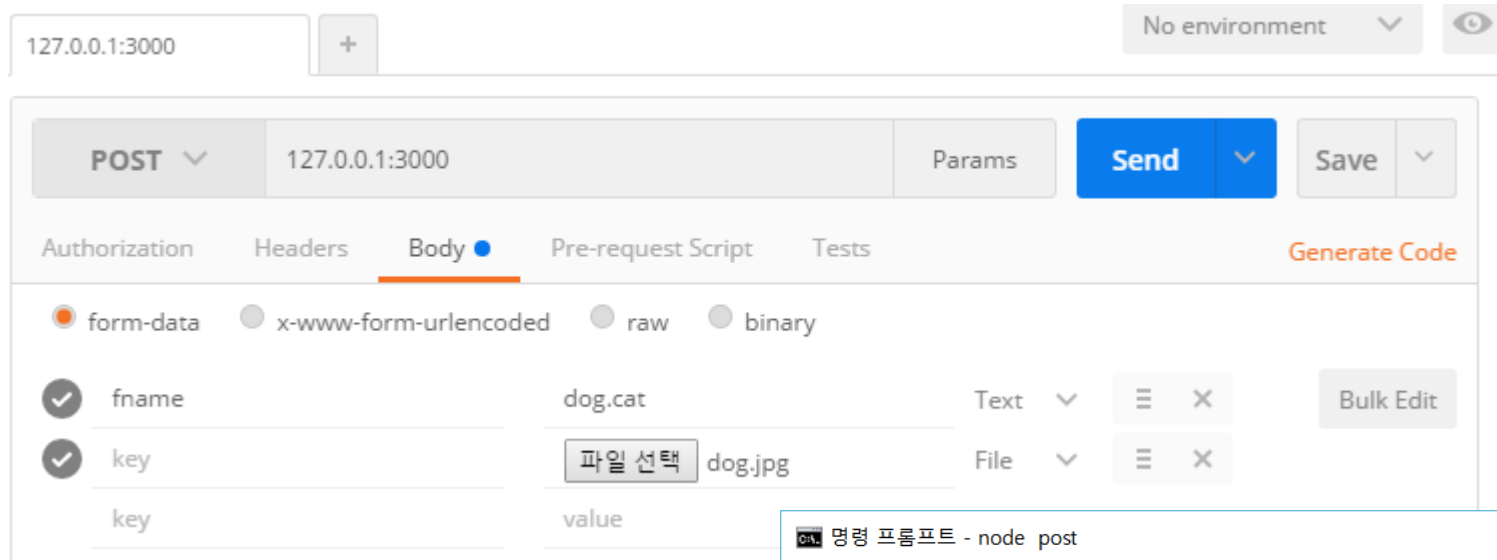
The screenshot shows a REST client interface with the following details:

- URL:** 127.0.0.1:3000
- Method:** POST
- Environment:** No environment
- Params:** (empty)
- Body:** Selected as **x-www-form-urlencoded**. The body is a JSON object: `{ "group": "빅뱅", "member": ["GD", "탑", "대성"] }`.
- Buttons:** Send, Save, Generate Code.

```
명령 프롬프트 - node post
c:\node_workspace\ch09>node post
data 이벤트 핸들러
end 이벤트 핸들러 : group=%EB%B9%85%EB%B1%85&member=GD&member
=%ED%83%91&member=%EB%8C%80%EC%84%B1
{ group: '빅뱅', member: [ 'GD', '탑', '대성' ] }
```

## POST 요청 처리

- POST 방식으로 form-data 전송



```
c:\workspace\ch09>node post
```

data 이벤트 핸들러  
end 이벤트 핸들러 : -----WebKitFormBoundaryKBXiemXDcgLCjWMX  
Content-Disposition: form-data; name="fname"

dog.cat  
-----WebKitFormBoundaryKBXiemXDcgLCjWMX  
Content-Disposition: form-data; name=""; filename="dog.jpg"  
Content-Type: image/jpeg

-----WebKitFormBoundaryKBXiemXDcgLCjWMX  
Content-Disposition: form-data; name=""; filename="dog.jpg"  
Content-Type: image/jpeg

## [실습] 영화 관리 어플리케이션 작성

- GET 방식으로 접근하면 전체 영화 목록을 출력한다.
- 영화 정보 입력 폼을 통해 POST 방식으로 새로운 영화 정보를 등록한다.
- POST 요청 후 refresh하면 중복 Post 요청이 되므로 PRG 패턴을 추가한다.
- 파일명 : 14\_httpWmovie.js

