KB금융그룹 | 국민의 평생 금융파트너

2024년 상반기 K-디지털 트레이닝

회원관리-정보 수정(백엔드)

[KB] IT's Your Life

KB국민은행

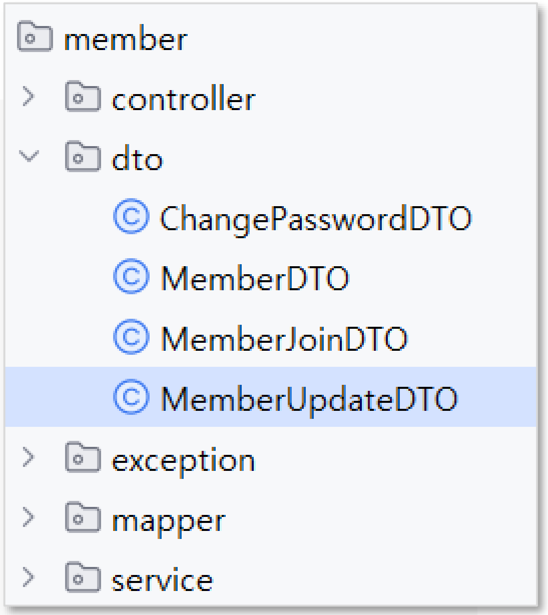
회원관리-정보 수정

MemberUpdateDTO.java

```
package org.scoula.member.dto;
…
@Data
@NoArgsConstructor
@AllArgsConstructor
public class MemberUpdateDTO {
    private String username;
    private String password;
    private String email;

    MultipartFile avatar;

    public MemberVO toVO() {
        return MemberVO.builder()
                .username(username)
                .email(email)
                .build();
    }
}
```



```
member
> controller
v dto
    ChangePasswordDTO
    MemberDTO
    MemberJoinDTO
    MemberUpdateDTO
> exception
> mapper
> service
```

회원관리-정보 수정

✓ 정보의 수정

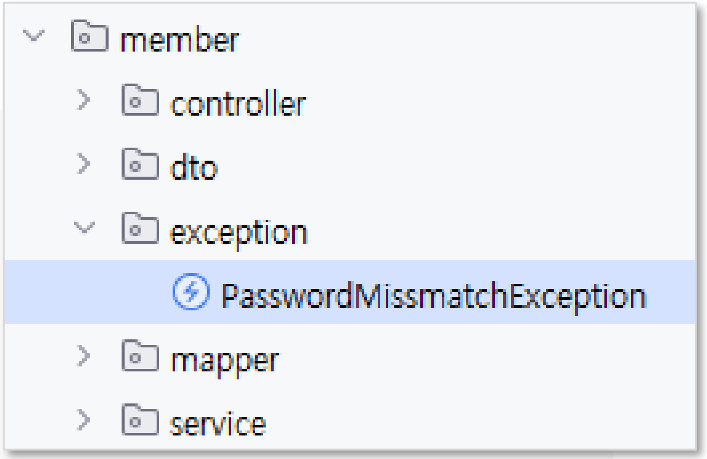
- 해당 비밀번호가 맞는 경우에만 수정
- 비밀번호가 틀리면 PasswordMissMatchException 발생

회원관리-정보 수정

PasswordMissmatchException.java

```java
package org.scoula.member.exception;

public class PasswordMissmatchException extends RuntimeException{
    public PasswordMissmatchException() {
        super("비밀번호가 일치하지 않습니다.");
    }
}
```



회원관리-정보 수정

MemberMapper.java

```
public interface MemberMapper {
     …

    int update(MemberVO member);

}
```

회원관리-정보 수정

MemberMapper.xml

```
<update id="update">
    UPDATE tbl_member
    SET
        email = #{email},
        update_date = now()
    WHERE username =#{username}
</update>
```

회원관리-정보 수정

MemberService.java

```
public interface MemberService {
    …

    MemberDTO update(MemberUpdateDTO member);

}
```

회원관리-정보 수정

KB국민은행

MemberServiceImpl.java

```java
@Override
public MemberDTO update(MemberUpdateDTO member) {
    MemberVO vo = mapper.get(member.getUsername());
    if(!passwordEncoder.matches(member.getPassword(),vo.getPassword())) {  // 비밀번호 일치 확인
        throw new PasswordMissmatchException();
    }

    mapper.update(member.toVO());
    saveAvatar(member.getAvatar(), member.getUsername());
    return get(member.getUsername());
}
```

