웹개발: client가 웹브라우저로 서비스를 요청하고 요청된 서비스를 서버(웹서버, WAS)가 확인한다.

웹브라우저: 컴파일을 한다. 무엇을? -> html, css, js 를...

웹 서버: 정적 데이터 처리 -> 그림, html, css, js소스 / e.g) 아파치

WAS: 동적 데이터 처리 -> 데이터베이스 정보 (이것을 처리하는 프레임워크=스프링) / e.g) 톰캣

jsp는 데이터베이스 정보를 html로 렌더링을 해준다. 데이터베이스의 정보를 dto, dao를 통해 전달하는데 이는 객체이다. 즉, 자바 언어란건데 html은 자바 언어가 아니라 dto, dao 를 인식을 못한다. 객체를 랜더링해서 html이 인식하게 만들어 주는 게 view template(jsp, thymeleaf)이다.

view template란? 클라이언트에게 응답할 화면

웹브라우저는 http라는 프로토콜로 통신하게 되어있다.

(통신규약): 통신하는 규칙정의

통신: 송신자와 수신자가 필요. client / server

통신방식: http는 tcp방식을 쓴다

TCP: 연결을 먼저 설정 -> 데이터 전송 -> 연결 종료

TCP는 서비스 요청 시 클라이언트가 응답을 계속 기다림.

UDP: 연결 안하고 그냥 보냄

UDP가 TCP보다는 딜레이가 더 많이 발생할 수 있다

TCP와 달리 응답을 기다리지는 않는다

주소:

