

Project: "This is my time"
FINAL

Generado por Doxygen 1.8.18

1 Índice de namespaces	1
1.1 Paquetes	1
2 Índice jerárquico	3
2.1 Jerarquía de la clase	3
3 Índice de clases	5
3.1 Lista de clases	5
4 Documentación de namespaces	7
4.1 Paquetes edu.app.server	7
4.1.1 Descripción detallada	7
4.2 Paquetes edu.app.server.config	7
4.2.1 Descripción detallada	8
4.3 Paquetes edu.app.server.controller	8
4.3.1 Descripción detallada	8
4.4 Paquetes edu.app.server.error	8
4.4.1 Descripción detallada	8
4.5 Paquetes edu.app.server.model	9
4.5.1 Descripción detallada	9
4.6 Paquetes edu.app.server.repository	9
4.6.1 Descripción detallada	10
4.7 Paquetes edu.app.server.service	10
4.7.1 Descripción detallada	10
5 Documentación de las clases	11
5.1 Referencia de la Clase edu.app.server.error.ApiError	11
5.1.1 Descripción detallada	11
5.2 Referencia de la Clase edu.app.server.model.Authority	11
5.2.1 Descripción detallada	12
5.2.2 Documentación del constructor y destructor	12
5.2.2.1 Authority()	12
5.3 Referencia de la Interfaz edu.app.server.repository.AuthorityRepository	12
5.3.1 Descripción detallada	13
5.3.2 Documentación de las funciones miembro	13
5.3.2.1 findByAuthority()	13
5.4 Referencia de la Clase edu.app.server.repository.AuthorityRepositoryTest	13
5.5 Referencia de la Clase edu.app.server.controller.AuthorityRestController	14
5.5.1 Descripción detallada	14
5.5.2 Documentación del constructor y destructor	14
5.5.2.1 AuthorityRestController()	14
5.5.3 Documentación de las funciones miembro	14
5.5.3.1 deleteOne()	14
5.5.3.2 findOne()	15

5.5.3.3 getAll()	15
5.5.3.4 newAuthority()	15
5.6 Referencia de la Clase edu.app.server.controller.AuthorityRestControllerTest	16
5.7 Referencia de la Clase edu.app.server.service.AuthorityService	16
5.7.1 Descripción detallada	16
5.7.2 Documentación del constructor y destructor	16
5.7.2.1 AuthorityService()	16
5.7.3 Documentación de las funciones miembro	17
5.7.3.1 deleteAuthority()	17
5.7.3.2 getAllAuthorities()	17
5.7.3.3 getById()	17
5.7.3.4 getName()	18
5.7.3.5 saveAuthority()	18
5.7.3.6 updateAuthority()	19
5.8 Referencia de la Clase edu.app.server.service.AuthorityServiceTest	19
5.9 Referencia de la Clase edu.app.server.model.AuthorityTest	19
5.10 Referencia de la Clase edu.app.server.config.Constants	20
5.11 Referencia de la Clase edu.app.server.utils.DateTimeTest	20
5.12 Referencia de la Clase edu.app.server.error.ErrorRestController	20
5.12.1 Descripción detallada	21
5.12.2 Documentación de las funciones miembro	21
5.12.2.1 handleConstraintViolation()	21
5.13 Referencia de la Clase edu.app.server.model.GroupTask	21
5.13.1 Descripción detallada	22
5.14 Referencia de la Interfaz edu.app.server.repository.GroupTaskRepository	22
5.14.1 Descripción detallada	22
5.14.2 Documentación de las funciones miembro	22
5.14.2.1 findByName()	22
5.14.2.2 getById()	23
5.15 Referencia de la Clase edu.app.server.repository.GroupTaskRepositoryTest	23
5.16 Referencia de la Clase edu.app.server.controller.GroupTaskRestController	24
5.16.1 Descripción detallada	24
5.16.2 Documentación del constructor y destructor	24
5.16.2.1 GroupTaskRestController()	24
5.16.3 Documentación de las funciones miembro	24
5.16.3.1 deleteOne()	24
5.16.3.2 findAll()	25
5.16.3.3 findOne()	25
5.16.3.4 newGroupTask()	25
5.17 Referencia de la Clase edu.app.server.controller.GroupTaskRestControllerTest	26
5.18 Referencia de la Clase edu.app.server.service.GroupTaskService	26
5.18.1 Descripción detallada	26

5.18.2 Documentación del constructor y destructor	26
5.18.2.1 GroupTaskService()	26
5.18.3 Documentación de las funciones miembro	27
5.18.3.1 deleteGroupTask()	27
5.18.3.2 getAllGroupTask()	27
5.18.3.3 getByld()	27
5.18.3.4 getByName()	28
5.18.3.5 saveGroupTask()	28
5.18.3.6 updateGroupTask()	29
5.19 Referencia de la Clase edu.app.server.service.GroupTaskServiceTest	29
5.20 Referencia de la Clase edu.app.server.model.GroupTaskTest	29
5.21 Referencia de la Clase edu.app.server.config.JWTAuthenticationFilter	29
5.21.1 Documentación de las funciones miembro	30
5.21.1.1 attemptAuthentication()	30
5.22 Referencia de la Clase edu.app.server.config.JWTAuthorizationFilter	30
5.23 Referencia de la Clase edu.app.server.model.NonPeriodicTask	31
5.23.1 Descripción detallada	31
5.24 Referencia de la Interfaz edu.app.server.repository.NonPeriodicTaskRepository	31
5.24.1 Descripción detallada	32
5.24.2 Documentación de las funciones miembro	32
5.24.2.1 findByName()	32
5.24.2.2 getByld()	32
5.25 Referencia de la Clase edu.app.server.repository.NonPeriodicTaskRepositoryTest	33
5.26 Referencia de la Clase edu.app.server.controller.NonPeriodicTaskRestController	33
5.26.1 Descripción detallada	33
5.26.2 Documentación del constructor y destructor	34
5.26.2.1 NonPeriodicTaskRestController()	34
5.26.3 Documentación de las funciones miembro	34
5.26.3.1 deleteOne()	34
5.26.3.2 findAll()	34
5.26.3.3 findOne()	35
5.26.3.4 newNonPeriodicTask()	35
5.27 Referencia de la Clase edu.app.server.controller.NonPeriodicTaskRestControllerTest	35
5.28 Referencia de la Clase edu.app.server.service.NonPeriodicTaskService	36
5.28.1 Descripción detallada	36
5.28.2 Documentación del constructor y destructor	36
5.28.2.1 NonPeriodicTaskService()	36
5.28.3 Documentación de las funciones miembro	36
5.28.3.1 deleteNonPeriodicTask()	37
5.28.3.2 getAllNonPeriodicTask()	37
5.28.3.3 getByld()	37
5.28.3.4 getByName()	38

