

M02-01 · GIBALJA VISUAL MANUAL

웹 서비스는 어떻게 움직이는가

한 문장 목표: 사용자가 버튼을 누른 뒤 화면에 결과가 나타날 때까지, 어떤 구성요소가 무엇을 주고받는지 설명합니다.

난이도	개념	실습	셀프 테스트	최종 산출물
Level 1	45분	45분	15분	요청·응답 흐름도, 웹 서비스 분석서

이 매뉴얼은 처음부터 끝까지 외우는 교재가 아닙니다. 그림을 먼저 보고, 자기 말로 설명하고, 실제 브라우저에서 같은 흐름을 찾아보는 학습지입니다.

한 번의 클릭은 여섯 역할을 지난다

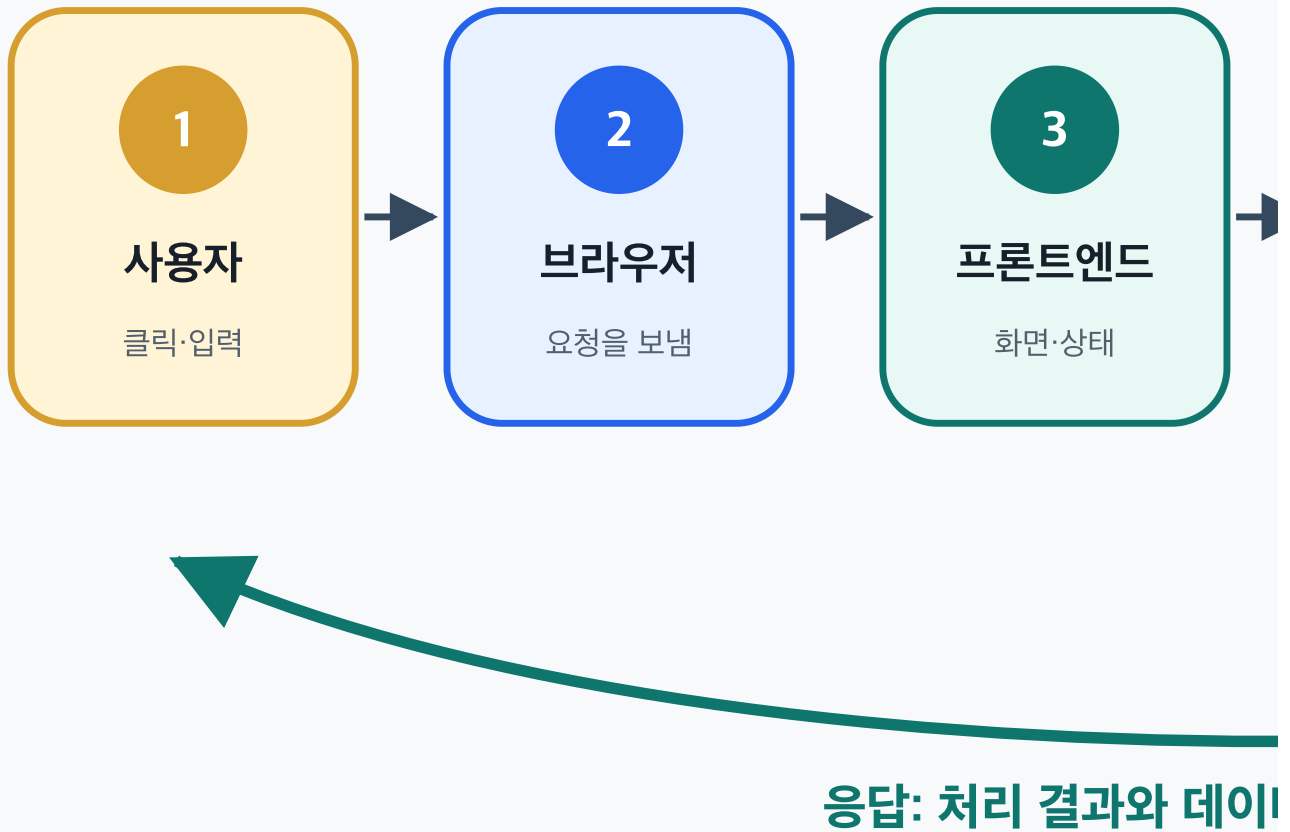
요청은 오른쪽으로, 결과는 다시 왼쪽으로 돌아온다



응답: 처리 결과와 데이터가 화면으로 돌아온다

핵심: 화면 하나가 아니라 여러 역할의 협업이 웹 서비스를 만든다

그림 1. 요청은 처리할 곳으로 이동하고, 응답은 다시 사용자 화면으로 돌아옵니다.





터가 화면으로 돌아온다

1. 이 PDF를 공부하는 방법

1회차: 그림만 보기 · 10분

그림의 화살표를 따라가며 구성요소 이름을 소리 내어 읽습니다. 모르는 용어가 있어도 멈추지 않습니다.

2회차: 본문과 그림 연결 · 35분

각 역할을 한 문장으로 설명합니다. 설명이 막히면 바로 위 그림을 다시 봅니다.

3회차: 실제 요청 찾기 · 45분

브라우저 개발자 도구의 Network 패널에서 요청 주소, 메서드, 상태 코드를 확인합니다.