IP주소 – 컴퓨터 고유의 주소

port번호 – 프로그램 주소 / http의 포트번호 = 80

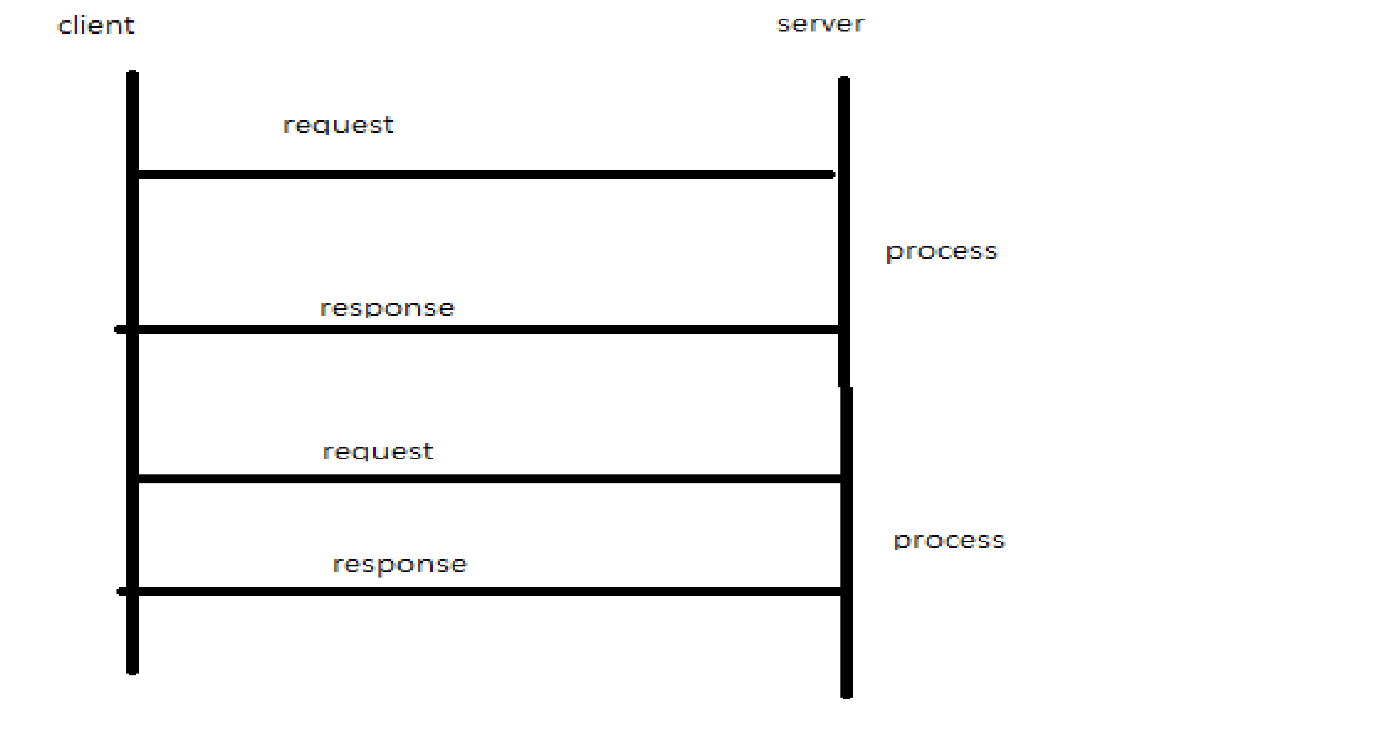
카카오톡의 포트넘버 등등… 해당 포트번호를 막으면

컴퓨터에서 해당 프로그램을 쓸 수 없다

// 웹브라우저는 http라는 프로토콜을 쓰고, tcp라는 통신방식을 쓴다. http의 포트넘버는 80번이다.

웹개발 방식:

동기 방식 – 요청(request)을 하고 응답(response)을 기다림



비동기 방식 – 요청 -> 응답 기다리지 않음 -> 이후 응답이 오면 처리

비동기는 화면을 refresh 하지 않고 화면 일부분만 업데이트 가능하다. 동기는 무조건 모든 정보를 다시 불러와야함(새로고침)

e.g) API

메소드를 콜 할 때는 매개변수를 넘김. 웹에서 서버는 클라이언트가 콜할 때 parameter를 넘김

TCP방식에서 클라이언트가 요청하고 답변이 오면 이후 연결은 끊긴다. 즉, 위에서는 2번의 TCP 연결 및 해제 과정을 거친 것

e.g) 만약 로그인 한 뒤 로그인 상태로 메일함에 들어가거나 하려면 연결이 끊기지 않아야한다. 하지만 TCP방식에선 무조건 연결이 끊겨야 한다. 그럼 해당 로그인 방식으로 로그인을 했다는 증거를 남겨서 메일함 서비스를 이용할 수 있게 해주는 정보를 쿠키(cookie)라고 부른다.

쿠키는 클라이언트 측에 저장되는 데이터

세션은 서버 측에서 저장되는 데이터

쿠키는 내 컴퓨터에 저장되는 것이라 볼 수 있기에 정보가 털리기 쉬움

**쿠키 (Cookie)**

* **정의**: 클라이언트(브라우저)에 저장되는 작은 데이터 조각이야.
* **용도**: 사용자 설정, 로그인 정보, 쇼핑카트 내용 등을 저장하는 데 사용돼.
* **특징**: 쿠키는 브라우저에 저장되고, 웹사이트에 다시 요청할 때마다 서버로 전송돼. 만료 기간을 설정할 수 있어서 지속적으로 사용할 수도 있어.

**세션 (Session)**

* **정의**: 서버 측에서 사용자 정보를 저장하는 방식이야.
* **용도**: 로그인 상태 유지, 장바구니 상태 관리 등 동적인 정보를 저장하는 데 주로 사용돼.
* **특징**: 세션 데이터는 서버에 저장되므로 보안이 더 높고, 사용자가 브라우저를 닫거나 일정 시간이 지나면 만료돼.

요약하자면, 쿠키는 클라이언트 측에 저장되는 반면, 세션은 서버 측에 저장된다는 점에서 차이가 있어. 각각의 용도와 특징을 고려해 적절히 사용하면 돼!