5.28.3.5 saveNonPeriodicTask()	38
5.28.3.6 updateNonPeriodicTask()	38
5.29 Referencia de la Clase edu.app.server.service.NonPeriodicTaskServiceTest	39
5.30 Referencia de la Clase edu.app.server.model.NonPeriodicTaskTest	39
5.31 Referencia de la Clase edu.app.server.model.Periodicity	39
5.31.1 Descripción detallada	39
5.32 Referencia de la Interfaz edu.app.server.repository.PeriodicityRepository	39
5.32.1 Descripción detallada	40
5.33 Referencia de la Clase edu.app.server.controller.PeriodicityRestController	40
5.33.1 Descripción detallada	40
5.33.2 Documentación del constructor y destructor	40
5.33.2.1 PeriodicityRestController()	40
5.33.3 Documentación de las funciones miembro	41
5.33.3.1 deleteOne()	41
5.33.3.2 findAll()	41
5.33.3.3 findOne()	41
5.33.3.4 newPeriodicity()	42
5.34 Referencia de la Clase edu.app.server.controller.PeriodicityRestControllerTest	42
5.35 Referencia de la Clase edu.app.server.service.PeriodicityService	42
5.35.1 Descripción detallada	43
5.35.2 Documentación del constructor y destructor	43
5.35.2.1 PeriodicityService()	43
5.35.3 Documentación de las funciones miembro	43
5.35.3.1 deletePeriodicity()	43
5.35.3.2 getAllPeriodicity()	44
5.35.3.3 getByName()	44
5.35.3.4 savePeriodicity()	44
5.35.3.5 updatePeriodicity()	45
5.36 Referencia del enum edu.app.server.model.PeriodicityType	45
5.36.1 Descripción detallada	46
5.36.2 Documentación de los datos miembro	46
5.36.2.1 DAY	46
5.36.2.2 MONTH	46
5.36.2.3 WEEK	46
5.36.2.4 YEAR	46
5.37 Referencia de la Clase edu.app.server.model.PeriodicTask	46
5.37.1 Descripción detallada	47
5.38 Referencia de la Interfaz edu.app.server.repository.PeriodicTaskRepository	47
5.38.1 Descripción detallada	47
5.38.2 Documentación de las funciones miembro	48
5.38.2.1 findByName()	48
5.38.2.2 getByName()	49

5.39 Referencia de la Clase edu.app.server.repository.PeriodicTaskRepositoryTest	49
5.40 Referencia de la Clase edu.app.server.controller.PeriodicTaskRestController	50
5.40.1 Descripción detallada	50
5.40.2 Documentación del constructor y destructor	50
5.40.2.1 PeriodicTaskRestController()	50
5.40.3 Documentación de las funciones miembro	50
5.40.3.1 deleteOne()	50
5.40.3.2 findAll()	51
5.40.3.3 findOne()	51
5.40.3.4 newPeriodicTask()	51
5.41 Referencia de la Clase edu.app.server.controller.PeriodicTaskRestControllerTest	52
5.42 Referencia de la Clase edu.app.server.service.PeriodicTaskService	52
5.42.1 Descripción detallada	52
5.42.2 Documentación del constructor y destructor	53
5.42.2.1 PeriodicTaskService()	53
5.42.3 Documentación de las funciones miembro	53
5.42.3.1 deletePeriodicTask()	53
5.42.3.2 getAllPeriodicTask()	53
5.42.3.3 getByld()	54
5.42.3.4 getName()	54
5.42.3.5 savePeriodicTask()	55
5.42.3.6 updatePeriodicTask()	55
5.43 Referencia de la Clase edu.app.server.service.PeriodicTaskServiceTest	56
5.44 Referencia de la Clase edu.app.server.model.PeriodicTaskTest	56
5.45 Referencia de la Clase edu.app.server.model.Resource	56
5.45.1 Descripción detallada	56
5.45.2 Documentación del constructor y destructor	56
5.45.2.1 Resource()	56
5.46 Referencia de la Interfaz edu.app.server.repository.ResourceRepository	57
5.46.1 Descripción detallada	57
5.46.2 Documentación de las funciones miembro	57
5.46.2.1 findByResource()	57
5.47 Referencia de la Clase edu.app.server.repository.ResourceRepositoryTest	58
5.48 Referencia de la Clase edu.app.server.service.ResourceService	58
5.48.1 Descripción detallada	58
5.48.2 Documentación del constructor y destructor	59
5.48.2.1 ResourceService()	59
5.48.3 Documentación de las funciones miembro	60
5.48.3.1 deleteResource()	60
5.48.3.2 getAllResources()	60
5.48.3.3 getByld()	60
5.48.3.4 getName()	61

5.48.3.5 saveResource()	61
5.48.3.6 updateResource()	62
5.49 Referencia de la Clase edu.app.server.service.ResourceServiceTest	62
5.50 Referencia de la Clase edu.app.server.model.ResourceTest	62
5.51 Referencia de la Clase edu.app.server.config.SecurityConfig	63
5.52 Referencia de la Clase edu.app.server.config.SecurityData	63
5.53 Referencia de la Clase edu.app.server.ServerRestApplication	63
5.54 Referencia del enum edu.app.server.model.State	64
5.54.1 Descripción detallada	64
5.54.2 Documentación de los datos miembro	64
5.54.2.1 IN_PROCESS	64
5.54.2.2 PENDING	64
5.54.2.3 SUSPENDED	64
5.54.2.4 TERMINATED	64
5.55 Referencia de la Clase edu.app.server.model.Task	65
5.55.1 Descripción detallada	65
5.55.2 Documentación del constructor y destructor	65
5.55.2.1 Task()	65
5.55.3 Documentación de los datos miembro	66
5.55.3.1 description	66
5.55.3.2 id	66
5.55.3.3 name	66
5.55.3.4 state	66
5.55.3.5 user	66
5.56 Referencia de la Clase edu.app.server.model.TaskTest	66
5.57 Referencia de la Clase edu.app.server.utils.Token	67
5.58 Referencia de la Clase edu.app.server.model.User	67
5.58.1 Descripción detallada	67
5.58.2 Documentación del constructor y destructor	67
5.58.2.1 User()	67
5.59 Referencia de la Clase edu.app.server.config.UserDetailServiceImpl	68
5.60 Referencia de la Interfaz edu.app.server.repository.UserRepository	68
5.60.1 Descripción detallada	68
5.60.2 Documentación de las funciones miembro	69
5.60.2.1 findByUsername()	69
5.61 Referencia de la Clase edu.app.server.repository.UserRepositoryTest	69
5.62 Referencia de la Clase edu.app.server.controller.UserRestController	69
5.62.1 Descripción detallada	70
5.62.2 Documentación del constructor y destructor	70
5.62.2.1 UserRestController()	70
5.62.3 Documentación de las funciones miembro	70
5.62.3.1 addUser()	70

