

2021 EZEN JavaScript

Ajax 업비트 open API

Team Project

6조 - 차지환, 김종규, 강예지, 서지영

01

프로젝트 주제 및
구현 기능 소개

6조 - 업비트 Open API

구현 기능

01 코인 종목 조회

02 코인 시세 조회

03 가격순 정렬

04 거래대금 정렬

05 전일 대비 퍼센트 정렬

프로젝트 주제

업비트 Open API 를 통해 현재 시장에서 판매되는 코인명과 현재가, 전일대비, 거래대금 정보 전달

02

활용 API

마켓 코드 조회 API

GET <https://api.upbit.com/v1/market/all>

업비트에서 거래 가능한 마켓 목록 조회 API

Response

필드명	설명	타입
market	업비트에서 제공중인 시장 정보	String
korean_name	거래 대상 암호화폐 한글명	String
english_name	거래 대상 암호화폐 영문명	String
market_warning	유의 종목 여부 NONE (해당 사항 없음), CAUTION(투자유의)	String

QUERY PARAMS

isDetails boolean

유의종목 필드과 같은 상세 정보 노출 여부(선택 파라미터)

false



사용한 응답 데이터

2. market

업비트에서 제공중인 시장 정보

2. korean_name

거래 대상 암호화폐 한글명

3. english_name

거래 대상 암호화폐 영문명

현재가 정보

GET https://api.upbit.com/v1/ticker

요청 당시 종목의 스냅샷을 반환

Response

필드	설명	타입
trade_price	종가	Double
signed_change_price	부호가 있는 변화액	Double
signed_change_rate	부호가 있는 변화율	Double
trade_volume	가장 최근 거래량	Double
acc_trade_price	누적 거래대금(UTC 0시 기준)	Double
acc_trade_price_24h	24시간 누적 거래대금	Double

QUERY PARAMS

markets string **required**

반점으로 구분되는 마켓 코드 (ex. KRW-BTC, BTC-ETH)