쿠키에 저장 될 정보는 편의성

세션에 저장 될 정보는 보안에 취약한 부분

Client Server

웹 브라우저 Oracle

WAS Tomcat 8.5

웹서비스를 처리하고 응답

처리를 누가 하는가? Spring Framework

응답을 누가 하는가? Html(view)

view로써는 thymeleaf나 jsp가 사용됨

// 웹 브라우저가 웹 서버에 url(http://)을 통해 요청(request)을 보낸다. 처리 가능한 정적 서비스는 웹서버가 처리하고 동적 데이터가 존재한다면 동적 서비스 처리를 위해 WAS에 요청을 보낸다.

WAS는 동적 처리를 위해 Spring을 사용하고, 필요한 경우 오라클 서버에 데이터베이스 요청을 보낸다.

오라클 서버가 데이터를 Spring에 반환하고, Spring은 이 데이터를 바탕으로 HTML 등의 응답화면(view template)을 생성한다.

랜더링: Spring은 가져온 데이터를 바탕으로 HTML 문자열\*[html은 모든 것을 문자열로 생각한다]을 생성. 이를 위해 JSP, Thymeleaf 같은 뷰 템플릿 엔진을 사용할 수 있다. 예를 들어, 사용자 정보가 필요하다면, 해당 정보를 포함한 HTML 코드가 만들어진다. (JSP나 Thymeleaf와 같은 뷰 템플릿 엔진은 서버 측에서 데이터를 기반으로 HTML 문자열\*을 생성하는 과정을 "랜더링"이라고 한다. 즉, 랜더링은 서버에서 이루어진다.)

생성된 HTML이 웹 브라우저로 전송된다.

이 응답이 웹 브라우저로 전송되면, 브라우저는 이를 컴파일하여 사용자에게 보여준다.

이는 동기 방식이다.

//

요청 처리는 controller, 응답 처리는 view가 한다.

// 웹 브라우저가 WAS에 요청을 보낸다.

WAS는 Spring을 통해 요청을 처리하고, 필요한 경우 오라클 서버에 데이터베이스 요청을 보낸다.

오라클 서버가 데이터를 Spring에 반환하고, Spring은 이 데이터를 바탕으로 HTML 등의 응답을 생성한다.

// Spring은 가져온 데이터를 바탕으로 HTML 페이지를 생성. 이를 위해 JSP, Thymeleaf 같은 템플릿 엔진을 사용할 수 있다.. 예를 들어, 사용자 정보가 필요하다면, 해당 정보를 포함한 HTML 코드가 만들어진다.

// **응답 전송**: 생성된 HTML이 웹 브라우저로 전송된다.

이후, 이 응답이 웹 브라우저로 전송되면, 브라우저는 이를 랜더링하여 사용자에게 보여준다

마지막으로, 날씨나 시간을 다른 웹에서 가져오려 하면 웹에서 해당 프로그램에 request를 해서 비동기 API를 가져온다. //

pom.xml 파일이 있다면 빌드툴은 메이븐임

dependency 리아브러리의 빌드 태그

OSI 7 layers

l1~l7

1. 전선

2. MAC주소

3. IP

4. TCP, UDP, port number

5. 세션

6.

7. http, socket

응용계층의 http 프로토콜은 l4에 있는 TCP전송방식을 사용하며 ip 주소체계를 사용하여 동작하고 있다.

url: 파라미터가 없는 경우와 있는 경우로 나뉜다.

HTML

0. html문서는 html을 기본으로, <style>영역, <javascript>영역이 있다.

jquery는 자바스크립트로 개발된 일종의 라이브러리로 html문서에서는 자바스크립트 영역에서 코딩하면 된다.

1. html문서에서는 div영역을 잘 나누어야 깔끔하다.

3. 태그는 태그명과 속성으로 되어 있다.

태그의 정의와 태그가 갖는 속성을 숙지해야 한다.