4회차: 셀프 테스트 · 15분

정답을 가린 채 문제를 풀고, 틀린 항목만 그림으로 돌아가 복습합니다.

학습 완료 기준

그림 없이도 “브라우저 → 프론트엔드 → API → 백엔드 → 데이터베이스 → 응답”을 자기 말로 설명할 수 있고, Network 패널에서 실제 요청 한 건을 찾을 수 있습니다.

2. 먼저 한 장면을 떠올려 봅시다

온라인 도서관에서 **AI 기획**을 검색한다고 가정합니다.

1. 사용자가 검색어를 입력하고 **검색** 버튼을 누릅니다.
2. 화면이 검색 조건을 읽습니다.
3. 브라우저가 서버에 도서 목록을 요청합니다.
4. 서버가 검색 규칙과 권한을 확인합니다.
5. 데이터베이스에서 조건에 맞는 도서를 찾습니다.
6. 서버가 검색 결과를 브라우저로 돌려보냅니다.
7. 화면이 결과를 목록으로 표시합니다.

사용자에게는 짧은 클릭 한 번이지만, 서비스 안에서는 여러 역할이 협업합니다.

BIG IDEA 01

웹 서비스는 화면 한 장이 아니라, 요청을 처리해 응답을 돌려주는 협업 시스템입니다.

3. 식당 주문으로 먼저 이해하기

처음에는 웹 용어보다 식당 주문을 떠올리는 편이 쉽습니다.