사용한 응답 데이터

2. trade_price

종가

2. acc_trade_price_24h

24시간 누적 거래대금

3. signed_change_rate

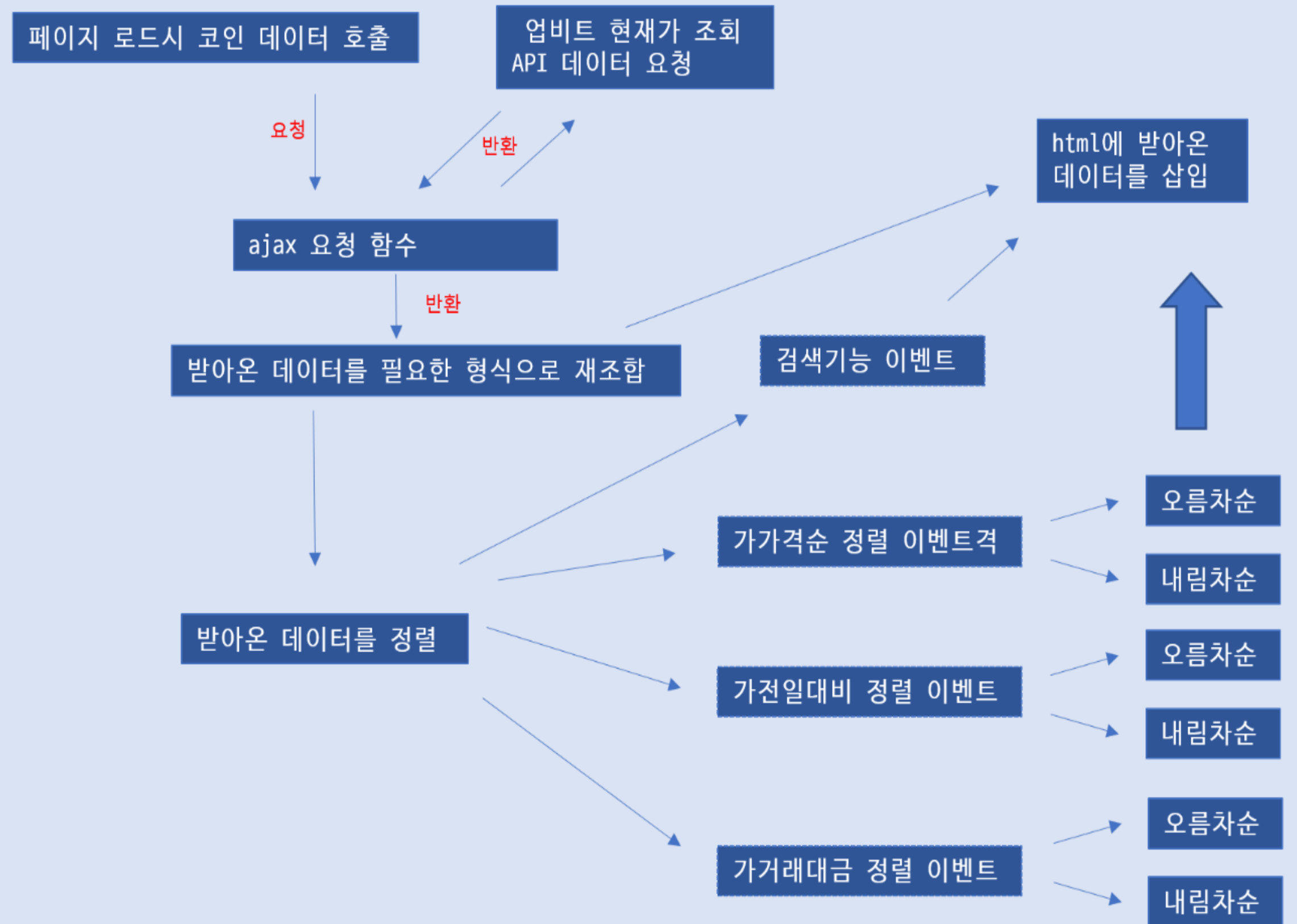
거래 대상 암호화폐 영문명

03

프로젝트 설계

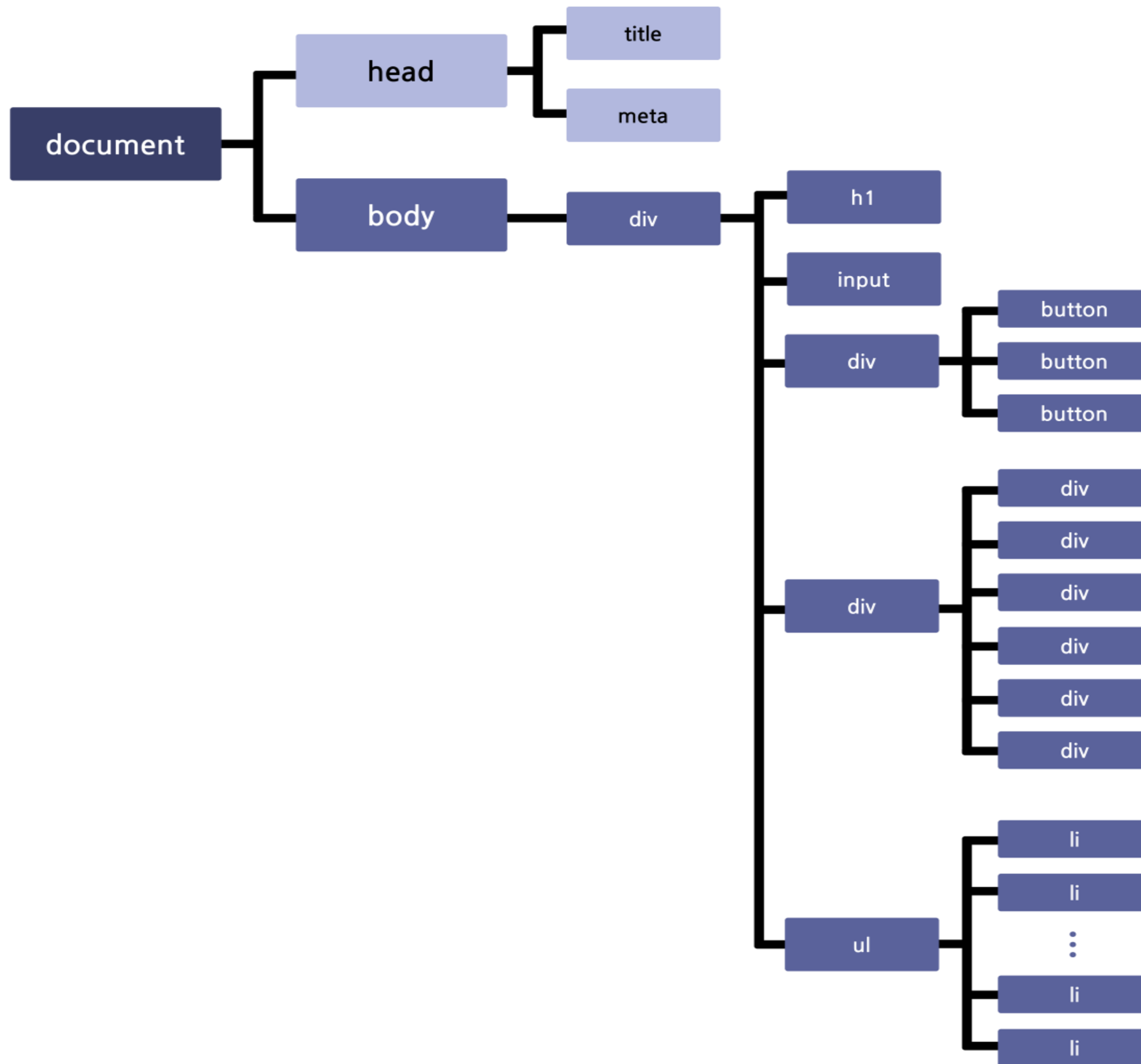
업비트 Open API

프로세스 흐름



업비트 Open API

HTML DOM 구조



04

프로세스별 JavaScript 구현

업비트 Open API

전체 구현 결과

업비트 코인 정보

코인 이름

가격 순으로 정렬

전일대비 정렬

거래대금 정렬

코인명 (한국)	코인명 (영문)	시장정보	현재가	전일대비	거래대금
비트코인	Bitcoin	KRW-BTC	62,950,000	0.52%	671,248백만
이더리움	Ethereum	KRW-ETH	5,375,000	-0.24%	591,588백만
네오	NEO	KRW-NEO	36,440	-0.16%	22,070백만
메탈	Metal	KRW-MTL	2,945	4.25%	50,736백만
라이트코인	Litecoin	KRW-LTC	197,200	-1.67%	20,067백만
리플	Ripple	KRW-XRP	1,030	0.00%	469,691백만
이더리움클래식	Ethereum Classic	KRW-ETC	48,550	-1.32%	121,666백만
오미세고	OmiseGo	KRW-OMG	8,190	2.31%	24,847백만
스테이터스네트워크토큰	Status Network Token	KRW-SNT	99	-1.98%	21,268백만
웨이브	Waves	KRW-WAVES	23,840	-1.61%	33,992백만
넴	NEM	KRW-XEM	169	0.60%	16,269백만
퀀텀	Qtum	KRW-QTUM	12,920	-2.12%	123,178백만
리스크	Lisk	KRW-LSK	3,600	0.84%	4,514백만
스팀	Steem	KRW-STEEM	608	-0.16%	10,866백만
스텔라루멘	Lumen	KRW-XLM	364	-0.27%	79,317백만
아더	Ardor	KRW-ARDR	327	-0.91%	12,497백만
아크	Ark	KRW-ARK	1,690	-0.29%	9,280백만