4. 하나의 문서를 Document라고 하고, Document를 구성하는 object를 현재 태그의 부모, 형제, 자식, 자손을 찾아 낼 수 있어야 한다.

5. html문서를 동적으로 제어하기 위해서는 Object를 선택하고 선택한 Object에 디자인 적용, 데이터 값 변경 등 전체 문서를 새로고침하지 않고, 부분 Object의 내용을 변경할 수 있다. > 예를 들어, 아이디중복체크, 좋아요.

6. 5번에서 언급한 Object를 선택할 때는 선택자를 선택한다고 한다. 방법은 태그로 선택하는 방법, 태그에 지정된 아이디로 선택하는 방법, 태그에 지정된 클래스로 선택하는 방법 등이 있다.

1. html문서의 기본 구조

2. 태그와 속성이해

3. 태그의 속성 중 아이디는 문서내에서 유니크 해야 한다.

4. 태그의 속성 중 클래스는 유니크하지 않아도 된다.

5. 스타일과 스크립트 태그의 의미와 위치에 따른 로딩순서

6. 스타일에서의 주석 표시 기호

7. 스크립트에서 주석 표시 기호

8. 스타일에서 선택자란? 선택하는 방법 3가지

9. div로 화면을 분할하는 방법

display 또는 float등으로 div위치를 배치시킬 수 있음

semantic Tag - 공간 분할/공간에 의미 부여

1. header: 제목

2. nav: 메뉴

3. section: 주요내용

4. aside: 서브메뉴

5. footer: 개발자

display

1. block: 한줄 차지 + 줄바꿈. 가로세로 크기 조절이 가능함 e.g) div, h1, p

2. inline: 자동으로 특정 공간만 영역으로 만듦. 가로세로 크기 지정이 불가능함. 줄바꿈 발생x e.g) span, a, strong

3. inline-block: 크기 조절 가능. 원하는 만큼의 영역 지정 가능. 줄바꿈 발생x e.g) img, button

주요 키워드:

html문서 버전

metadata - charset

metadata - viewport

1. html기본 뼈대: header body

2. 화면 분할 시멘틱태그나 div로

3. display모드 3가지 border, inline, inline-border

4. display의 여백 조절이 힘드니 그냥 float left 혹은 flex를 사용

5. 깔끔한 정렬을 위해선 많은 div를 써보자

6. margin과 padding갑을 잘 써서 깔끔히 정렬해보자

7. span과 p로 꼭 묶어서 글을 쓰자

8. 목록 태그 ul, li

9. table 태그

1. row -> <tr>

2. column -> <td>

3. row를 합치는 것 -> rowspan

4. col을 합치는 것 -> colspan

10. form 태그로 테이블을 감싸야만 함

api view template 에서 response 타입이 html일 수도 있지만 int string json일 수도 있다.

이것을 비동기방식이라고 한다.

html로 response하는 것은 동기 방식이라고 한다. e.g) 화면 전체 갱신

유효성 체크는 js와 정규식으로 하는 것

js를 잘만 쓰면 서버에 부담을 줄일 수 있다.

1. html에서 태그, id, 클래스, 자식, 자손 등 object를 선택할 수 있어야 한다.

2. 선택한 object에서 value, innertext, innerhtml로 데이터를 조작할 수 있다.

3. 새로운 object를 생성해서 문서에 포함시킬 수 있다.

4. 비동기 통신. API 사용 능력

5. 객체 활용 -> 그래프 -> 공공 API

web js 기능

1. 메뉴 펼치기

2. 메뉴바 고정

3. 자동 화면 갱신(interval)

크롬의 개발자 모드 사용 가능하여 문제해결하는 단서를 얻게 됨(프로젝트 때 언급하면 좋을 것)

1. element 탐색 (노드 탐색 및 선택)

2. html 문서 소스 탐색 및 구조 분석

3. 콘솔 활용 - js간단한 테스팅

4. network - request와 response 과정 분석. url, method, code(304, 404)