웹 서비스는 식당 주문과 닮았다

비유로 감을 잡고, 아래 기술 용어로 정확히 연결한다

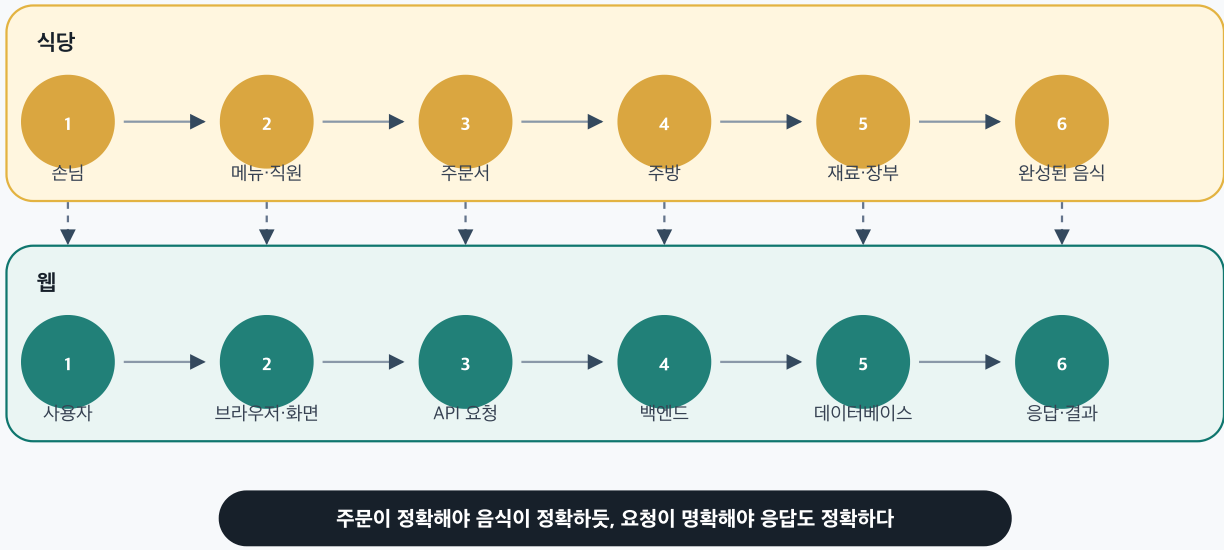


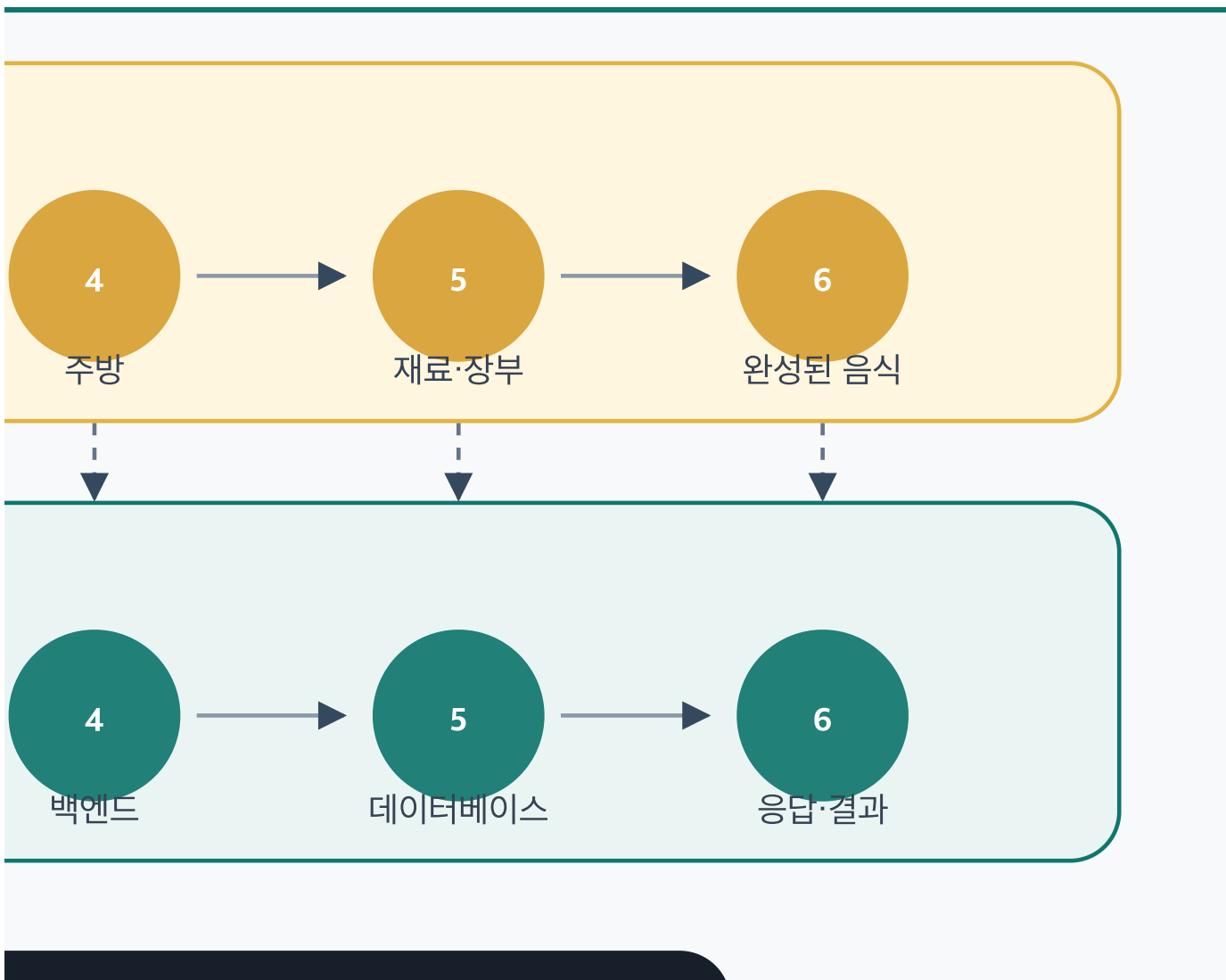
그림 2. 손님의 주문이 주방과 재료 창고를 거쳐 음식으로 돌아오는 과정은 웹의 요청·응답과 닮았습니다.

식당



웹





식당	웹 서비스	핵심 역할
손님	사용자	원하는 결과를 요청함
메뉴판·직원	브라우저·프론트엔드	선택을 받고 보기 쉽게 전달함
주문서	API 요청	무엇을 원하는지 정해진 형식으로 전달함
주방	백엔드	규칙에 따라 요청을 처리함
재료·장부	데이터베이스	필요한 정보를 저장하고 찾아 줌
완성된 음식	응답·화면 결과	처리 결과를 사용자에게 보여 줌

비유는 입문을 돕지만 완전히 같지는 않습니다. 실제 웹 서비스에서는 여러 요청이 동시에 처리되고, 네트워크·보안·서버 운영이 추가됩니다.

30초 확인

식당의 주문서에 가장 가까운 웹 구성요소는 무엇일까요?

정답 보기

API 요청입니다. 무엇을 원하는지 주소·메서드·데이터 등 약속된 형식으로 전달합니다.

4. 여섯 역할을 구분하면 절반은 끝납니다

여섯 역할을 한 문장으로 구분하기

회의에서 역할이 섞이면 요구사항과 책임도 함께 섞인다



API는 데이터베이스가 아니라, 구성요소 사이의 “약속된 통로”다

그림 3. 역할을 정확히 구분하면 요구사항을 어디에 반영하고 누구에게 질문할지 보이기 시작합니다.



브라우저

화면 표시

클릭·입력 수집



프론트엔드

화면 규칙

API 호출·상태 관리



백엔드

업무 규칙

검증·계산·권한



데이터베이스

기억 저장소

조회·저장·관계

≡

3

API

통신 약속

요청·응답 형식

이스

6

서버·클라우드

실행 장소

연결·확장·운영

4.1. 브라우저: 사용자의 행동을 받아 화면을 보여 주는 프로그램

Chrome, Safari, Edge 같은 브라우저는 웹 서비스의 **클라이언트(Client)** 역할을 합니다.

- 주소를 입력받음
- HTML·CSS·JavaScript 파일을 내려받음
- 화면을 그림
- 클릭과 입력을 받음
- 서버에 요청을 보냄
- 응답을 화면에 반영함

4.2. 프론트엔드: 사용자에게 보이는 화면과 상호작용 규칙

프론트엔드(Frontend)는 단순한 디자인 그림이 아닙니다. 버튼을 눌렀을 때 무엇을 할지, 기다리는 동안 무엇을 보여 줄지, 오류가 났을 때 어떻게 안내할지 정합니다.

4.3. API: 서로 대화하는 약속된 통로

응용 프로그래밍 인터페이스(Application Programming Interface, API)는 프론트엔드와 백엔드가 **어떤 요청을 보내고 어떤 응답을 받을지 정한 약속**입니다.

API는 데이터베이스가 아닙니다. API가 백엔드에 요청을 전달하고, 백엔드가 필요할 때 데이터베이스를 사용합니다.

