

Spring & JPA Project

Campnic *by* 와장창 2.0

박찬미

남기문

이수민



목차

01

프로젝트
기획

02

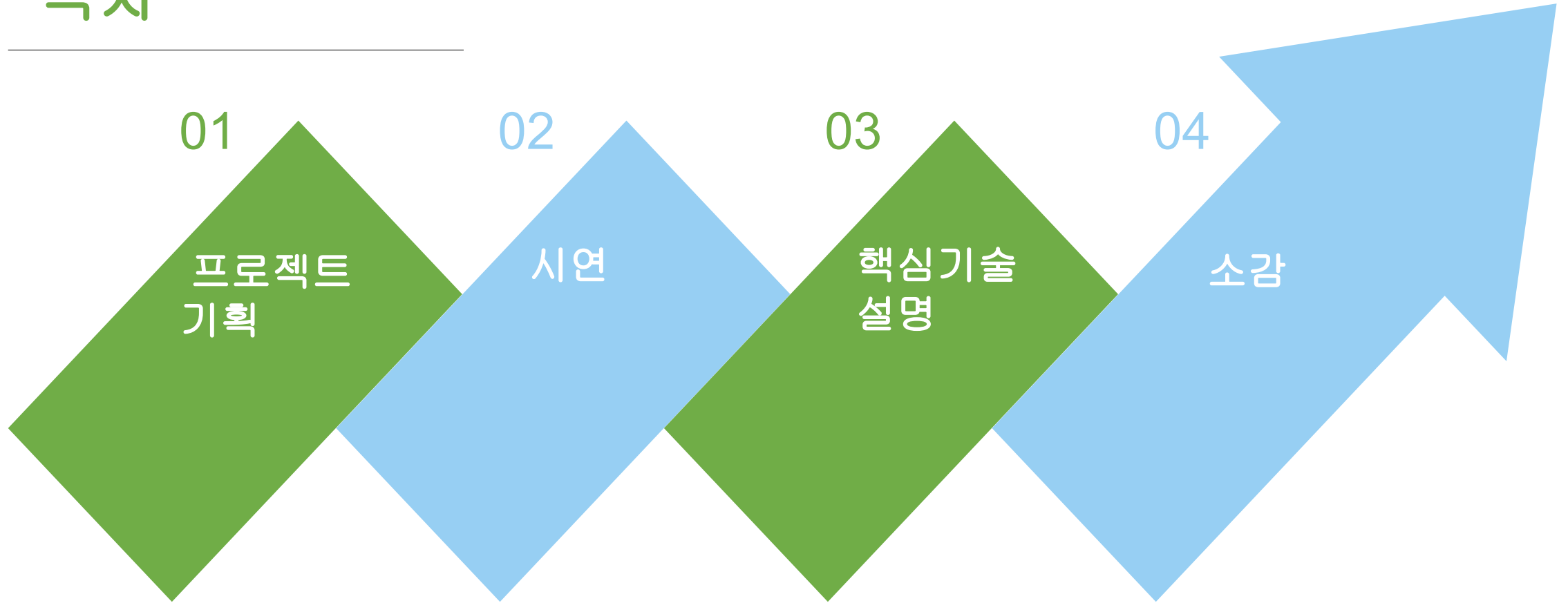
시연

03

핵심기술
설명

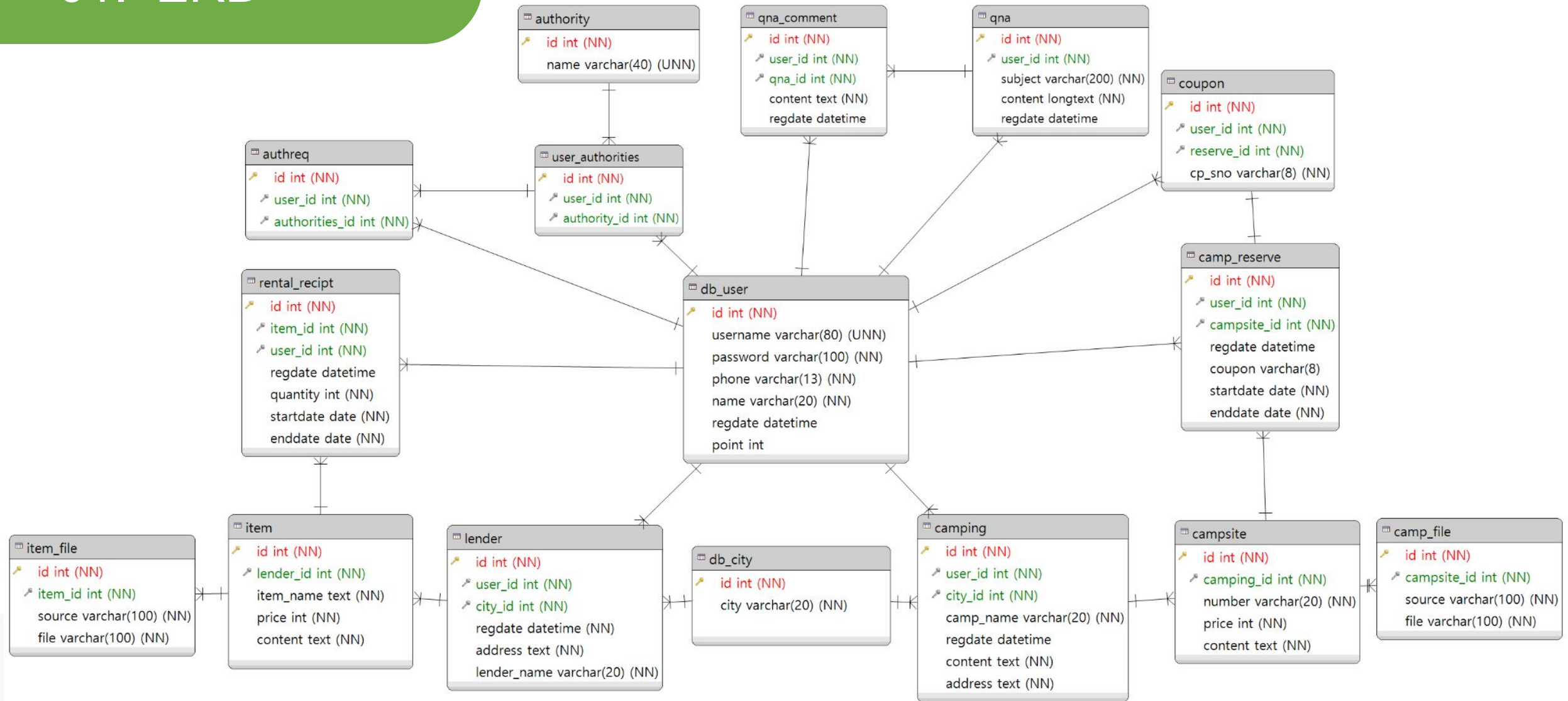
04

소감



1.프로젝트 기획

01. ERD



- 기획




campnic



02. 역할

박찬미

- 권한인증/인가
- QnA
- 로그인api
- 헤더/메인
- 마이페이지

남기문

- 캠핑(유저)
- 캠핑(관리자)
- 쿠폰

이수민

- 캠핑용품
유저용
- 캠핑용품
관리자용
- 날씨api

03. 기술 스택



1. 프로젝트 기획

2.시연

[user]

- 1) 카카오 로그인
- 2) 캠핑예약 - 도시로 검색, 퇴실일이 입실일보다 빠르면 alert
- 3) 캠핑영수증 확인- 쿠폰 등록하기(중복 안되는 거 확인)
- 4) 아이템 렌탈 - 퇴실일이 입실일보다 빠르면 alert
- 5) 렌탈 영수증 - 날짜가 당일로 예약, 대여일이 오늘이면 취소 불가
- 6) 관리자모드 버튼 확인
- 7) mypage- camping 권한 신청

[admin]

- 8) admin -> 권한 받고 -> city 작성 -> qna 글 작성 -> 공지글 위치 확인

[camping/admin]