5.62.3.2 deleteOne()	71
5.62.3.3 findAll()	71
5.63 Referencia de la Clase edu.app.server.controller.UserRestControllerTest	71
5.64 Referencia de la Clase edu.app.server.service.UserService	71
5.64.1 Descripción detallada	72
5.64.2 Documentación del constructor y destructor	72
5.64.2.1 UserService()	72
5.64.3 Documentación de las funciones miembro	72
5.64.3.1 deleteUser()	72
5.64.3.2 getAllUsers()	73
5.64.3.3 getByld()	73
5.64.3.4 getByUsername()	74
5.64.3.5 saveUser()	74
5.64.3.6 updateUser()	74
5.65 Referencia de la Clase edu.app.server.service.UserServiceTest	75
5.66 Referencia de la Clase edu.app.server.model.UserTest	75
5.67 Referencia de la Clase edu.app.server.config.ValidationMessage	75
Índice alfabético	77

Capítulo 1

Índice de namespaces

1.1. Paquetes

Aquí van los paquetes con una breve descripción (si está disponible):

edu.app.server	7
edu.app.server.config	7
edu.app.server.controller	8
edu.app.server.error	8
edu.app.server.model	9
edu.app.server.repository	9
edu.app.server.service	10

Capítulo 2

Índice jerárquico

2.1. Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

edu.app.server.error.ApiError	11
edu.app.server.repository.AuthorityRepositoryTest	13
edu.app.server.controller.AuthorityRestController	14
edu.app.server.controller.AuthorityRestControllerTest	16
edu.app.server.service.AuthorityService	16
edu.app.server.service.AuthorityServiceTest	19
edu.app.server.model.AuthorityTest	19
edu.app.server.config.Constants	20
edu.app.server.utils.DateTimeTest	20
edu.app.server.model.GroupTask	21
edu.app.server.repository.GroupTaskRepositoryTest	23
edu.app.server.controller.GroupTaskRestController	24
edu.app.server.controller.GroupTaskRestControllerTest	26
edu.app.server.service.GroupTaskService	26
edu.app.server.service.GroupTaskServiceTest	29
edu.app.server.model.GroupTaskTest	29
edu.app.server.repository.NonPeriodicTaskRepositoryTest	33
edu.app.server.controller.NonPeriodicTaskRestController	33
edu.app.server.controller.NonPeriodicTaskRestControllerTest	35
edu.app.server.service.NonPeriodicTaskService	36
edu.app.server.service.NonPeriodicTaskServiceTest	39
edu.app.server.model.NonPeriodicTaskTest	39
edu.app.server.model.Periodicity	39
edu.app.server.controller.PeriodicityRestController	40
edu.app.server.controller.PeriodicityRestControllerTest	42
edu.app.server.service.PeriodicityService	42
edu.app.server.model.PeriodicityType	45
edu.app.server.repository.PeriodicTaskRepositoryTest	49
edu.app.server.controller.PeriodicTaskRestController	50
edu.app.server.controller.PeriodicTaskRestControllerTest	52
edu.app.server.service.PeriodicTaskService	52
edu.app.server.service.PeriodicTaskServiceTest	56
edu.app.server.model.PeriodicTaskTest	56
edu.app.server.model.Resource	56
edu.app.server.repository.ResourceRepositoryTest	58

edu.app.server.service.ResourceService	58
edu.app.server.service.ResourceServiceTest	62
edu.app.server.model.ResourceTest	62
edu.app.server.config.SecurityData	63
edu.app.server.ServerRestApplication	63
edu.app.server.model.State	64
edu.app.server.model.Task	65
edu.app.server.model.NonPeriodicTask	31
edu.app.server.model.PeriodicTask	46
edu.app.server.model.TaskTest	66
edu.app.server.utils.Token	67
edu.app.server.model.User	67
edu.app.server.repository.UserRepositoryTest	69
edu.app.server.controller.UserRestController	69
edu.app.server.controller.UserRestControllerTest	71
edu.app.server.service.UserService	71
edu.app.server.service.UserServiceTest	75
edu.app.server.model.UserTest	75
edu.app.server.config.ValidationMessage	75
AbstractAuthenticationProcessingFilter	
edu.app.server.config.JWTAuthenticationFilter	29
GrantedAuthority	
edu.app.server.model.Authority	11
JpaRepository	
edu.app.server.repository.AuthorityRepository	12
edu.app.server.repository.GroupTaskRepository	22
edu.app.server.repository.NonPeriodicTaskRepository	31
edu.app.server.repository.PeriodicityRepository	39
edu.app.server.repository.PeriodicTaskRepository	47
edu.app.server.repository.ResourceRepository	57
edu.app.server.repository.UserRepository	68
OncePerRequestFilter	
edu.app.server.config.JWTAuthorizationFilter	30
ResponseEntityExceptionHandler	
edu.app.server.error.ErrorRestController	20
UserDetailsService	
edu.app.server.config.UserDetailServiceImpl	68
WebSecurityConfigurerAdapter	
edu.app.server.config.SecurityConfig	63

Capítulo 3

Índice de clases

3.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

edu.app.server.error.ApiError	11
edu.app.server.model.Authority	11
edu.app.server.repository.AuthorityRepository	12
edu.app.server.repository.AuthorityRepositoryTest	13
edu.app.server.controller.AuthorityRestController	14
edu.app.server.controller.AuthorityRestControllerTest	16
edu.app.server.service.AuthorityService	16
edu.app.server.service.AuthorityServiceTest	19
edu.app.server.model.AuthorityTest	19
edu.app.server.config.Constants	20
edu.app.server.utils.DateTimeTest	20
edu.app.server.error.ErrorRestController	20
edu.app.server.model.GroupTask	21
edu.app.server.repository.GroupTaskRepository	22
edu.app.server.repository.GroupTaskRepositoryTest	23
edu.app.server.controller.GroupTaskRestController	24
edu.app.server.controller.GroupTaskRestControllerTest	26
edu.app.server.service.GroupTaskService	26
edu.app.server.service.GroupTaskServiceTest	29
edu.app.server.model.GroupTaskTest	29
edu.app.server.config.JWTAuthenticationFilter	29
edu.app.server.config.JWTAuthorizationFilter	30
edu.app.server.model.NonPeriodicTask	31
edu.app.server.repository.NonPeriodicTaskRepository	31
edu.app.server.repository.NonPeriodicTaskRepositoryTest	33
edu.app.server.controller.NonPeriodicTaskRestController	33
edu.app.server.controller.NonPeriodicTaskRestControllerTest	35
edu.app.server.service.NonPeriodicTaskService	36
edu.app.server.service.NonPeriodicTaskServiceTest	39
edu.app.server.model.NonPeriodicTaskTest	39
edu.app.server.model.Periodicity	39
edu.app.server.repository.PeriodicityRepository	39
edu.app.server.controller.PeriodicityRestController	40
edu.app.server.controller.PeriodicityRestControllerTest	42
edu.app.server.service.PeriodicityService	42