4.4. 백엔드: 업무 규칙을 실행하는 처리 영역

백엔드(Backend)는 입력을 검증하고, 권한을 확인하고, 필요한 계산을 수행하고, 데이터베이스를 조회하거나 변경합니다.

4.5. 데이터베이스: 서비스의 기억 저장소

데이터베이스(Database)는 회원, 게시글, 주문, 권한, 처리 상태 같은 데이터를 저장하고 다시 찾습니다. 일반적인 웹 서비스에서 브라우저가 데이터베이스에 직접 접속하지는 않습니다.

4.6. 서버·클라우드: 프로그램이 실행되는 환경

서버(Server)는 요청을 받고 응답을 보내는 프로그램 또는 그 프로그램이 실행되는 컴퓨터를 가리킬 수 있습니다. 클라우드는 필요한 서버·저장소·네트워크를 빌려 쓰는 운영 방식입니다.

기발자 체크포인트

회의에서 “서버가 문제예요”라는 말을 들으면, 서버 컴퓨터·백엔드 프로그램·네트워크·외부 서비스 중 어디를 말하는지 다시 확인합니다.

5. 요청과 응답은 무엇으로 이루어질까요

하이퍼텍스트 전송 규약(Hypertext Transfer Protocol, HTTP)은 웹에서 요청과 응답을 주고받는 기본 규칙입니다.

요청과 응답은 두 개의 메시지다

무엇을 원하는지 보내고, 처리 결과를 돌려받는다

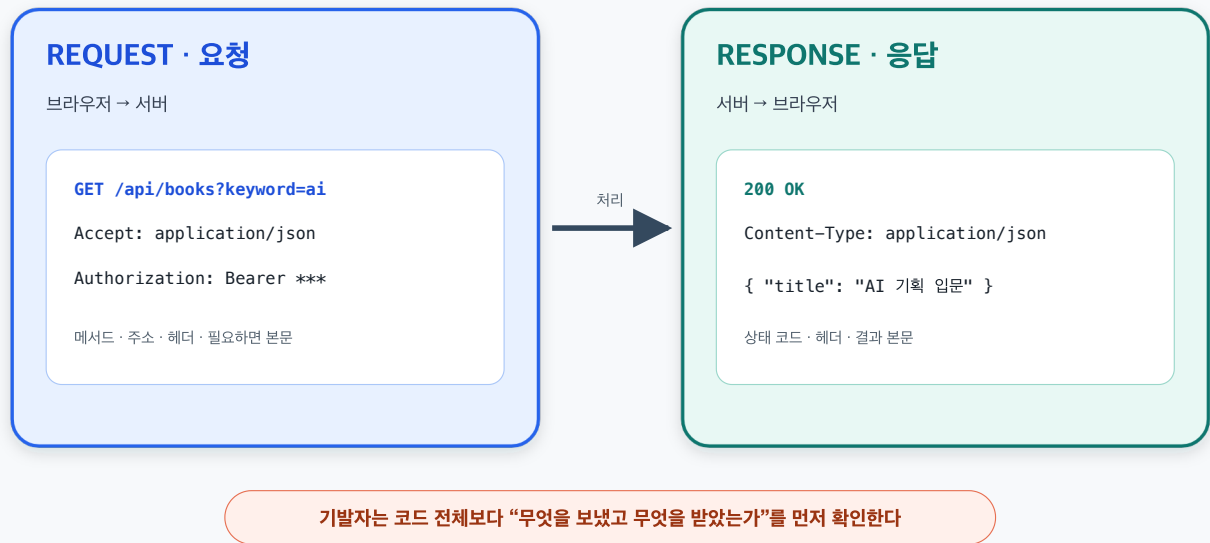


그림 4. 요청과 응답은 방향이 다른 두 개의 메시지입니다.

REQUEST · 요청

브라우저 → 서버

GET /api/books?keyword=ai

Accept: application/json

Authorization: Bearer ***

메서드 · 주소 · 헤더 · 필요하면 본문

처

RESPONSE · 응답

서버 → 브라우저

200 OK

Content-Type: application/json

```
{ "title": "AI 기획 입문" }
```

상태 코드 · 헤더 · 결과 본문

리



요청에서 먼저 볼 네 가지

항목	쉬운 질문	예시
메서드(Method)	무엇을 하려는가	GET , POST
주소(URL)	어디에 요청하는가	/api/books
헤더(Header)	요청의 조건과 부가정보는 무엇인가	인증 정보, 데이터 형식
본문(Body)	서버에 보낼 실제 데이터가 있는가	검색 조건, 작성 내용

응답에서 먼저 볼 세 가지

항목	쉬운 질문	예시
상태 코드(Status Code)	요청이 어떻게 처리됐는가	200 , 404 , 500
헤더	응답의 형식과 조건은 무엇인가	Content-Type
본문	실제 결과는 무엇인가	HTML, JSON, 이미지

이 단계에서는 모든 메서드와 상태 코드를 외울 필요가 없습니다. **요청 주소·메서드·상태 코드·응답 내용** 네 가지부터 찾으면 됩니다.

30초 확인

404 는 데이터베이스가 고장났다는 뜻일까요?

정답 보기
아닙니다. 일반적으로 요청한 자원을 찾지 못했다는 응답 상태입니다. 잘못된 주소, 없는 API 경로, 삭제된 자원 등 여러 원인이 있을 수 있습니다.

