



class es.com.disastercode.springangularjs.business

es.com.disastercode.springangularjs.business.entity

+ XxxEntity

+ YyyyyEntity

es.com.disastercode.springangularjs.business.repository

+ XxxRepository

es.com.disastercode.springangularjs.business.service

+ CrudService<T>

+ XxxService

es.com.disastercode.springangularjs.business.service.impl

+ CrudServiceImpl<T>

+ XxxServiceImpl

es.com.disastercode.springangularjs.business.service.criteria

+ XxxSearchCriteria

Capa de entidades.

class es.com.disastercode.springangularjs.business.entity

XxxxxEntity

Entidades de la BD.

YyyyyEntity

Entidades de la BD.

Capa de los repositorios de las entidades.

class es.com.disastercode.springangularjs.repository

«interface» XxxRepository

Interfaces que extienden de JpaRepository<XxxxxEntity, Long>

Capa de los servicios.

class es.com.disastercode.springangularjs.business.service

«interface» CrudService<T>

Interface genérico que define las operaciones CRUD para que el resto de interfaces de esta capa hereden de él y sólo tengan que añadir el resto de métodos personalizados.

«interface» XxxService

class es.com.disastercode.springangularjs.business.service.impl

CrudServiceImpl<T>

Clase genérica heredada por el resto de servicios que provee de los métodos CRUD.

XxxxxServiceImpl

Implementación del servicio del XxxEntity que debe heredar del crud genérico e implementar su servicio.
extends CrudServiceImpl<XxxxxEntity>
implements XxxService

class es.com.disastercode.springangularjs.business.service.criteria

XxxxxSearchCriteria

Clase que contiene los datos del XxxEntity por los que se va a consultar. Debe tener implementado un setter a partir de la request así como un método que retorne los predicados de consulta con un Specification:
public static Specification<XxxxxEntity> search(final XxxSearchCriteria criteria)

es.com.disastercode.springangularjs.business

+ es.com.disastercode.springangularjs.business.entity

+ es.com.disastercode.springangularjs.business.repository

+ es.com.disastercode.springangularjs.business.service

+ es.com.disastercode.springangularjs.business.service.criteria

+ es.com.disastercode.springangularjs.business.service.impl

+ es.com.disastercode.springangularjs.business.entity

+ es.com.disastercode.springangularjs.business.service

+ es.com.disastercode.springangularjs.business.service.criteria

+ es.com.disastercode.springangularjs.business.service.impl

+ es.com.disastercode.springangularjs.repository

class es.com.disastercode.springangularjs.web

es.com.disastercode.springangularjs.web.filter

+ ParseFilter

es.com.disastercode.springangularjs.web.controller

+ CrudController<T>

+ XxxController

es.gobcan.carreteras.migr.web.controller.rest

+ CRUDControllerRest<Entity>

+ XxxControllerRest

es.gobcan.carreteras.migr.web.controller.resource

+ XxxResource

es.gobcan.carreteras.migr.web.controller.report

+ XxxControllerReport

Capa de los controladores: MVC, REST, reports.

class es.com.disastercode.springangularjs.web.filter

ParseFilter

Clase que implementa a la clase Filter. Se utiliza para transformar las url's de peticiones a la aplicación, para facilitar el funcionamiento en distintos entornos (Esto es debido a que en local desplegamos directamente con un tomcat embebido).

class es.com.disastercode.springangularjs.web.controller

CrudController<T>

Clase abstracta que implementa los métodos necesarios para cargar las pantallas del CRUD del Entity utilizando Spring MVC.

XxxxController

Clase que hereda del controlador crud genérico personalizada para el XxxEntity.

class es.com.disastercode.springangularjs.web.controller.rest

CRUDControllerRest<Entity>

Clase abstracta utilizada para implementar los métodos comunes a todos los RestController de cada entidad.

XxxxxControllerRest

Controlador REST FULL de la entidad, que permite realizar las tareas CRUD desde el cliente mediante llamadas asíncronas.

class es.com.disastercode.springangularjs.web.controller.resource

XxxxxResource

Clase que hereda de ResourceSupport extends ResourceSupport implementa Serializable. Se utiliza en los servicios rest para seguir el estándar HATEOAS. Deben implementar un populate y un set para pasar de entity a resource y viceversa.

class es.com.disastercode.springangularjs.web.controller.report

XxxxxControllerReport

Clase que se encarga de la generación de los informes para la entidad xxxxxx.

es.com.disastercode.springangularjs.web

+ es.com.disastercode.springangularjs.web.controller

+ es.com.disastercode.springangularjs.web.filter

+ es.gobcan.carreteras.migr.web.controller.report

+ es.gobcan.carreteras.migr.web.controller.resource

+ es.gobcan.carreteras.migr.web.controller.rest

+ es.com.disastercode.springangularjs.web.controller

+ es.com.disastercode.springangularjs.web.controller.report

+ es.com.disastercode.springangularjs.web.controller.resource

+ es.com.disastercode.springangularjs.web.controller.rest

+ es.com.disastercode.springangularjs.web.filter