edu.app.server.model.PeriodicityType	45
edu.app.server.model.PeriodicTask	46
edu.app.server.repository.PeriodicTaskRepository	47
edu.app.server.repository.PeriodicTaskRepositoryTest	49
edu.app.server.controller.PeriodicTaskRestController	50
edu.app.server.controller.PeriodicTaskRestControllerTest	52
edu.app.server.service.PeriodicTaskService	52
edu.app.server.service.PeriodicTaskServiceTest	56
edu.app.server.model.PeriodicTaskTest	56
edu.app.server.model.Resource	56
edu.app.server.repository.ResourceRepository	57
edu.app.server.repository.ResourceRepositoryTest	58
edu.app.server.service.ResourceService	58
edu.app.server.service.ResourceServiceTest	62
edu.app.server.model.ResourceTest	62
edu.app.server.config.SecurityConfig	63
edu.app.server.config.SecurityData	63
edu.app.server.ServerRestApplication	63
edu.app.server.model.State	64
edu.app.server.model.Task	65
edu.app.server.model.TaskTest	66
edu.app.server.utils.Token	67
edu.app.server.model.User	67
edu.app.server.config.UserDetailServiceImpl	68
edu.app.server.repository.UserRepository	68
edu.app.server.repository.UserRepositoryTest	69
edu.app.server.controller.UserRestController	69
edu.app.server.controller.UserRestControllerTest	71
edu.app.server.service.UserService	71
edu.app.server.service.UserServiceTest	75
edu.app.server.model.UserTest	75
edu.app.server.config.ValidationMessage	75

Capítulo 4

Documentación de namespaces

4.1. Paquetes edu.app.server

Paquetes

- package [config](#)
- package [controller](#)
- package [error](#)
- package [model](#)
- package [repository](#)
- package [service](#)

Clases

- class [ServerRestApplication](#)
- class **ServerRestApplicationTests**

4.1.1. Descripción detallada

Este es el paquete principal donde se encuentra el archivo maestro de ejecución del Proyecto Spring.

Autor

Erick David Carpio Hachiri

4.2. Paquetes edu.app.server.config

Clases

- class [Constants](#)
- class [JWTAuthenticationFilter](#)
- class [JWTAuthorizationFilter](#)
- class [SecurityConfig](#)
- class [SecurityData](#)
- class [UserDetailServiceImpl](#)
- class [ValidationMessage](#)

4.2.1. Descripción detallada

Tiene clases relacionadas a la configuración de Spring Framework, así como las configuraciones de seguridad filtros y beans necesarios para la ejecución del aplicativo REST.

Autor

Erick David Carpio Hachiri

4.3. Paquetes edu.app.server.controller

Clases

- class [AuthorityRestController](#)
- class [AuthorityRestControllerTest](#)
- class [GroupTaskRestController](#)
- class [GroupTaskRestControllerTest](#)
- class [NonPeriodicTaskRestController](#)
- class [NonPeriodicTaskRestControllerTest](#)
- class [PeriodicityRestController](#)
- class [PeriodicityRestControllerTest](#)
- class [PeriodicTaskRestController](#)
- class [PeriodicTaskRestControllerTest](#)
- class [UserRestController](#)
- class [UserRestControllerTest](#)

4.3.1. Descripción detallada

Almacena clases relacionadas a los controladores de REST de la aplicación.

Autor

Erick David Carpio Hachiri

4.4. Paquetes edu.app.server.error

Clases

- class [ApiError](#)
- class [ErrorRestController](#)

4.4.1. Descripción detallada

Almacena clases relacionadas con el manejo de errores que se producen en los controladores.

Autor

Erick David Carpio Hachiri

4.5. Paquetes edu.app.server.model

Clases

- class [Authority](#)
- class [AuthorityTest](#)
- class [GroupTask](#)
- class [GroupTaskTest](#)
- class [NonPeriodicTask](#)
- class [NonPeriodicTaskTest](#)
- class [Periodicity](#)
- enum [PeriodicityType](#)
- class [PeriodicTask](#)
- class [PeriodicTaskTest](#)
- class [Resource](#)
- class [ResourceTest](#)
- enum [State](#)
- class [Task](#)
- class [TaskTest](#)
- class [User](#)
- class [UserTest](#)

4.5.1. Descripción detallada

Almacena clases relacionadas a los modelos de datos usados en el aplicativo y que estan relacionados con la base de datos.

Autor

Erick David Carpio Hachiri

4.6. Paquetes edu.app.server.repository

Clases

- interface [AuthorityRepository](#)
- class [AuthorityRepositoryTest](#)
- interface [GroupTaskRepository](#)
- class [GroupTaskRepositoryTest](#)
- interface [NonPeriodicTaskRepository](#)
- class [NonPeriodicTaskRepositoryTest](#)
- interface [PeriodicityRepository](#)
- interface [PeriodicTaskRepository](#)
- class [PeriodicTaskRepositoryTest](#)
- interface [ResourceRepository](#)
- class [ResourceRepositoryTest](#)
- interface [UserRepository](#)
- class [UserRepositoryTest](#)

4.6.1. Descripción detallada

Almacena clases relacionadas a las operaciones con la base de datos usando Spring Data JPA.

Autor

Erick David Carpio Hachiri

4.7. Paquetes edu.app.server.service

Clases

- class [AuthorityService](#)
- class [AuthorityServiceTest](#)
- class [GroupTaskService](#)
- class [GroupTaskServiceTest](#)
- class [NonPeriodicTaskService](#)
- class [NonPeriodicTaskServiceTest](#)
- class [PeriodicityService](#)
- class [PeriodicTaskService](#)
- class [PeriodicTaskServiceTest](#)
- class [ResourceService](#)
- class [ResourceServiceTest](#)
- class [UserService](#)
- class [UserServiceTest](#)

4.7.1. Descripción detallada

Almacena clases relacionadas a la aplicación de la logica de negocios, formateo de datos o tipo de operación requerida (transacciones).