6. 한 번의 클릭을 8단계로 따라가기

버튼을 누른 뒤 일어나는 8단계

실제 서비스는 이 순서가 겹치거나 반복되지만, 처음에는 이 지도로 충분하다



반드시 기억할 것: 데이터베이스는 보통 브라우저와 직접 통신하지 않는다

그림 5. 실제로는 일부 단계가 동시에 일어나거나 반복되지만, 입문 단계에서는 이 순서로 이해하면 충분합니다.

1 주소·클릭
사용자 행동이 시작점



2 DNS 조회
도메인을 서버 주소로 찾을



5 API 요청
필요한 데이터 요청



6 업무 처리
백엔드가 규칙 확인

3 안전한 연결

HTTPS 통신 준비

4 화면 받기

HTML·CSS·JS 로드

7 데이터 처리

DB 조회·저장

8 화면 갱신

응답을 사용자가 봄

1단계. 사용자가 주소를 입력하거나 버튼을 누름

모든 요청은 사용자 행동이나 화면 내부의 자동 동작에서 시작합니다.

2단계. 도메인 이름을 서버 주소로 찾음

도메인 네임 시스템(Domain Name System, DNS)이 사람이 읽는 도메인을 통신할 수 있는 인터넷 프로토콜 주소(IP 주소)와 연결합니다.

3단계. 안전한 연결을 준비함

HTTPS를 사용하는 서비스는 상대 서버를 확인하고 통신 내용을 보호할 준비를 합니다.

4단계. 브라우저가 화면 파일을 받음

브라우저는 HTML·CSS·JavaScript·이미지 등 화면에 필요한 자원을 받습니다.

5단계. 프론트엔드가 API를 호출함

검색·로그인·저장처럼 서버 처리가 필요한 기능은 정해진 API로 요청합니다.

6단계. 백엔드가 규칙과 권한을 확인함

입력값이 올바른지, 사용자가 이 작업을 할 수 있는지, 어떤 처리가 필요한지 확인합니다.

7단계. 데이터베이스를 조회하거나 변경함

백엔드는 필요한 데이터를 읽고 쓰며 처리 결과를 만듭니다.

8단계. 응답을 받아 화면을 갱신함

프론트엔드는 성공·빈 결과·오류에 맞는 화면 상태를 사용자에게 보여 줍니다.

BIG IDEA 02

브라우저가 데이터베이스로 바로 가지 않습니다. 백엔드가 권한과 업무 규칙을 확인한 뒤 데이터에 접근합니다.

7. 실제 서비스 한 장 분석

사례: 온라인 도서관에서 도서 검색

단계	실제로 일어나는 일	개발자가 확인할 것
사용자 행동	AI 기획 입력 후 검색 버튼 선택	빈 검색어도 허용하는가
프론트엔드	검색어를 읽고 로딩 상태 표시	중복 클릭을 막는가
API 요청	GET /api/books?keyword=AI%20기획	검색 조건과 페이지 번호가 있는가
백엔드	검색어 길이·권한·검색 규칙 확인	금지어·정렬 기준은 무엇인가
데이터베이스	제목·저자·키워드에서 조건 검색	대소문자·띄어쓰기를 어떻게 처리하는가
API 응답	도서 목록·전체 개수·페이지 정보 반환	결과가 0개일 때 응답은 무엇인가
프론트엔드	목록 또는 빈 결과 안내 표시	오류·빈 결과·로딩 상태를 구분하는가

개발자에게 물어볼 질문

- 어떤 사용자 행동이 API 요청을 만드나요?
- 요청 주소와 메서드는 무엇인가요?
- 입력값 검증은 프론트엔드와 백엔드에서 각각 무엇을 하나요?
- 권한이 없거나 데이터가 없을 때 어떤 상태 코드와 메시지를 주나요?
- 어떤 데이터를 저장하고 얼마나 보관하나요?
- 실패했을 때 사용자 화면과 운영 로그에 무엇이 남나요?

8. 자주 생기는 오해

오해 1. 프론트엔드는 예쁜 화면만 만든다

프론트엔드는 입력, 상태, API 호출, 로딩, 빈 결과, 오류, 접근성까지 다룹니다.

오해 2. API와 데이터베이스는 같은 것이다

API는 통신 약속이고, 데이터베이스는 데이터를 저장·조회하는 시스템입니다.

오해 3. 200 이면 서비스 기능도 항상 성공했다

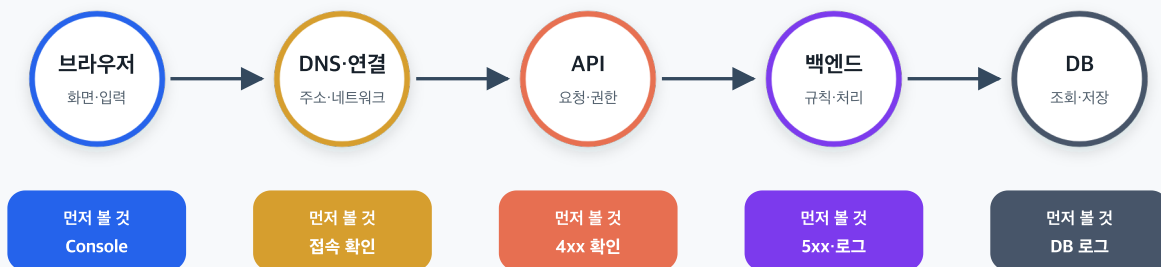
200 은 HTTP 요청이 성공적으로 처리됐다는 신호입니다. 실제 업무 결과는 응답 본문과 서비스 규칙도 함께 확인해야 합니다.

