

```

package com.cybercenter.core.domain.resource.service;

import com.cybercenter.core.domain.resource.dto.ResourceRequestDTO;
import com.cybercenter.core.domain.resource.dto.ResourceResponseDTO;
import com.cybercenter.core.domain.resource.entity.ResourceEntity;
import com.cybercenter.filetoken.FileTokenUtils;
import lombok.RequiredArgsConstructor;
import org.springframework.beans.BeanUtils;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
@RequiredArgsConstructor
public class ResourceService {

    private final ResourceCacheService resourceCacheService;
    private final FileTokenUtils fileTokenUtils;

    @Value("${fileManager.modelTypeId.resource}")
    private int resourceModelType;

    public ResourceResponseDTO create(ResourceRequestDTO dto) {
        ResourceEntity entity = ResourceEntity.builder()
            .scopeld(dto.getScopeld())
            .type(dto.getType())
            .name(dto.getName())
            .coverImage(dto.getCoverImage())
            .description(dto.getDescription())
            .address(dto.getAddress())
            .website(dto.getWebsite())
            .phones(dto.getPhones())
            .email(dto.getEmail())
            .manager(dto.getManager())
            .isActive(true)
            .build();
        return toResponseDTO(resourceCacheService.save(entity));
    }

    public ResourceResponseDTO getResource(Long id, int scopeld) {
        return toResponseDTO(resourceCacheService.getByldAndScopeld(id, scopeld));
    }
}

```

```

public List<ResourceResponseDTO> getScopeResource(int scopelId) {
    return resourceCacheService.getScopeResource(scopelId).stream()
        .map(this::toResponseDTO)
        .toList();
}

public ResourceResponseDTO update(Long id, ResourceRequestDTO dto) {
    ResourceEntity entity = resourceCacheService.getByIdAndScopelId(id, dto.getScopelId());
    BeanUtils.copyProperties(dto, entity, "id", "createdAt");
    return toResponseDTO(resourceCacheService.save(entity));
}

private ResourceResponseDTO toResponseDTO(ResourceEntity entity) {
    ResourceResponseDTO dto = new ResourceResponseDTO();
    BeanUtils.copyProperties(entity, dto);
    if (entity.getCoverImage() != null) {
        dto.setFileKey(fileTokenUtils.generateToken(resourceModelType, entity.getScopelId(),
null, entity.getCoverImage()));
    }
    return dto;
}
}

```