Edit

# Adding AddPhone and DeletePhone Methods

We are adding two more methods to IPersonAppService interface as shown below:

```
Task DeletePhone(EntityDto<long> input);
Task<PhoneInPersonListDto> AddPhone(AddPhoneInput input);
```

We could create a new, separated IPhoneAppService. It's your choice. But, we can consider Person as an aggregate and add phone related methods here. AddPhoneInput DTO is shown below:

```
public class AddPhoneInput
{
    [Range(1, int.MaxValue)]
    public int PersonId { get; set; }

    [Required]
    public PhoneType Type { get; set; }

    [Required]
    [MaxLength(PhoneConsts.MaxNumberLength)]
    public string Number { get; set; }
}
```

We used PhoneConsts.MaxNumberLength for Number field. You should create this consts in **.Core.Shared**.

```
public class PhoneConsts
{
    public const int MaxNumberLength = 16;
}
```

```
public async Task DeletePhone(EntityDto<long> input)
{
    await _phoneRepository.DeleteAsync(input.Id);
}

[AbpAuthorize(AppPermissions.Pages_Tenant_PhoneBook_EditPerson)]
public async Task<PhoneInPersonListDto> AddPhone(AddPhoneInput input)
{
    var person = _personRepository.Get(input.PersonId);
    await _personRepository.EnsureCollectionLoadedAsync(person, p => p.Phones);

    var phone = ObjectMapper.Map<Phone>(input);
    person.Phones.Add(phone);

    //Get auto increment Id of the new Phone by saving to database
    await CurrentUnitOfWork.SaveChangesAsync();

    return ObjectMapper.Map<PhoneInPersonListDto>(phone);
}
```



Then we add configuration for AutoMapper into CustomDtoMapper.cs like below:

```
configuration.CreateMap<AddPhoneInput, Phone>();
```

(Note: We injected **IRepository<Phone, long>** in the constructor and set to _phoneRepository field, as similar to _personRepository)

**DeletePhone** method is simple. It only deletes phone with given id.

**AddPhone** method **gets** the person from database and add new phone to person.Phones collection. Then is **save changes**. Saving changes causes inserting new added phone to database and get its **Id**. Because, we are returning a DTO that contains newly created phone informations including Id. So, it should be assigned before mapping in the last line. (Notice that; normally it's not needed to call CurrentUnitOfWork.SaveChangesAsync. It's automatically called at the end of the method. We called it in the method since we need to save entity and get its Id immediately. See UOW document for more.)

There may be different approaches for AddPhone method. You can directly work with a **phone repository** to insert new phone. They all have different pros and cons. It's your choice.

- Edit Mode for Phone Numbers