스토리지	Storj	KRW-STORJ	2,285	0.66%	47,371백만

아이오에스티	IOST	KRW-IOST	42.5	1.43%	6,886백만
리퍼리움	Refereum	KRW-RFR	17.7	-0.56%	17,839백만
시빅	Civic	KRW-CVC	490	3.81%	41,351백만
에브리피디아	Everipedia	KRW-IQ	17.2	0.58%	8,769백만
아이오타	IOTA	KRW-IOTA	1,470	1.03%	9,603백만
메인프레임	Mainframe	KRW-MFT	12.7	1.60%	14,081백만
온톨로지가스	ONG	KRW-ONG	1,100	-1.35%	34,621백만
가스	GAS	KRW-GAS	8,000	-1.78%	15,140백만
센티넬프로토콜	Sentinel Protocol	KRW-UPP	198	0.51%	20,670백만
엘프	aelf	KRW-ELF	620	7.27%	378,877백만
카이버네트워크	Kyber Network	KRW-KNC	1,785	1.71%	11,535백만
비트코인에스브이	Bitcoin SV	KRW-BSV	179,900	-5.42%	95,177백만
세타토큰	Theta Token	KRW-THETA	6,160	0.16%	4,416백만
쿼크체인	QuarkChain	KRW-QKC	29.5	0.00%	10,132백만
비트토허트	BitTorrent	KRW-BTT	4.03	-2.89%	978,993백만
모스코인	Moss Coin	KRW-MOC	319	-0.31%	72,054백만
엔진코인	Enjin	KRW-ENJ	3,505	-2.09%	17,000백만
세타퓨엘	Theta Fuel	KRW-TFUEL	315	-0.94%	21,333백만
디센트럴랜드	Decentraland	KRW-MANA	4,695	0.32%	170,184백만
앵커	Ankr	KRW-ANKR	145	5.84%	40,679백만
아르고	Aergo	KRW-AERGO	352	-1.40%	22,531백만
코스모스	Cosmos	KRW-ATOM	30,340	0.36%	29,344백만
썬더토큰	Thunder Token	KRW-TT	15.4	0.00%	19,851백만
캐리프로토콜	Carry Protocol	KRW-CRE	12.7	-0.78%	23,279백만