Autor

Erick David Carpio Hachiri

Capítulo 5

Documentación de las clases

5.1. Referencia de la Clase edu.app.server.error.ApiError

Métodos públicos

- **ApiError** (HttpStatus status, String message, List< String > errors)
- **ApiError** (HttpStatus status, String message, String error)

5.1.1. Descripción detallada

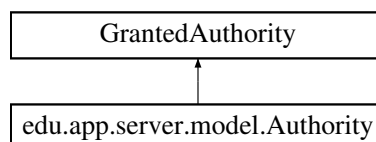
Sirve para enviar respuestas de los errores que se producen al realizar un petición del usuario.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/error/ApiError.java

5.2. Referencia de la Clase edu.app.server.model.Authority

Diagrama de herencias de edu.app.server.model.Authority



Métodos públicos

- **Authority** (String authority, Boolean isEnabled)

5.2.1. Descripción detallada

La clase [Authority](#) sirve para dar roles a los usuarios y a que recursos pueden acceder.

5.2.2. Documentación del constructor y destructor

5.2.2.1. Authority()

```
edu.app.server.model.Authority.Authority (
    String authority,
    Boolean isEnabled )
```

Este es el constructor de la clase con los atributos inicializables.

Parámetros

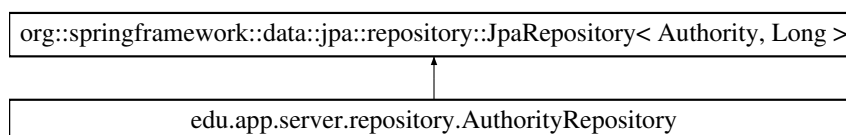
<i>authority</i>	Es el nombre de la autoridad.
<i>isEnabled</i>	Es el estado de la autoridad.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/model/Authority.java

5.3. Referencia de la Interfaz edu.app.server.repository.AuthorityRepository

Diagrama de herencias de edu.app.server.repository.AuthorityRepository



Métodos públicos

- Optional< [Authority](#) > [findByAuthority](#) (String authority)

5.3.1. Descripción detallada

Es el repositorio que se encarga de realizar las operaciones con la base de datos mediante JPA, además trae más métodos con extender la interface JpaRepository.

Autor

Erick David Carpio Hachiri

Ver también

JpaRepository

Authority

5.3.2. Documentación de las funciones miembro

5.3.2.1. findByAuthority()

```
Optional<Authority> edu.app.server.repository.AuthorityRepository.findByAuthority (
    String authority )
```

Busca el por el nombre de la autoridad.

Parámetros

<i>authority</i>	Es el nombre de autoridad.
------------------	----------------------------

Devuelve

Retorna un Optional de tipo Authority.

Ver también

Optional

La documentación para este interfaz fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/repository/AuthorityRepository.java

5.4. Referencia de la Clase edu.app.server.repository.AuthorityRepositoryTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/repository/AuthorityRepositoryTest.java

5.5. Referencia de la Clase

edu.app.server.controller.AuthorityRestController

Métodos públicos

- [AuthorityRestController](#) ([AuthorityService](#) authorityService)
- `ResponseEntity< List< Authority > > getAll ()`
- `ResponseEntity< Authority > newAuthority (@Valid @RequestBody Authority authority)`
- `ResponseEntity< Authority > findOne (@PathVariable @Min(1) Long id)`
- `void deleteOne (@PathVariable @Min(1) Long id)`

5.5.1. Descripción detallada

Es el controlador Rest de Authority y admite conexiones con cualquier IP.

Autor

Erick David Carpio Hachiri

Ver también

[AuthorityService](#)

5.5.2. Documentación del constructor y destructor

5.5.2.1. AuthorityRestController()

```
edu.app.server.controller.AuthorityRestController.AuthorityRestController (
    AuthorityService authorityService )
```

Constructor que inyecta las dependencias de Spring container.

Parámetros

<i>authorityService</i>	Es el servicio que sera inyectado por Spring.
-------------------------	---

5.5.3. Documentación de las funciones miembro

5.5.3.1. deleteOne()

```
void edu.app.server.controller.AuthorityRestController.deleteOne (
    @PathVariable @Min(1) Long id )
```


Es el servicio Rest que elimina una autoridad mediante el metodo DELETE.

Parámetros

<i>id</i>	Es el identificador de la autoridad.
-----------	--------------------------------------

5.5.3.2. findOne()

```
ResponseEntity<Authority> edu.app.server.controller.AuthorityRestController.findOne (
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que busca una autoridad por id su mediante el metodo POST.

Parámetros

<i>id</i>	Es el identificador con el que se buscara la autoridad.
-----------	---

Devuelve

Es la respuesta que dara el servicio.

5.5.3.3. getAll()

```
ResponseEntity<List<Authority> > edu.app.server.controller.AuthorityRestController.getAll ( )
```

Es el servicio Rest que devuelve todas las autoridades mediante el metodo GET.

Devuelve

Es la respuesta que dara el servicio.

5.5.3.4. newAuthority()

```
ResponseEntity<Authority> edu.app.server.controller.AuthorityRestController.newAuthority (
    @Valid @RequestBody Authority authority )
```

Es el servicio Rest que crea una nueva autoridad mediante el metodo PUT.

Parámetros

<i>authority</i>	Es la autoridad que se guardara.
------------------	----------------------------------

Devuelve

Es la respuesta que dara el servicio.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/app/server/controller/AuthorityRestController.java`

5.6. Referencia de la Clase `edu.app.server.controller.AuthorityRestControllerTest`

Métodos públicos

- `void whenGetRequestAllAuthorities ()` throws Exception
- `void whenPostRequestGetAuthority ()` throws Exception

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/test/java/edu/app/server/controller/AuthorityRestControllerTest.java`

5.7. Referencia de la Clase `edu.app.server.service.AuthorityService`

Métodos públicos

- `AuthorityService (AuthorityRepository authorityRepository)`
- `Authority getByld (Long id)`
- `List< Authority > getAllAuthorities ()`
- `Authority saveAuthority (Authority authority)`
- `Authority updateAuthority (Authority authority)`
- `String deleteAuthority (Authority authority)`
- `Authority getName (String name)`

5.7.1. Descripción detallada

Es el modulo de servicio que implementa las logicas de negocio o transformaciones a los datos de Authority.

Autor

Erick David Carpio Hachiri

Ver también

Authority

5.7.2. Documentación del constructor y destructor

5.7.2.1. `AuthorityService()`

```
edu.app.server.service.AuthorityService.AuthorityService (
    AuthorityRepository authorityRepository )
```

Contructor que inyecta las dependencias de Spring container.

Parámetros

<code>authorityRepository</code>	Es el repositorio que sera inyectado por Spring.
----------------------------------	--

5.7.3. Documentación de las funciones miembro

5.7.3.1. deleteAuthority()