- 9) camping1 -> 관리자 모드 -> 캠핑장 등록, 캠핑구역 등록

[lender/admin]

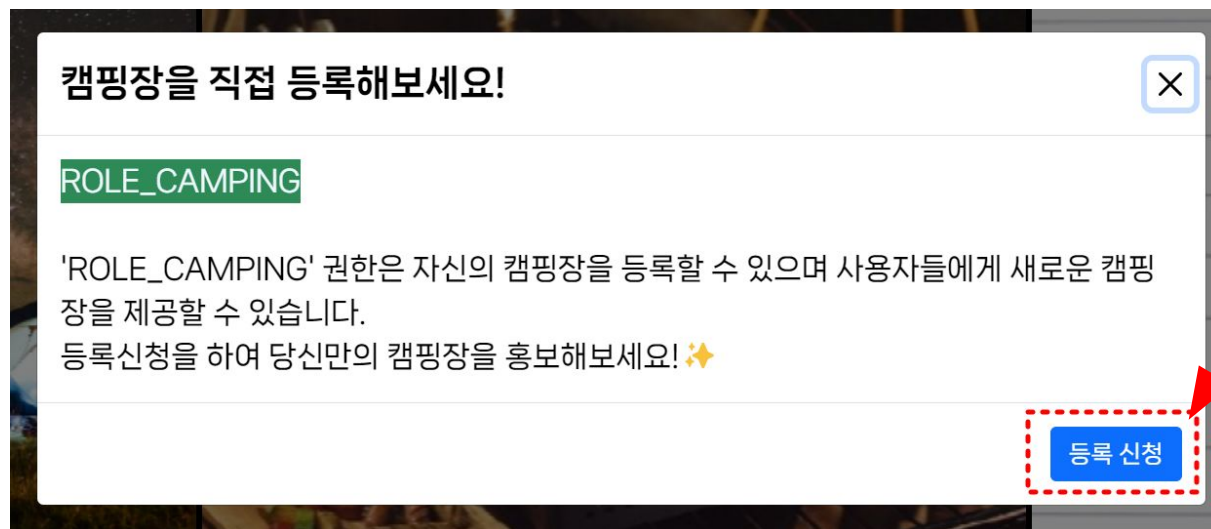
- 10) lender1 -> 관리자 모드 -> 사업장 등록, 아이템 등록

3.핵심기술설명

1. 권한요청 & 권한허가
2. **City**페이지 구현
3. 로그인 **api** / 홈 프론트

- 권한요청(멤버)

박찬미



UserController

```

@ParkChami
@PostMapping("/adminreq")
public String adminReq( String auth
                        , String username
                        , Model model
){
    userService.registerAuthReq(auth, username);
    return "redirect:mypage";
}

```

```

public void registerAuthReq(String auth, String username){
    User u = userRepository.findByUsername(username);
    Authority au = authorityRepository.findByName(auth);
    Authreq ar = new Authreq();
    ar.setUser(u);
    ar.setAuthority(au);
    authreqRepository.save(ar);    // authreq에 INSERT
}

```

UserService

- 권한허가(관리자)

박찬미

User	Authority	O/X
USER2	ROLE_CAMPING	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
USER2	ROLE_LENDER	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
USER3	ROLE_CAMPING	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
USER3	ROLE_LENDER	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

UserService

```
// 권한 수락
1 usage 박찬미
public void acceptAuth(String authreqId, String userId, String authId){
    User user = userRepository.findById(Long.parseLong(userId)).orElse( other: null);
    Authority auth = authorityRepository.findById(Long.parseLong(authId)).orElse( other: null);
    user.addAuthority(auth);

    Authreq a = authreqRepository.findById(Long.parseLong(authreqId)).orElse( other: null);
    authreqRepository.delete(a);
}

// 권한 거절
1 usage 박찬미
public void refuseAuth(String authreqId){
    Authreq a = authreqRepository.findById(Long.parseLong(authreqId)).orElse( other: null);
    authreqRepository.delete(a);
}
```

