

Дипломная работа

Модуль **application** обязанность - запуск приложения. Содержит конфигурацию всего проекта.

- application/src/main/resources/application.properties- конфигурационный файл spring который содержит настройку приложения: параметры подключения в БД, настройку для миграции данных и т.д.
- application/src/main/java/com/gallery/application/configuration/ApplicationConfiguration.java - класс который агрегирует конфигурацию из модулей проекта
- application/src/main/java/com/gallery/application/GalleryApplication.java - класс для запуска Spring Boot проекта

Модуль **datatables-pagination-plugin** интерфейс api для взаимодействия с фреймворком datatables(<https://datatables.net/>).

- datatables-pagination-plugin/src/main/java/com/gallery/datatablespagination/model - каталог который содержит POJO классы описывающие модель запроса datatables.
- datatables-pagination-plugin/src/main/java/com/gallery/datatablespagination/service - каталог который содержит базовые сервисы и интерфейсы для реализации взаимодействия с datatables

Модуль **datatables-pagination-api** реализация интерфейса для взаимодействия с datatables(<https://datatables.net/>)

- datatables-pagination-api/src/main/java/com/goganesh/gallery/datatablespagination/configuration/DataTablesPaginationConfiguration.java - класс который создает бины данного модуля. Данный класс импортируется в модуль application
- datatables-pagination-api/src/main/java/com/goganesh/gallery/datatablespagination/controller - каталог который содержит контроллеры, которые обрабатывают запросы от клиента datatables. Реализация api datatables.
- datatables-pagination-api/src/main/java/com/goganesh/gallery/datatablespagination/dto - каталог который содержит DTO классы
- datatables-pagination-api/src/main/java/com/goganesh/gallery/datatablespagination/mapper - каталог который содержит классы реализуют логику трансформации model → dto
- datatables-pagination-api/src/main/java/com/goganesh/gallery/datatablespagination/model - каталог который содержит расширение базовой модели из datatables-pagination-plugin
- datatables-pagination-api/src/main/java/com/goganesh/gallery/datatablespagination/service - каталог который содержит классы сервисы инкапсулирующие логику получения данных

Модуль **migration** отвечает за миграцию структуры и данных в СУБД. Миграция данных выполняется средствами Liquibase.

- migration/src/main/java/com/goganesh/gallery/migration/configuration/MigrationConfiguration.java - класс который создает бины данного модуля. Данный класс импортируется в модуль application
- migration/src/main/resources/changelog/changelog-root.xml - файл который содержит пути для определения файлов миграции

- migration/src/main/resources/changelog/data - каталог который содержит файлы миграции

Модуль **model** отвечает за обработку бизнес логики. Взаимодействие с СУБД. Содержит POJO модель (Entity) и реализацию взаимодействия с СУБД. Слой доступа к БД.

- model/src/main/java/com/goganesh/gallery/model/configuration/ModelConfiguration.java - класс который создает бины данного модуля. Данный класс импортируется в модуль application
- model/src/main/java/com/goganesh/gallery/model/domain - каталог который POJO классы Entity
- model/src/main/java/com/goganesh/gallery/model/exception - каталог который содержит исключения приложения
- model/src/main/java/com/goganesh/gallery/model/repository - слой для взаимодействия с БД. Используется Spring Data Jpa
- model/src/main/java/com/goganesh/gallery/model/service - слой бизнес логики

Модуль **security** обеспечивает аутентификацию и авторизацию. Может быть применена к любому слою для обеспечения безопасности.

- security/src/main/java/com/goganesh/gallery/security/configuration/SecurityConfiguration.java - класс который создает бины данного модуля. Данный класс импортируется в модуль application
- security/src/main/java/com/goganesh/gallery/security/model/UserDetailsImpl.java - реализация класса `UserDetails`
- security/src/main/java/com/goganesh/gallery/security/service/UserDetailsServiceImpl.java - реализация класса `UserDetailsService`

Модуль **web-api** отвечает за предоставление API. Слой отвечает за принятие REST запросов от клиента.

- web-api/src/main/java/com/goganesh/gallery/webapi/configuration/WebApiConfiguration.java - класс который создает бины данного модуля. Данный класс импортируется в модуль applicatio
- web-api/src/main/java/com/goganesh/gallery/webapi/controller - каталог который содержит контроллеры, которые обрабатывают запросы на создание/редактирование объектов.
- web-api/src/main/java/com/goganesh/gallery/webapi/dto - каталог содержит DTO для API

Модуль **web-ui** отвечает отображение web страниц панели администратора и сайта. Состоит из статических HTML файлов и клиентской логики с использованием Java Script. В качестве web-сервера выступает само приложение.

- web-ui/src/main/java/com/goganesh/gallery/webui/configuration/WebUiConfiguration.java - класс который создает бины данного модуля. Данный класс импортируется в модуль applicatio
- web-ui/src/main/java/com/goganesh/gallery/webui/controller - каталог который содержит контроллеры, которые обрабатывают запросы на получения страниц
- web-ui/src/main/resources/templates - каталог содержит шаблоны html страниц для фреймворка Thymleaf
- web-ui/src/main/resources/static - каталог содержит статику проекта css, js, изображения, шрифты и т.д.