```
String edu.app.server.service.AuthorityService.deleteAuthority (
    Authority authority )
```

Elimina la autoridad, mediante una transacción de escritura.

Parámetros

<code>authority</code>	Es la autoridad a eliminar.
------------------------	-----------------------------

Devuelve

Es una cadena con la confirmación.

Ver también

AuthorityRepository

5.7.3.2. getAllAuthorities()

```
List<Authority> edu.app.server.service.AuthorityService.getAllAuthorities ( )
```

Obtiene todas las autoridades, mediante una transacción de solo lectura.

Devuelve

Son todas las autoridades realizadas por el repositorio.

Ver también

AuthorityRepository

5.7.3.3. getById()

```
Authority edu.app.server.service.AuthorityService.getById (
    Long id )
```

Obtiene la autoridad mediante su id, mediante una transacción de solo lectura.

Parámetros

<i>id</i>	Es el id del Authority.
-----------	-------------------------

Devuelve

Retorna el valor existe o no existe que es retornado por el repositorio.

Ver también

AuthorityRepository

5.7.3.4. `getByName()`

```
Authority edu.app.server.service.AuthorityService.getByName (
    String name )
```

Realiza la búsqueda por nombre, mediante una transacción de solo lectura.

Parámetros

<i>name</i>	Es el nombre de la autoridad.
-------------	-------------------------------

Devuelve

Es la autoridad que se busco.

Ver también

AuthorityRepository

5.7.3.5. `saveAuthority()`

```
Authority edu.app.server.service.AuthorityService.saveAuthority (
    Authority authority )
```

Guarda la autoridad, mediante una transacción de escritura.

Parámetros

<i>authority</i>	Es la autoridad a guardar.
------------------	----------------------------

Devuelve

Es la autoridad con el id asignado en la base de datos.

Ver también

AuthorityRepository

5.7.3.6. updateAuthority()

```
Authority edu.app.server.service.AuthorityService.updateAuthority (
    Authority authority )
```

Actualiza la autoridad, mediante una transacción de escritura.

Parámetros

<i>authority</i>	Es la autoridad a actualizar.
------------------	-------------------------------

Devuelve

Es la autoridad con el los valores actualizados en la base de datos.

Ver también

AuthorityRepository

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/service/AuthorityService.java

5.8. Referencia de la Clase edu.app.server.service.AuthorityServiceTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/service/AuthorityServiceTest.java

5.9. Referencia de la Clase edu.app.server.model.AuthorityTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/model/AuthorityTest.java

5.10. Referencia de la Clase edu.app.server.config.Constants

Atributos públicos estáticos

- static final String **LOGIN_URL** = "/login"
- static final String **HEADER_AUTHORIZACION_KEY** = "Authorization"
- static final String **TOKEN_BEARER_PREFIX** = "Bearer "
- static final String **AUTHORITIES_KEY** = "CLAIM_TOKEN"
- static final String **ISSUER_INFO** = "https://www.autentia.com/"
- static final String **SUPER_SECRET_KEY** = "1234"
- static final long **TOKEN_EXPIRATION_TIME** = 864_000_000

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/config/Constants.java

5.11. Referencia de la Clase edu.app.server.utils.DateTimeTest

Métodos públicos

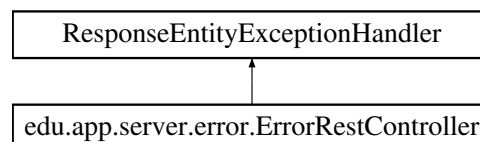
- void **dairyTest** (long day, long month, long year)

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/utils/DateTimeTest.java

5.12. Referencia de la Clase edu.app.server.error.ErrorRestController

Diagrama de herencias de edu.app.server.error.ErrorRestController



Métodos públicos

- `ResponseEntity< Object >` **handleConstraintViolation** (ConstraintViolationException ex, WebRequest request)
- `ResponseEntity< Object >` **handleMethodArgumentTypeMismatch** (MethodArgumentTypeMismatchException ex, WebRequest request)

Métodos protegidos

- `ResponseEntity< Object > handleMethodArgumentNotValid` (`MethodArgumentNotValidException ex`, `HttpHeaders headers`, `HttpStatus status`, `WebRequest request`)
- `ResponseEntity< Object > handleMissingServletRequestParameter` (`MissingServletRequestParameterException ex`, `HttpHeaders headers`, `HttpStatus status`, `WebRequest request`)

5.12.1. Descripción detallada

Es el controlador encargado de recopilar los errores en formato [ApiError](#) y enviarlos en respuesta.

Ver también

[ApiError](#)

5.12.2. Documentación de las funciones miembro

5.12.2.1. handleConstraintViolation()

```
ResponseEntity<Object> edu.app.server.error.ErrorRestController.handleConstraintViolation (
    ConstraintViolationException ex,
    WebRequest request )
```

Se encarga de transformar los errores de validación y estructurarlos y enviarlos al usuario en caso de que llene los datos mal.

Parámetros

<i>ex</i>	Excepción que es lanzada por las validaciones incorrectas.
<i>request</i>	Es el la respuesta enviada por el cliente.

Devuelve

Es un response entity con el modelo de [ApiError](#).

Ver también

[ApiError](#)

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/app/server/error/ErrorRestController.java`

5.13. Referencia de la Clase edu.app.server.model.GroupTask

Métodos públicos

- `GroupTask` (`String name`, [User](#) `user`)

5.13.1. Descripción detallada

Es la clase que define los grupos de las tareas y las asociaciones con los usuarios.

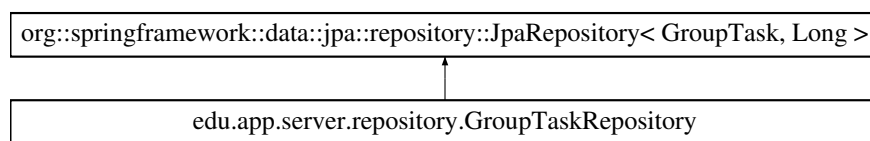
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/app/server/model/GroupTask.java`

5.14. Referencia de la Interfaz

`edu.app.server.repository.GroupTaskRepository`

Diagrama de herencias de `edu.app.server.repository.GroupTaskRepository`



Métodos públicos

- `Optional< GroupTask > getById (Long id)`
- `Optional< GroupTask > findByName (String name)`

5.14.1. Descripción detallada

Es el repositorio de `GroupTask` el cual se encarga de realizar las operaciones de la base de datos. Hace uso de `JpaRepository` para poder extender ciertos metodos ya programados.

Ver también

`JpaRepository`
`GroupTask`

Autor

Erick David Carpio Hachiri

5.14.2. Documentación de las funciones miembro

5.14.2.1. `findByName()`