AdminController

```
박찬미
@GetMapping("/authCheck")
public String authcheck(Model model){
    model.addAttribute( attributeName: "list", userService.getAllAuthreq());
    return "/admin/authCheck";
}

박찬미
@PostMapping("/authAccept")
public String authAccept(String authreqId, String userId, String authId){
    userService.acceptAuth(authreqId, userId, authId);
    return "redirect:/admin/authCheck";
}

박찬미
@PostMapping("/authRefuse")
public String authRefuse(String authreqId){
    userService.refuseAuth(authreqId);
    return "redirect:/admin/authCheck";
}

// 권한등록 리스트
1 usage 박찬미
public List<Authreq> getAllAuthreq(){
    List<Authreq> list = authreqRepository.findAll();
    return list;
}
```

UserService

- City 페이지 구현

박찬미

지역 등록 🏠

지역이름

지역을 입력하세요

등록

지역 목록 📄

Citylist

서울시

삭제

수원시

삭제

하남시

삭제

```
<div style="...">
<h2>지역 등록 🏠</h2>
<form th:action="@{/admin/city}" method="post">
  <div class="mb-3 mt-3">
    <label for="subject" style="...">지역이름</label>
    <input type="text" class="form-control" id="subject" placeholder="지역을 입력하세요" name="city">
  </div>
  <button type="submit" class="btn btn-outline-dark" style="...">등록</button>
</form>
</div>

<div>
<h2>지역 목록 📄</h2>
<table class="table table-hover">
  <thead class="table-success" style="...">
    <tr>
      <th>Citylist</th>
    </tr>
  </thead>
  <tbody style="...">
    <tr th:each="dto : ${list}">
      <td>[[${dto.city}]]</td>
      <td>
        <form name="frmDelete" th:action="@{/admin/cityDel}" method="post">
          <input type="hidden" name="id" th:value="${dto.id}">
          <button class="btn btn-danger" id="btnDel">삭제</button>
        </form>
      </td>
    </tr>
  </tbody>
</table>
</div>
```

HTML

AdminController

박찬미

```
@GetMapping("/city")
public void city(Model model) { model.addAttribute( attributeName: "list", lenderService.cityList()); }
```

박찬미

```
@PostMapping("/city")
public String cityOk(@ModelAttribute("dto") City city, Model model){
  model.addAttribute( attributeName: "result", lenderService.addCity(city));
  return "/admin/cityOk";
}
```

박찬미

```
@PostMapping("/cityDel")
public String deleteOk(long id, Model model){
  model.addAttribute( attributeName: "result", lenderService.delCity(id));
  return "/admin/cityDelOk";
}
```