오해 4. 화면이 느리면 프론트엔드 문제다

DNS, 네트워크, 큰 이미지, 느린 API, 백엔드 계산, 데이터베이스 조회 등 여러 위치에서 지연될 수 있습니다.

오류는 메시지가 아니라 위치부터 찾는다

같은 “안 돼요”라도 멈춘 지점에 따라 질문이 달라진다



좋은 오류 질문의 네 요소

① 내가 한 행동 ② 기대한 결과 ③ 실제 결과 ④ Network·Console·로그에서 확인한 사실

그림 6. 오류 메시지보다 멈춘 위치를 먼저 찾으면 질문이 구체적으로 바뀝니다.



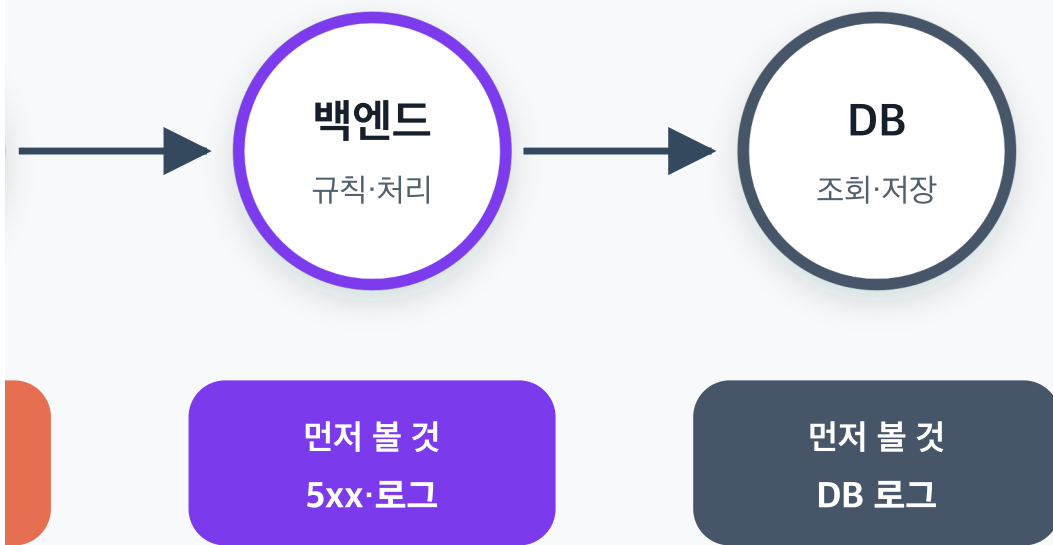
먼저 볼 것
Console

먼저 볼 것
접속 확인

먼저 볼 것
4xx 확인

좋은 오류 질문의 네 요소

① 내가 한 행동 ② 기대한 결과 ③ 실제 결과 ④ Network·Console·로그



||서 확인한 사실

나쁜 질문 “로그인이 안 돼요.”

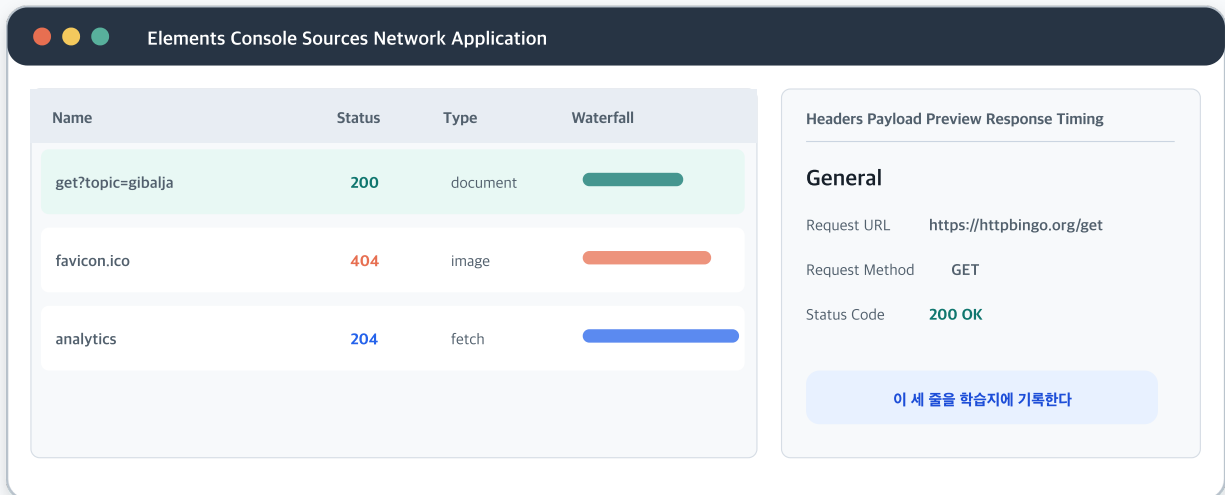
좋은 질문 “로그인 버튼을 누르면 `POST /api/login`이 `401`을 반환합니다. 올바른 시험 계정을 사용했고, 응답 메시지는 `invalid credentials`입니다.”

