

```

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

import com.cybercenter.core.domain.scope.dto.ScopeDTO;
import com.cybercenter.core.domain.scope.entity.Scope;
import com.cybercenter.core.domain.scope.mapper.ScopeMapper;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;

@Service
@RequiredArgsConstructor
public class ScopeService {

    private final ScopeCacheService scopeCacheService;
    private final ScopeMapper scopeMapper;

    public List<ScopeDTO> getScopes(Boolean enable) {
        List<Scope> all = scopeCacheService.findAll();

        return scopeMapper.toDTOs(
            Optional.ofNullable(enable)
                .map(e -> all.stream().filter(scope -> scope.isEnabled() ==
e).collect(Collectors.toList()))
                .orElse(all)
        );
    }

    public List<ScopeDTO> getScopes() {
        return scopeCacheService.findAll().stream().map(scopeMapper::toDTO).toList();
    }

    /**
     * Retrieves a Scope object representing a specific scope from the cache service based on
     the provided scope ID.
     *
     * @param scopelId The ID of the scope to retrieve.
     * @return A ScopeDTO object representing the specified scope.
     */
    public ScopeDTO findById(int scopelId) {
        return scopeMapper.toDTO(scopeCacheService.findById(scopelId));
    }

```

```
}
```

```
public ScopeDTO save(ScopeDTO dto) {  
    return scopeMapper.toDTO(scopeCacheService.save(scopeMapper.toEntity(dto)));  
}
```

```
/**
```

* Updates an existing scope in the cache service based on the details provided in a ScopeDTO and a scope ID.

* The method first sets the ID of the ScopeDTO to the provided scope ID.

* The ScopeDTO is then mapped to a Scope entity using the scopeMapper.

* The resulting Scope entity object is then saved to the cache service, effectively updating the existing scope.

* Finally, a success response DTO is returned indicating the success of the update operation.

```
*/
```

```
public void update(int scopeId, ScopeDTO dto) {  
    dto.setId(scopeId);  
    scopeCacheService.save(scopeMapper.toEntity(dto));  
}  
}
```