회원관리-정보 수정

KB국민은행

MemberController

```
@PutMapping("/{username}")
public ResponseEntity<MemberDTO> changeProfile(MemberUpdateDTO member) {
    return ResponseEntity.ok(service.update(member));
}
```

KB금융그룹 | 국민의 평생 금융파트너

2024년 상반기 K-디지털 트레이닝

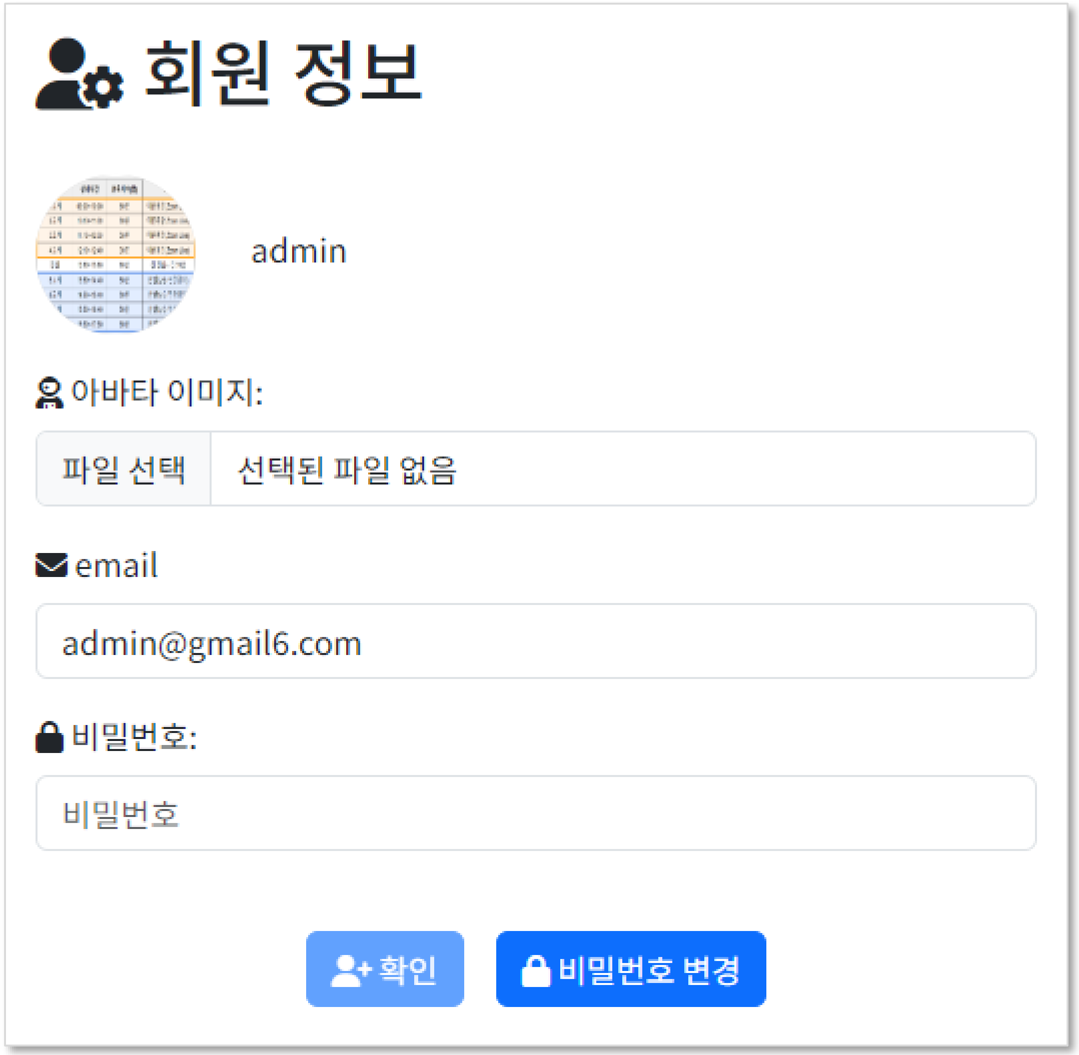
회원관리-정보 수정(프론트엔드)

[KB] IT's Your Life

KB국민은행

회원관리-정보 수정

- 인터셉터가 적용된 axios를 이용
  - 인증 헤더를 추가하고 요청해야 함
  - 수정 요청 성공 후 store에도 반영해야 함.