프로세스별 JavaScript 구현 코드

페이지 로드시 코인 데이터 호출

```
const coinList = document.querySelector("#coin_list");

// 페이지 로드시 코인 데이터 호출
function loadMarket() {
  getData("market/all?isDetails=false", null, loadTicker);
}

loadMarket();
```



JavaScript

프로세스별 JavaScript 구현 코드

업비트 현재가 조회 API 데이터 요청

```
// 업비트 현재가 조회 API 데이터 요청
function loadTicker(data, names, callback = assignData) {
  const markets = data
    .map((e) => e.market)
    .filter((e) => e.substring(0, 3) === "KRW");
  const coinNames = data.filter((e) => e.market.substring(0, 3) ===
    "KRW");

  getData(`ticker?markets=${markets}`, coinNames, callback);
}
```



JavaScript

프로세스별 JavaScript 구현 코드

ajax 요청 함수

```
// ajax 요청 함수
function getData(url, names, callback) {
  const xhr = new XMLHttpRequest();
  const addr = `https://api.upbit.com/v1/${url}`;

  xhr.onreadystatechange = (e) => {
    const { target } = e;

    if (target.readyState === XMLHttpRequest.DONE) {
      if (target.status === 200) {
        const req = JSON.parse(target.response);

        callback(req, names);
      }
    }
  };
  xhr.open("GET", addr);
  xhr.send();
}
```

JavaScript

프로세스별 JavaScript 구현 코드

받은 데이터를 필요한 형식으로 재조합

```
// 받은 데이터를 필요한 형식으로 재조합
function assignData(data, names, callback) {
  const assigns = names.map((e, i) => {
    return {
      ...e,
      market: data[i].market,
      trade_price: data[i].trade_price,
      acc_trade_price: data[i].acc_trade_price_24h,
      signed_change_rate: data[i].signed_change_rate,
    };
  });
  sort(assigns);
}
```



JavaScript

프로세스별 JavaScript 구현 코드

받은 데이터를 정렬

```
// 받은 데이터를 정렬
function sort(data) {
  const assigns = data;
  const sortPrice = document.querySelector("#sort-price");
  const sortRate = document.querySelector("#sort-rate");
  const sortTrade = document.querySelector("#sort-accTrade");
  let handlePrice = false;
  let handleRate = false;
  let handleTrade = false;
  sortPrice.addEventListener("click", () => {
    if (!handlePrice) {
      assigns.sort((a, b) => b.trade_price - a.trade_price);
      printCoin(assigns);
      handlePrice = true;
    } else {
      assigns.sort((a, b) => a.trade_price - b.trade_price);
      printCoin(assigns);
      handlePrice = false;
    }
  });
  sortRate.addEventListener("click", () => {
    if (!handleRate) {
      assigns.sort((a, b) => b.signed_change_rate - a.signed_change_rate);
      printCoin(assigns);
      handleRate = true;
    } else {
      assigns.sort((a, b) => a.signed_change_rate - b.signed_change_rate);
      printCoin(assigns);
      handleRate = false;
    }
  });
}
```