```
Optional<GroupTask> edu.app.server.repository.GroupTaskRepository.findByName (
    String name )
```

Busca un `GroupTask` por su nombre.

Parámetros

<i>name</i>	Es el nombre del GroupTask que se desea buscar.
-------------	---

Devuelve

Es un valor opcional puede o no puede estar.(null or notnull)

Ver también

Optional

5.14.2.2. getById()

```
Optional<GroupTask> edu.app.server.repository.GroupTaskRepository.getById (
    Long id )
```

Busca un GroupTask por su id.

Parámetros

<i>id</i>	Es el id del registro del cual queremos los datos.
-----------	--

Devuelve

Es un valor opcional puede o no puede estar.(null or notnull)

Ver también

Optional

La documentación para este interfaz fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/repository/GroupTaskRepository.java

5.15. Referencia de la Clase edu.app.server.repository.GroupTaskRepositoryTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/repository/GroupTaskRepositoryTest.java

5.16. Referencia de la Clase

edu.app.server.controller.GroupTaskRestController

Métodos públicos

- [GroupTaskRestController](#) ([GroupTaskService](#) groupTaskService)
- ResponseEntity< List< [GroupTask](#) > > [findAll](#) ()
- ResponseEntity< [GroupTask](#) > [newGroupTask](#) (@Valid @RequestBody [GroupTask](#) groupTask)
- ResponseEntity< [GroupTask](#) > [findOne](#) (@PathVariable @Min(1) Long id)
- void [deleteOne](#) (@PathVariable @Min(1) Long id)

5.16.1. Descripción detallada

Es el controlador Rest de GroupTask y admite conexiones con cualquier IP.

Autor

Erick David Carpio Hachiri

Ver también

[GroupTaskService](#)

5.16.2. Documentación del constructor y destructor

5.16.2.1. GroupTaskRestController()

```
edu.app.server.controller.GroupTaskRestController.GroupTaskRestController (
    GroupTaskService groupTaskService )
```

Constructor que inyecta las dependencias de Spring container.

Parámetros

<i>groupTaskService</i>	Es el servicio que sera inyectado por Spring.
-------------------------	---

5.16.3. Documentación de las funciones miembro

5.16.3.1. deleteOne()

```
void edu.app.server.controller.GroupTaskRestController.deleteOne (
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que elimina un groupTask por su mediante el metodo DELETE.

Parámetros

<i>id</i>	Es el identificador del GroupTask.
-----------	------------------------------------

5.16.3.2. findAll()

```
ResponseEntity<List<GroupTask> > edu.app.server.controller.GroupTaskRestController.findAll (
)
```

Es el servicio Rest que devuelve todos los groupTask mediante el metodo GET.

Devuelve

Es la respuesta que dara el servicio.

5.16.3.3. findOne()

```
ResponseEntity<GroupTask> edu.app.server.controller.GroupTaskRestController.findOne (
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que busca un groupTask por id su mediante el metodo POST.

Parámetros

<i>id</i>	Es el identificador del groupTask.
-----------	------------------------------------

Devuelve

Es la respuesta que dara el servicio.

5.16.3.4. newGroupTask()

```
ResponseEntity<GroupTask> edu.app.server.controller.GroupTaskRestController.newGroupTask (
    @Valid @RequestBody GroupTask groupTask )
```

Es el servicio Rest que crea una nueva groupTask mediante el metodo PUT.

Parámetros

<i>groupTask</i>	Es el groupTask que se guardara.
------------------	----------------------------------

Devuelve

Es la respuesta que dara el servicio.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/app/server/controller/GroupTaskRestController.java`

5.17. Referencia de la Clase `edu.app.server.controller.GroupTaskRestControllerTest`

Métodos públicos

- `void whenGetRequestAllGroupTasks ()` throws Exception
- `void whenPostRequestGetGroupTask ()` throws Exception

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/test/java/edu/app/server/controller/GroupTaskRestControllerTest.java`

5.18. Referencia de la Clase `edu.app.server.service.GroupTaskService`

Métodos públicos

- `GroupTaskService (GroupTaskRepository groupTaskRepository)`
- `GroupTask getById (Long id)`
- `List< GroupTask > getAllGroupTask ()`
- `GroupTask saveGroupTask (GroupTask periodicTask)`
- `GroupTask updateGroupTask (GroupTask periodicTask)`
- `String deleteGroupTask (GroupTask periodicTask)`
- `GroupTask getByName (String name)`

5.18.1. Descripción detallada

Es el modulo de servicio que implementa las logicas de negocio o transformaciones para los datos del GroupTask.

Autor

Erick David Carpio Hachiri

Ver también

[edu.app.server.model.Authority](#)

5.18.2. Documentación del constructor y destructor

5.18.2.1. `GroupTaskService()`

```
edu.app.server.service.GroupTaskService.GroupTaskService (
    GroupTaskRepository groupTaskRepository )
```

Contructor que inyecta las dependencias de Spring container.

Parámetros

<code>groupTaskRepository</code>	Es el repositorio del GroupTask
----------------------------------	---------------------------------

Ver también

GroupTaskRepository

5.18.3. Documentación de las funciones miembro

5.18.3.1. deleteGroupTask()

```
String edu.app.server.service.GroupTaskService.deleteGroupTask (
    GroupTask periodicTask )
```

Elimina la groupTask, mediante una transacción de escritura.

Parámetros

<code>periodicTask</code>	Es la groupTask que se desea eliminar.
---------------------------	--

Devuelve

Es la cadena que confirma si la operación tuvo éxito o no.

5.18.3.2. getAllGroupTask()

```
List<GroupTask> edu.app.server.service.GroupTaskService.getAllGroupTask ( )
```

Obtiene todos los groupTasks, mediante una transacción de solo lectura.

Devuelve

Es una lista con todas las grouptask.

5.18.3.3. getById()

```
GroupTask edu.app.server.service.GroupTaskService.getById (
    Long id )
```

Obtiene una GroupTask por su identificador, mediante una transacción de solo lectura.

Parámetros

<i>id</i>	Es el indentificador del GroupTask
-----------	------------------------------------

Devuelve

Es el groupTask si es que se encuentra y sino retorna un valor nulo.

Ver también

Optional

5.18.3.4. `getByName()`

```
GroupTask edu.app.server.service.GroupTaskService.getByName (
    String name )
```

Obtiene la Grouptask por su nombre, mediante una transaccipon de lectura.

Parámetros

<i>name</i>	Es el nombre de la groupTask que se desea buscar.
-------------	---

Devuelve

Es la grouptask que se busca o un nulo si no existe.

5.18.3.5. `saveGroupTask()`