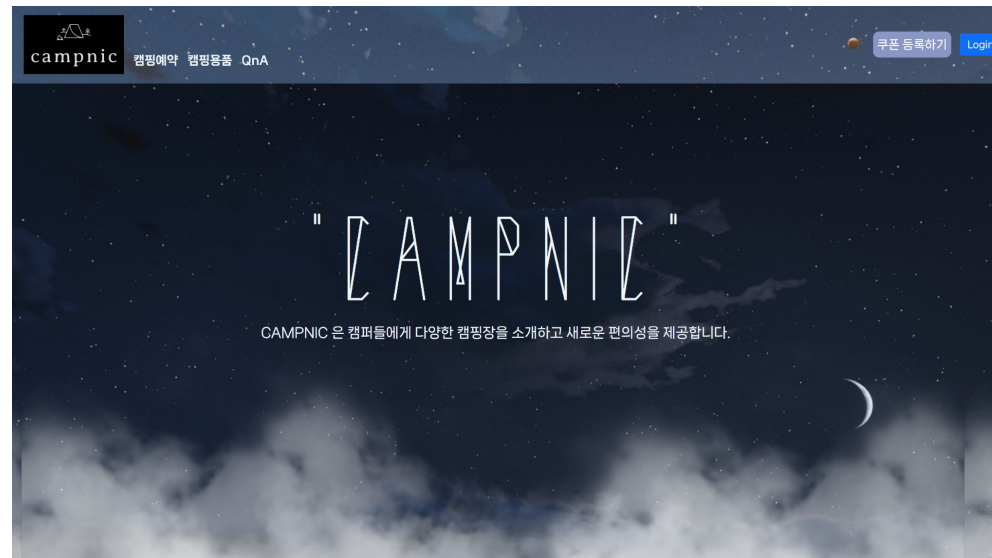

- 로그인 api / 홈

JavaScript

```
Kakao.init( ); //발급받은 키 중 javascript키를 사용해  
//console.log(Kakao.isInitialized()); // sdk초기화여부판단  
kakaoLogout();  
  
//카카오로그인  
function loginWithKakao(){  
  Kakao.Auth.login({  
    success: function(response){  
      Kakao.API.request({  
        url: '/v2/user/me',  
        success: function(response){  
          let kakaoForm = document.createElement("form");  
          kakaoForm.setAttribute("charset", "UTF-8");  
          kakaoForm.setAttribute("method", "post");  
          kakaoForm.setAttribute("action", "apiLogin");  
          document.body.appendChild(kakaoForm);  
  
          console.log(response);  
  
          const userId = "kakao" + response.id;  
          console.log(userId);  
  
          const input1 = document.createElement('input');  
          input1.type = "hidden"; input1.name = "id"; input1.value = userId;  
          kakaoForm.append(input1);  
  
          kakaoForm.submit();  
        }  
      });  
    }  
  });  
}
```

```
<form id="frm" th:action="@{/user/login}" method="POST">  
  <input type="hidden" name="username" th:value="${username}">  
  <input type="hidden" name="password" th:value="${username.toUpperCase()}+!!">  
</form>  
<body>  
<script>  
  const frm = document.getElementById("frm");  
  frm.submit();  
</script>
```

HTML



CSS

```
background: url("/img/cloud.png") center top repeat-x;  
padding-top: 100px;  
padding-bottom: 10px;  
width: 100%;  
padding: 0px;  
margin: 0px;  
animation: awan-animasi 2s linear infinite;  
-ms-animation: awan-animasi 2s linear infinite;  
-moz-animation: awan-animasi 2s linear infinite;  
-webkit-animation: awan-animasi 20s linear infinite;  
}
```

```
.conA-bg{  
  position: relative;
```

```
.campnic{  
  font-family: Delta;  
  position: absolute;
```

HTML

```
<div class="conA">  
  <div class="conA-bg">  
    <p class="campnic"> C A M P N I C </p>  
    <p class="inform"> CAMPNIC은 캠퍼들에게 다양한
```

박찬미

1. Coupon 2. Point

- Coupon

남기문

```
public boolean findByCoupon(String coupon) {  
    boolean result = false;  
    CampReserve campReserve = campReserveRepository.findByCoupon(coupon);  
    if(campReserve != null){  
        result = true;  
    }  
    return result;  
}
```

1 usage 남기문

```
public boolean couponCheck(String coupon){  
    boolean result = false;  
    Coupon couponCheck = couponRepository.findByCpNum(coupon);  
    if(couponCheck == null) {  
        result = true;  
    }  
    return result;  
}
```

등록할 쿠폰번호가 발급된
쿠폰인지 확인

localhost:8081 내용:

존재하지 않는 쿠폰 번호입니다.

확인

등록할 쿠폰번호가 이미
사용되었는지 확인

localhost:8081 내용:

이미 사용한 쿠폰입니다.

확인

- Coupon

남기문

```
if(campingService.findByCoupon(coupon.getCpNum())){  
  
    if(campingService.couponCheck(coupon.getCpNum())) {  
        model.addAttribute( attributeName: "result", campingService.addCoupon(coupon));  
        model.addAttribute( attributeName: "dto", coupon);  
  
        User user = U.getLoggedInUser();  
  
        // 위 정보는 session 의 정보이고, 일단 DB 에서 다시 읽어온다  
        user = userRepository.findById(user.getId()).orElse( other: null);  
  
        int point = user.getPoint() + 5000;  
  
        user.setPoint(point);  
  
        userRepository.saveAndFlush(user);  
  
        return "camp/couponOK";  
    }else{  
        return "camp/useCoupon";  
    }  
}  
else {  
    return "camp/noReserve";  
}
```