JavaScript

```
sortTrade.addEventListener("click", () => {
  if (!handleTrade) {
    assigns.sort(
      (a, b) => Math.round(b.acc_trade_price) - Math.round(a.acc_trade_price)
    );
    printCoin(assigns);
    handleTrade = true;
  } else {
    assigns.sort(
      (a, b) => Math.round(a.acc_trade_price) - Math.round(b.acc_trade_price)
    );
    printCoin(assigns);
    handleTrade = false;
  }
});

printCoin(assigns);
}
```


프로세스별 JavaScript 구현 코드

html에 받아온 데이터를 삽입

```
<!-- 코인 리스트 -->
<ul id="coin_list"></ul>
```

```
// html에 받아온 데이터를 삽입
function printCoin(assigns) {
  coinList.innerHTML = "";
  assigns.forEach((e) => {
    const coinDetails = document.createElement("li");
    const korean_name = document.createElement("div");
    const english_name = document.createElement("div");
    const market_name = document.createElement("div");
    const trade_price = document.createElement("div");
    const signed_change_rate = document.createElement("div");
    const acc_trade_price = document.createElement("div");

    coinDetails.classList.add("coin-each");
    korean_name.innerText = e.korean_name;
    english_name.innerText = e.english_name;
    market_name.innerText = e.market;
    trade_price.innerText = `${e.trade_price.toLocaleString("ko-KR")}`;
    signed_change_rate.innerText = `${e.signed_change_rate * 100}.toFixed(2) + "%";
    const price = e.acc_trade_price.toFixed(0);
    const price_num = price.slice(-(price + "").length, -6);
    acc_trade_price.innerText = `${Number(price_num).toLocaleString("ko-KR")}백만`;

    coinDetails.appendChild(korean_name);
    coinDetails.appendChild(english_name);
    coinDetails.appendChild(market_name);
    coinDetails.appendChild(trade_price);
    coinDetails.appendChild(signed_change_rate);
    coinDetails.appendChild(acc_trade_price);
    coinList.appendChild(coinDetails);
  });
}
```

```
signed_change_rate.innerText = `${e.signed_change_rate * 100}.toFixed(2) + "%";
const price = e.acc_trade_price.toFixed(0);
const price_num = price.slice(-(price + "").length, -6);
acc_trade_price.innerText = `${Number(price_num).toLocaleString("ko-KR")}백만`;

coinDetails.appendChild(korean_name);
coinDetails.appendChild(english_name);
coinDetails.appendChild(market_name);
coinDetails.appendChild(trade_price);
coinDetails.appendChild(signed_change_rate);
coinDetails.appendChild(acc_trade_price);
coinList.appendChild(coinDetails);
});
}
```

HTML

JavaScript

프로세스별 JavaScript 구현 코드

검색 기능

```
// 검색기능
const inputs = document.querySelector("#query");
inputs.addEventListener("keyup", (e) => {
  e.preventDefault();
  const value = e.target.value;
  const list = document.querySelectorAll(".coin-each");

  list.forEach((e, i) => {
    const name = e.firstChild;
    name.innerText.indexOf(value) > -1
      ? (e.style.display = "block")
      : (e.style.display = "none");
  });
});
```



JavaScript

05

Error Messages Cause and Solution

Error Messages Cause and Solution

에러메세지 & 수정 전 코드

Live reload enabled. [index.html:70](#)

✖ Uncaught TypeError: Cannot read properties of undefined (reading 'forEach') [index.js:107](#)

- at printCoin ([index.js:107](#))
- at sort ([index.js:101](#))
- at assignData ([index.js:51](#))
- at XMLHttpRequest.xhr.onreadystatechange ([index.js:32](#))



Error Message

```
const sortTrade = document.querySelector("#sort-accTrade");
let handlePrice = false;
let handleRate = false;
let handleTrade = false;
sortPrice.addEventListener("click", () => {
  if (!handlePrice) {
    assigns.sort((a, b) => b.trade_price - a.trade_price);
    printCoin(assigns);
    handlePrice = true;
  } else {
    assigns.sort((a, b) => a.trade_price - b.trade_price);
    printCoin(assigns);
    handlePrice = false;
  }
});
sortRate.addEventListener("click", () => {
  if (!handleRate) {
    assigns.sort((a, b) => b.signed_change_rate - a.signed_change_rate);
    printCoin(assigns);
    handleRate = true;
  } else {
    assigns.sort((a, b) => a.signed_change_rate - b.signed_change_rate);
    printCoin(assigns);
    handleRate = false;
  }
});
sortTrade.addEventListener("click", () => {
  if (!handleTrade) {
    assigns.sort(
      (a, b) => Math.round(b.acc_trade_price) - Math.round(a.acc_trade_price)
    );
    printCoin(assigns);
    handleTrade = true;
  } else {
    assigns.sort(
      (a, b) => Math.round(a.acc_trade_price) - Math.round(b.acc_trade_price)
    );
    printCoin(assigns);
    handleTrade = false;
  }
});
printCoin();
}
```