9. 실습: 브라우저에서 실제 요청 찾기

실습에서는 개인 서비스가 아닌 공개 시험 주소를 사용합니다. 자세한 절차는 L02-01 실습 문서에 있습니다.

Network 패널에서는 네 칸부터 본다

Name · Status · Type · Waterfall을 확인한 뒤 상세 요청을 연다



주의: 개인 서비스의 Network 화면에는 쿠키와 토큰이 보일 수 있으므로 공유하지 않는다

그림 7. 처음에는 요청 주소, 메서드, 상태 코드 세 줄만 찾아도 충분합니다.



Elements Console Sources Network Application

Name	Status	Type	Waterf
get?topic=gibalja	200	document	
favicon.ico	404	image	
analytics	204	fetch	

fall

Headers Payload Preview Response Timing

General

Request URL **https://httpbingo.org/get**

Request Method **GET**

Status Code **200 OK**

[이 세 줄을 학습지에 기록한다](#)

실습 목표

- Network 패널에서 요청 한 건 찾기
- 요청 주소, 메서드, 상태 코드 기록하기
- 응답 본문에서 보낸 검색값 확인하기

시험 주소

<https://httpbingo.org/get?topic=gibalja>

핵심 절차

1. Chrome에서 시험 주소를 엽니다.
2. 개발자 도구를 엽니다.
 - Windows·Linux: **F12** 또는 **Ctrl + Shift + I**
 - macOS: **Option + Command + I**
3. **Network**를 선택하고 페이지를 새로 고칩니다.
4. 이름에 **get?topic=gibalja** 가 있는 요청을 선택합니다.
5. **Headers**에서 Request URL, Request Method, Status Code를 기록합니다.
6. **Response**에서 **topic** 값이 **gibalja** 인지 확인합니다.

보안 주의

개인 메일·금융·회사 내부 시스템의 Network 화면에는 쿠키, 토큰, 개인정보가 포함될 수 있습니다. 화면 전체를 캡처하거나 다른 사람에게 공유하지 않습니다.

10. 인쇄용 학습지

A. 요청·응답 관찰 기록

관찰 항목	기록
내가 한 행동	
Request URL	

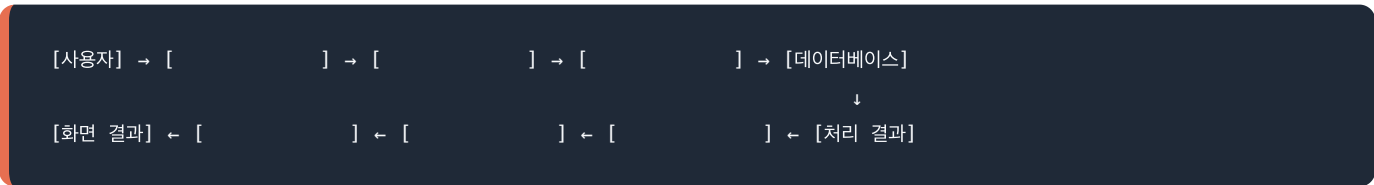
관찰 항목	기록
Request Method	
Status Code	
Response에서 찾은 값	
요청이 지나간 구성요소	
궁금한 점	

B. 내가 아는 서비스 분석

평소 사용하는 서비스 하나를 골라 한 기능만 분석합니다.

질문	나의 분석
사용자와 행동은 무엇인가	
화면은 어떤 입력을 받는가	
어떤 API 요청이 필요해 보이는가	
백엔드는 어떤 규칙을 확인해야 하는가	
데이터베이스에는 무엇을 저장할 것 같은가	
성공·빈 결과·오류 화면은 어떻게 다른가	

C. 화살표로 직접 그리기



11. 셀프 테스트

정답을 보기 전에 그림 없이 먼저 풀어 봅니다.

문제 1 · 한 문장 설명

웹 서비스를 “요청”과 “응답”이라는 두 단어를 사용해 설명하세요.

문제 2 · 순서 배열

다음 항목을 일반적인 처리 순서로 배열하세요.

화면 갱신 / API 요청 / 사용자 클릭 / 데이터베이스 조회 / 백엔드 처리

문제 3 · 역할 연결

다음 역할을 알맞은 구성요소와 연결하세요.

1. 데이터를 저장하고 찾을
2. 사용자 입력과 화면 상태를 관리함
3. 입력 검증과 업무 규칙을 실행함
4. 요청과 응답의 형식을 약속함

API / 백엔드 / 데이터베이스 / 프론트엔드

문제 4 · 요청 읽기

다음 요청에서 메서드와 주소를 찾으세요.

```
GET /api/books?keyword=ai HTTP/1.1
Accept: application/json
```

문제 5 · 상태 코드 판단

요청한 주소가 존재하지 않아 404 가 반환됐습니다. 무조건 데이터베이스 장애라고 판단해도 될까요? 이유를 쓰세요.

문제 6 · 오류 질문 바꾸기

“검색이 안 돼요”를 개발자가 확인할 수 있는 질문으로 바꾸세요. 행동, 기대 결과, 실제 결과, 확인한 사실을 포함합니다.