두가지 조건 모두
충족되면 현재
로그인된 User
정보를 읽어와 해당
User에 Point 지급

- Point 적립

남기문

1.point 정보 변경 안됨
->session정보 수정 후 다시
불러오기

USER1 님 환영합니다!(Point: 0)

```
protected Authentication createNewAuthentication(Authentication currentAuth, String username)
{
    UserDetails newPrincipal = principalDetailsService.loadUserByUsername(username);
    UsernamePasswordAuthenticationToken newAuth =
        new UsernamePasswordAuthenticationToken(
            newPrincipal,
            currentAuth.getCredentials(),
            newPrincipal.getAuthorities()
        );
    newAuth.setDetails(currentAuth.getDetails());
    return newAuth;
}
```

현재 사용자 인증
정보를 새로운
사용자 정보로
업데이트

```
// 뒤편에서
userService.updateUser(user.getId(), point);

//수정된 유저정보 principal에 업데이트
Authentication authentication = SecurityContextHolder.getContext().getAuthentication();
UserDetails userAccount = (UserDetails) authentication.getPrincipal();
SecurityContextHolder.getContext().setAuthentication(
    createNewAuthentication(authentication, userAccount.getUsername()));
```

업데이트된
정보를
principal에
업데이트

- 1. 날씨 api**
- 2. 렌탈 취소 제한**
- 3. 지역별 용품 목록**

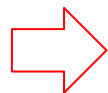
- 날씨 api(1)

이수민

api key 노출 -> 서비스단에 옮김

```
DevTools - 127.0.0.1:5500/test.html
DevTools is now available in Korean! Always match Chrome's language Switch
Elements Console Sources Network Performance Men
<html lang="ko">
  <head> </head>
  <body>
    
    <!-- Code injected by live-server -->
    <script> </script> == $0
    <script>
      document.addEventListener('DOMContentLoaded', function() {
        {
          fetchWeather();
        });

        function fetchWeather() {
          fetch('http://api.openweathermap.org/data/2.5/weather?q=Seoul&appid=[REDACTED]')
            .then(response => response.json())
            .then(data => {
              const icon = data.weather[0].icon;
              const iconUrl = `http://openweathermap.org/img/w/${icon}.png`;
              const weatherIcon = document.getElementById('weather-icon');
              weatherIcon.src = iconUrl;
            })
            .catch(error => console.error(error));
        }
      });
    </script>
  </body>
</html>
```



```
@Service
public class WeatherService {
    private static final String API_KEY = "[REDACTED]";
    private static final String API_URL = "http://api.openweathermap.org/data/2.5/weather?q={city}&appid={apiKey}";

    public Weather getWeather(String city) {
        RestTemplate restTemplate = new RestTemplate();
        ResponseEntity<Weather> response = restTemplate.getForEntity(API_URL, Weather.class, city, API_KEY);
        return response.getBody();
    }
}
```

```
public class Weather {
    private String name;
    private WeatherInfo[] weather;
}

class WeatherInfo {
    @JsonProperty("icon")
    private String icon;
}
```

- 날씨 api(2)

이수민

1. controller단에서 사용 -> api key 노출 x
2. 필요한 부분만 js로 parsing

```
// 날씨
@GetMapping("/weather")
@ResponseBody
public Weather getWeather() { return weatherService.getWeather( city: "Seoul"); }
```

localhost:8081/weather

{ "name": "Seoul", "weather": [{ "icon": "01n" }] }

DevTools - localhost:8081/weather

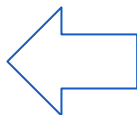
DevTools is now available in Korean! Always match Chrome's language Switch D