JavaScript

Error Messages Cause and Solution

수정 후 코드 & 원인과 해결 방법

```
sortPrice.addEventListener('click', () => {
  if (!handlePrice) {
    assigns.sort((a, b) => b.trade_price - a.trade_price);
    printCoin(assigns);
    handlePrice = true;
  } else {
    assigns.sort((a, b) => a.trade_price - b.trade_price);
    printCoin(assigns);
    handlePrice = false;
  }
});
sortRate.addEventListener('click', () => {
  if (!handleRate) {
    assigns.sort((a, b) => b.signed_change_rate - a.signed_change_rate);
    printCoin(assigns);
    handleRate = true;
  } else {
    assigns.sort((a, b) => a.signed_change_rate - b.signed_change_rate);
    printCoin(assigns);
    handleRate = false;
  }
});
sortTrade.addEventListener('click', () => {
  if (!handleTrade) {
    assigns.sort(
      (a, b) => Math.round(b.acc_trade_price) - Math.round(a.acc_trade_price)
    );
    printCoin(assigns);
    handleTrade = true;
  } else {
    assigns.sort(
      (a, b) => Math.round(a.acc_trade_price) - Math.round(b.acc_trade_price)
    );
    printCoin(assigns);
    handleTrade = false;
  }
});
printCoin(assigns);
}
```

JavaScript

원인

정확한 위치에 매개변수를 대입하지 않아 에러가 발생

해결 방법

코드 확인 후 매개변수를 대입하여 해결

Error Messages Cause and Solution

에러메세지 & 수정 전 코드

✖ Uncaught ReferenceError: res is not defined
at app.js:56

app.js:56



Error Message

```
function upbitApi() {  
  const xhr = new XMLHttpRequest();  
  const method = "GET";  
  const url = "https://api.upbit.com/v1/market/all?isDetails=false";  
  
  xhr.onreadystatechange = (e) => {  
    const { target } = e;  
  
    if (target.readyState === XMLHttpRequest.DONE) {  
      if (target.status === 200) {  
        const res = JSON.parse(target.response);  
        // 응답받아 parse로 파싱해서 JSON형태로 리턴시킨 값 대입 -> data  
        // 응답받은 데이터를 전역변수에 저장  
  
      } else {  
        const status = parseInt(target.status / 100);  
        let error = null;  
        if (status === 4) {  
          error = `[${target.status}] ${target.statusText}요청주소가 잘못되었습니다.`;  
        } else if (status === 5) {  
          error = `[${target.status}] ${target.statusText}서버의 응답이 없습니다.`;  
        } else {  
          error = `[${target.status}] ${target.statusText}요청에 실패했습니다.`;  
        }  
        alert(error);  
      }  
    }  
  };  
  xhr.open(method, url);  
  xhr.send();  
}  
upbitShow(res);  
upbitApi();
```

JavaScript

Error Messages Cause and Solution

수정 후 코드 & 원인과 해결 방법

```
function upbitApi() {
  const xhr = new XMLHttpRequest();
  const method = "GET";
  const url = "https://api.upbit.com/v1/market/all?isDetails=false";

  xhr.onreadystatechange = (e) => {
    const { target } = e;

    if (target.readyState === XMLHttpRequest.DONE) {
      if (target.status === 200) {
        const res = JSON.parse(target.response);
        // 응답받아 parse로 파싱해서 JSON형태로 리턴시킨 값 대입 -> data
        // 응답받은 데이터를 전역변수에 저장
        upbitShow(res);
      } else {
        const status = parseInt(target.status / 100);
        let error = null;
        if (status === 4) {
          error = `[${target.status}] ${target.statusText}요청주소가 잘못되었습니다.`;
        } else if (status === 5) {
          error = `[${target.status}] ${target.statusText}서버의 응답이 없습니다.`;
        } else {
          error = `[${target.status}] ${target.statusText}요청에 실패했습니다.`;
        }
        alert(error);
      }
    }
  };
  xhr.open(method, url);
  xhr.send();
}
```