회원 정보

admin

아바타 이미지:

파일 선택 | 선택된 파일 없음

email

admin@gmail6.com

비밀번호:

비밀번호

회원관리-정보 수정

KB국민은행

api/authApi.js

```
import api from '@/api';
…
const headers = { 'Content-Type': 'multipart/form-data' };

export default {
  …
  async update(member) {
    const formData = new FormData();
    formData.append('username', member.username);
    formData.append('password', member.password);
    formData.append('email', member.email);

    if (member.avatar) {
      formData.append('avatar', member.avatar);
    }

    const { data } = await api.put(`${BASE_URL}/${member.username}`, formData, headers);
    console.log('AUTH PUT: ', data);
    return data;
  }
};
```

회원관리-정보 수정

KB국민은행

stores/auth.js

```
…
export const useAuthStore = defineStore('auth', () => {
  …
  const changeProfile = (member) => {
    state.value.user.email = member.email;
    localStorage.setItem('auth', JSON.stringify(state.value));
  };

  load();
  return { state, username, email, isLogin, changeProfile, login, logout, getToken };
});
```

회원관리-정보 수정

pages/auth/ProfilePage.vue

```
<script setup>
import authApi from '@/api/authApi';
import { useAuthStore } from '@/stores/auth';
import { computed, reactive, ref } from 'vue';

const auth = useAuthStore();

const avatar = ref(null);
const avatarPath = `/api/member/${auth.username}/avatar`;
const member = reactive({
  username: auth.username,
  email: auth.email,
  password: '',
  avatar: null,
});

const error = ref('');

const disableSubmit = computed(() => !member.email || !member.password);
```

회원관리-정보 수정

KB국민은행

pages/auth/ProfilePage.vue

```
const onSubmit = async () => {
  if (!confirm('수정하시겠습니까?')) return;

  if (avatar.value.files.length > 0) {
    member.avatar = avatar.value.files[0];
  }

  try {
    await authApi.update(member);
    error.value = '';
    auth.changeProfile(member);
    alert('정보를 수정하였습니다.');
  } catch (e) {
    error.value = e.response.data;
  }
};
</script>
```

회원관리-정보 수정

pages/auth/ProfilePage.vue

```
<template>
  <div class="mt-5 mx-auto" style="width: 500px">
    <h1><i class="fa-solid fa-user-gear my-3"></i> 회원 정보</h1>

    <form @submit.prevent="onSubmit">
      <div class="mb-3 mt-3">
         <img :src="avatarPath" class="avatar avatar-lg me-4" /> {{ member.username }}
      </div>

      <div class="mb-3 mt-3">
        <label for="avatar" class="form-label">
          <i class="fa-solid fa-user-astronaut"></i>
          아바타 이미지:
        </label>
        <input type="file" class="form-control" ref="avatar" id="avatar" accept="image/png, image/jpeg" />
      </div>
```

회원관리-정보 수정

pages/auth/ProfilePage.vue

```
    <div class="mb-3 mt-3">
      <label for="email" class="form-label">
        <i class="fa-solid fa-envelope"></i>
        email
      </label>
      <input type="email" class="form-control" placeholder="Email" id="email" v-model="member.email" />
    </div>

    <div class="mb-3">
      <label for="password" class="form-label">
        <i class="fa-solid fa-lock"></i>
        비밀번호:
      </label>
      <input type="password" class="form-control" placeholder="비밀번호" id="password" v-model="member.password" />
    </div>

    <div v-if="error" class="text-danger">{{ error }}</div>
```

회원관리-정보 수정

KB국민은행

pages/auth/ProfilePage.vue

```
        <div class="text-center">
          <button type="submit" class="btn btn-primary mt-4 me-3" :disabled="disableSubmit">
            <i class="fa-solid fa-user-plus"></i>
            확인
          </button>

          <router-link class="btn btn-primary mt-4" to="/auth/changepassword">
            <i class="fa-solid fa-lock"></i>
            비밀번호 변경
          </router-link>
        </div>
      </form>
    </div>
</template>
```