Elements Console Sources Network Performance Memo

```
<html>
<head>
  <meta name="color-scheme" content="light dark">
</head>
<body>
  <pre style="word-wrap: break-word; white-space: pre-wrap;">
    { "name": "Seoul", "weather": [ { "icon": "01n" } ] }</pre>
</body>
</html>
```

```
<script>
  document.addEventListener('DOMContentLoaded', function() {
    fetchWeather();
  });

  function fetchWeather() {
    fetch('/weather')
      .then(response => response.json())
      .then(data => {
        const icon = data.weather[0].icon;
        const iconUrl = `http://openweathermap.org/img/w/${icon}.png`;
        const weatherIcon = document.getElementById('weather-icon');
        weatherIcon.src = iconUrl;
      })
      .catch(error => console.error(error));
  }
</script>
```



- 렌탈 취소 제한

이수민

80000000

서울시 동작구

23. 4. 7 ~ 23. 4. 8.

2023-04-07



대여일이
이미
지났다

오후 9:35

2023-04-08

localhost:8081 내용:

취소가 불가능합니다.

확인

```
// 취소일이 렌트 당일인지 check
public boolean checkDays(RentalRecipt rentalRecipt){
    LocalDate sdate = rentalRecipt.getSdate();
    if(sdate.isAfter(LocalDate.now())) {return true;}
    return false;
}

// rental 취소
public int delRental(Long id){
    RentalRecipt rentalRecipt = rentalReciptRepository.findById(id).orElse( other: null);
    if(rentalRecipt != null && checkDays(rentalRecipt)){
        rentalReciptRepository.delete(rentalRecipt);
        return 1;
    }
    return 0;
}
```

일반적으로 취소기간이 정해져 있음(대여일 > 취소일 -> 취소가능) -> 서비스단에서 취소일을 check

- 지역별용품목록

이수민

보통 캠핑용품은 캠핑장 주변 -> 지역별 캠핑용품 검색 기능

```
// 지역 Item 목록
@Transactional
public List<Item> searchItemList(Long cityId){
    City city = cityRepository.findById(cityId).orElse(new City());
    List<Lender> lenderList = lenderRepository.findByCity(city);
    List<Item> itemList = new ArrayList<>();
    for(Lender l : lenderList){
        if(city.getCity().equals(l.getCity().getCity())) {
            List<Item> items = itemRepository.findByLender(l);
            itemList.addAll(items);
        }
    }
    return itemList;
}
```

도시

도시

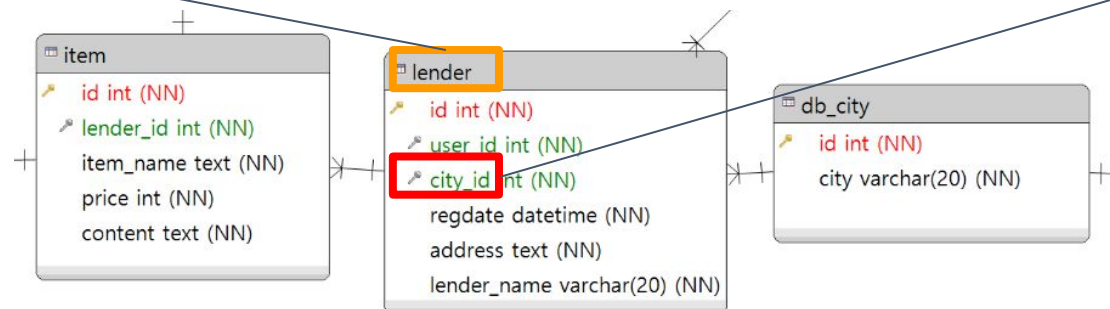
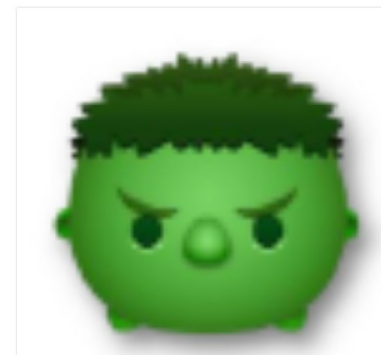
서울시

수원시

하남시

모두보기

캠핑용품



4. 소감

박찬미

남기문

이수민



THANK YOU