JavaScript

원인

응답받은 데이터를 화면에 보여주는 함수의 호출 위치로 인한 오류

해결 방법

데이터를 요청, 응답받는 함수 내부에서 호출하여 문제 해결

Error Messages Cause and Solution

에러메세지 & 수정 전 코드

```
✖ ▼Uncaught TypeError: Cannot read properties of null (reading 'appendChild')
    at XMLHttpRequest.xhr.onreadystatechange (upbit-coin.html:51)
    xhr.onreadystatechange    @ upbit-coin.html:51
    XMLHttpRequest.send (async)
    (anonymous)              @ upbit-coin.html:72
    (anonymous)              @ upbit-coin.html:73
```



Error Message

```
(() => {
  const xhr = new XMLHttpRequest();
  const method = 'GET';
  const url = 'https://api.upbit.com/v1/market/all?isDetails=false';

  xhr.onreadystatechange = (e) => {
    const { target } = e;
    console.log(e);

    if (target.readyState === XMLHttpRequest.DONE) {
      if (target.status === 200) {
        const req = JSON.parse(target.response);
        const list = document.querySelector("coinList");

        const li1 = document.createElement("li");
        li1.innerHTML = req.korean_name;

        const li2 = document.createElement("li");
        li1.innerHTML = req.english_name;

        list.appendChild(li1);
        list.appendChild(li2);

      } else {
        const s = parseInt(target.status / 100);
        let errMsg = null;

        if (s == 4) {
          errMsg = '[' + target.status + ']' + target.statusText + ' - 요청 주소가 잘못되었습니다.';
        } else if (s == 5) {
          errMsg = '[' + target.status + ']' + target.statusText + ' - 서버의 응답이 없습니다.';
        } else {
          errMsg = '[' + target.status + ']' + target.statusText + ' - 요청에 실패했습니다.';
        }

        alert(errMsg);
      }
    }
  };

  xhr.open(method, url);
  xhr.send();
})();
```

1) # 아이디로 받았어야 하는데 누락되어 오류 뜸

2) 배열로 받아오지 못해서 undefined 뜸

JavaScript

Error Messages Cause and Solution

수정 후 코드 & 원인과 해결 방법

```
(() => {  
  const xhr = new XMLHttpRequest();  
  const method = 'GET';  
  const url = 'https://api.upbit.com/v1/market/all?isDetails=false';  
  
  xhr.onreadystatechange = (e) => {  
    const { target } = e;  
    console.log(e);  
  
    if (target.readyState === XMLHttpRequest.DONE) {  
      if (target.status === 200) {  
        const req = JSON.parse(target.response);  
        const list = document.querySelector("#coinList");  
  
        console.log(req);  
  
        req.forEach((e, i) => {  
          const li1 = document.createElement("li");  
          li1.innerHTML = e.korean_name;  
  
          list.appendChild(li1);  
        });  
  
        req.forEach((e, i) => {  
          const li2 = document.createElement("li");  
          li2.innerHTML = e.english_name;  
  
          list.appendChild(li2);  
        });  
      } else {  
        const s = parseInt(target.status / 100);  
        let errMsg = null;  
  
        if (s == 4) {  
          errMsg = '[' + target.status + ']' + target.statusText + ' - 요청 주소가 잘못되었습니다.';  
        } else if (s == 5) {  
          errMsg = '[' + target.status + ']' + target.statusText + ' - 서버의 응답이 없습니다.';  
        } else {  
          errMsg = '[' + target.status + ']' + target.statusText + ' - 요청에 실패했습니다.';  
        }  
  
        alert(errMsg);  
      }  
    }  
  };  
  
  xhr.open(method, url);  
  xhr.send();  
})();
```

1) 아이디값으로 불러옴

2) forEach구문을 써서 배열로 받아옴

원인

querySelector 안에 아이디 값으로 받아와야 할 것을 이름만 작성하여 값을 받아오지 못해 undefined 에러 발생

해결 방법

querySelector 안에 #붙혀 원하는 요소로 치환하여 해결
forEach구문으로 값을 올바르게 받아 와 해결

JavaScript

업비트 Open API

Thank you

2021.12.07

차지환,강예지,서지영,김종규