문제 7 · 개발자 질문

회원 정보 수정 기능을 기획할 때 개발자에게 확인할 질문을 세 개 쓰세요.

문제 8 · 실습 확인

Network 패널에서 시험 요청의 Request Method와 Status Code를 기록하세요.

12. 정답과 해설

문제 1

예시 답: 웹 서비스는 사용자의 요청을 받아 업무 규칙과 데이터를 처리하고, 그 결과를 응답으로 돌려주는 시스템입니다.

문제 2

사용자 클릭 → API 요청 → 백엔드 처리 → 데이터베이스 조회 → 화면 갱신

실제로는 화면이 요청 전에 입력을 읽고, 백엔드와 데이터베이스 사이의 통신이 여러 번 반복될 수 있습니다.

문제 3

1. 데이터베이스
2. 프론트엔드
3. 백엔드
4. API

문제 4

- 메서드: GET
- 주소: /api/books?keyword=ai

문제 5

아닙니다. 404 는 요청한 자원을 찾지 못했다는 응답입니다. 잘못된 주소, 없는 API 경로, 삭제된 자원 등 여러 원인이 있으므로 요청 주소와 서버의 경로 설정부터 확인합니다.

문제 6

예시 답: “AI 기획 을 입력하고 검색 버튼을 눌렀지만 결과 목록이 나타나지 않았습니다. 빈 결과 안내가 나올 것으로 기대했습니다. Network에서 GET /api/books?keyword=AI%20기획 요청이 500 을 반환한 것을 확인했습니다.”

문제 7

예시 답:

- 본인 정보만 수정할 수 있도록 권한을 어디서 확인하나요?

- 수정 가능한 항목과 입력 검증 규칙은 무엇인가요?
- 변경 전후 값과 작업자를 감사 로그에 남기나요?

문제 8

정상적인 시험 환경의 예시:

- Request Method: **GET**
- Status Code: **200**

네트워크 또는 시험 사이트 상태에 따라 다른 결과가 나올 수 있습니다. 이 경우 실제 값을 그대로 기록하고 오류 위치 지도를 사용해 원인을 추정합니다.

13. 한 장 요약

M02-01 한 장 요약

웹 서비스 = 요청을 받아 규칙과 데이터를 처리하고 응답하는 협업 시스템



기억할 4문장

1. 브라우저는 클라이언트 역할을 한다.
2. API는 요청·응답의 약속이다.
3. 백엔드는 규칙, DB는 기억을 맡는다.
4. 오류는 멈춘 위치부터 찾는다.

회의에서 물을 4가지

1. 어떤 사용자 행동이 요청을 만드는가?
2. 어떤 API가 어떤 데이터를 주는가?
3. 권한과 입력 검증은 어디서 하는가?
4. 실패하면 사용자와 로그에 무엇이 남는가?

설명할 수 있고, Network에서 찾을 수 있으면 학습 완료

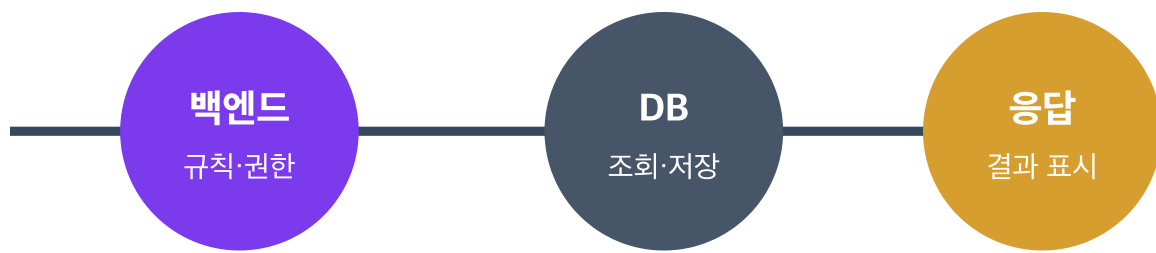
그림 8. 이 페이지를 인쇄하거나 화면에 띄워 두고 전체 흐름을 반복 설명해 보세요.



기억할 4문장

1. 브라우저는 클라이언트 역할을 한다.
2. API는 요청·응답의 약속이다.
3. 백엔드는 규칙, DB는 기억을 맡는다.
4. 오류는 멈춘 위치부터 찾는다.

선명한 스 인 그 Network에



회의에서 물을 4가지

1. 어떤 사용자 행동이 요청을 만드는가?
2. 어떤 API가 어떤 데이터를 주는가?
3. 권한과 입력 검증은 어디서 하는가?
4. 실패하면 사용자와 로그에 무엇이 남는가?

서 찬은 스 인 O며 하스 와르

14. 다음 매뉴얼

M02-02 「URL·도메인·DNS·포트 읽기」에서 요청이 어느 서버로 찾아가는지 더 자세히 배웁니다.

15. 출처와 확인일

- MDN, How the web works, 2026. 7. 15. 확인
- MDN, Overview of HTTP, 2026. 7. 15. 확인
- RFC Editor, RFC 9110: HTTP Semantics, 2026. 7. 15. 확인
- Chrome for Developers, Network panel, 2026. 7. 15. 확인

16. 변경 이력

버전	날짜	변경 내용	상태
0.1.0	2026. 7. 15.	파일럿 초안·시각 자료·실습·셀프 테스트 통합	파일럿