* Frontend
* Model (hrm-lib/lib/data-access/models):

1. index.ts imports all model classes from current dir.
2. Models are inherited from BaseModel.

* Service (hrm-lib/lib/data-access/services):

1. index.ts imports all service classes from current dir.
2. Services are inherited from BaseService.

* Route:

1. Specifies URL, component and data (e.g. title, animation,…)
2. Config which one is lazy loading.
3. URLs from hrm-lib/lib/config/urlConstants.ts use kebab-case.

* Component:

1. Inherited from BaseSomethingComponent. (e.g. BaseAdminComponent, BaseUserComponent, …)

* Module:

1. Imports child modules, components and services.

* Backend
* Controller

1. private readonly fields use camelCase and start with an underscore (e.g. \_employeeService).
2. Methods, Classes use PascalCase.
3. API routes use kebab-case.

* Model

1. DTOs: inherited from a BaseDto. Specific DTOs for searching/paging PaginationNoSort/SearchDataModel.
2. Models/Entities: are inherited from a EntityBase.
3. Both: use PascalCase for field names.
4. Mapping: many DTOs -> one entity and vice versa.

* Service

1. Interface: contains abstract methods.
2. Implementation of its interface.

* Use repository/unit of work.
* Services should be registered with scoped lifetime.