```
GroupTask edu.app.server.service.GroupTaskService.saveGroupTask (
    GroupTask periodicTask )
```

Guarda la groupTask, mediante una transacción de escritura.

Parámetros

<i>periodicTask</i>	Es la groupTask que se desea guardar.
---------------------	---------------------------------------

Devuelve

Es la grouptask que se guardo en la BD con su id generada.

5.18.3.6. updateGroupTask()

```
GroupTask edu.app.server.service.GroupTaskService.updateGroupTask (
    GroupTask periodicTask )
```

Actualiza la groupTask, mediante una transacción de escritura.

Parámetros

<i>periodicTask</i>	Es la groupTask que se desea actualizar.
---------------------	--

Devuelve

Es la grouptask que se guardo en la BD o un nulo si la operación no se concluye con éxito.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/service/GroupTaskService.java

5.19. Referencia de la Clase edu.app.server.service.GroupTaskServiceTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/service/GroupTaskServiceTest.java

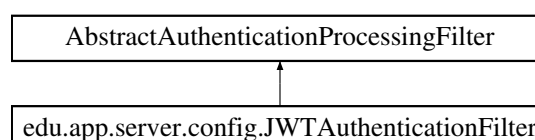
5.20. Referencia de la Clase edu.app.server.model.GroupTaskTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/model/GroupTaskTest.java

5.21. Referencia de la Clase edu.app.server.config.JWTAuthenticationFilter

Diagrama de herencias de edu.app.server.config.JWTAuthenticationFilter



Métodos públicos

- **JWTAuthenticationFilter** (String url, AuthenticationManager authenticationManager, [UserService](#) userService, BCryptPasswordEncoder bCryptPasswordEncoder)
- Authentication [attemptAuthentication](#) (HttpServletRequest request, HttpServletResponse response) throws AuthenticationException

Métodos protegidos

- void **successfulAuthentication** (HttpServletRequest request, HttpServletResponse response, FilterChain chain, Authentication auth) throws IOException, ServletException

5.21.1. Documentación de las funciones miembro

5.21.1.1. attemptAuthentication()

```
Authentication edu.app.server.config.JWTAuthenticationFilter.attemptAuthentication (
    HttpServletRequest request,
    HttpServletResponse response ) throws AuthenticationException
```

Podría restringirse la entrada mediante errores y comparando los recursos que tiene el usuario dentro de sus authorities y dentro de los recursos.

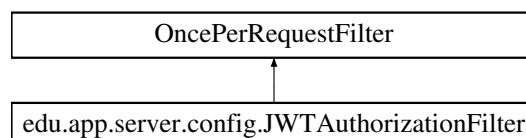
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/config/JWTAuthenticationFilter.java

5.22. Referencia de la Clase

edu.app.server.config.JWTAuthorizationFilter

Diagrama de herencias de edu.app.server.config.JWTAuthorizationFilter



Métodos públicos

- **JWTAuthorizationFilter** ([UserService](#) userService)

Métodos protegidos

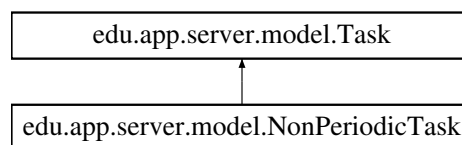
- void **doFilterInternal** (HttpServletRequest req, HttpServletResponse res, FilterChain chain) throws IOException, ServletException

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/config/JWTAuthorizationFilter.java

5.23. Referencia de la Clase edu.app.server.model.NonPeriodicTask

Diagrama de herencias de edu.app.server.model.NonPeriodicTask



Métodos públicos

- **NonPeriodicTask** (String [name](#), String [description](#), [State state](#), [User user](#), LocalDate initDate, LocalTime initTime)

Otros miembros heredados

5.23.1. Descripción detallada

Esta clase extiende la clase [Task](#). Define la tarea que no tiene un periodicidad y se llevara a cabo en un solo día.

Ver también

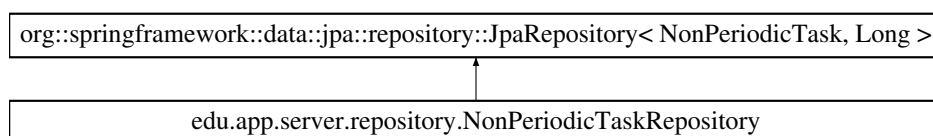
[Task](#)

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/model/NonPeriodicTask.java

5.24. Referencia de la Interfaz edu.app.server.repository.NonPeriodicTaskRepository

Diagrama de herencias de edu.app.server.repository.NonPeriodicTaskRepository



Métodos públicos

- Optional< [NonPeriodicTask](#) > [getById](#) (Long id)
- Optional< [NonPeriodicTask](#) > [findByName](#) (String name)

5.24.1. Descripción detallada

Es el repositorio de NonPeriodicTask el cual hace uso de JpaRepository para importar varios metodos de intercambio con la base de datos.

Autor

Erick David Carpio Hachiri

Ver también

JpaRepository
NonPeriodicTask

5.24.2. Documentación de las funciones miembro

5.24.2.1. findByName()

```
Optional<NonPeriodicTask> edu.app.server.repository.NonPeriodicTaskRepository.findByName (
    String name )
```

Busca la NonPeriodicTask por su nombre.

Parámetros

<i>name</i>	Es el nombre de la NonPeriodicTask.
-------------	-------------------------------------

Devuelve

Retorna un valor opcional que puede estar o no. (null or not null)

Ver también

Optional

5.24.2.2. getById()

```
Optional<NonPeriodicTask> edu.app.server.repository.NonPeriodicTaskRepository.getById (
    Long id )
```

Busca la NonPeriodicTask por su id.

Parámetros

<i>id</i>	Es el id del registro que se desea buscar.
-----------	--

Devuelve

Retorna un valor opcional que puede estar o no. (null or not null)

Ver también

Optional

La documentación para este interfaz fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/repository/NonPeriodicTaskRepository.java

5.25. Referencia de la Clase edu.app.server.repository.NonPeriodicTaskRepositoryTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/repository/NonPeriodicTaskRepositoryTest.java

5.26. Referencia de la Clase edu.app.server.controller.NonPeriodicTaskRestController

Métodos públicos

- [NonPeriodicTaskRestController](#) ([NonPeriodicTaskService](#) nonPeriodicTaskService)
- ResponseEntity< List< [NonPeriodicTask](#) > > [findAll](#) ()
- ResponseEntity< [NonPeriodicTask](#) > [newNonPeriodicTask](#) (@Valid @RequestBody [NonPeriodicTask](#) non←
PeriodicTask)
- ResponseEntity< [NonPeriodicTask](#) > [findOne](#) (@PathVariable @Min(1) Long id)
- void [deleteOne](#) (@PathVariable @Min(1) Long id)

5.26.1. Descripción detallada

Es el controlador Rest de NonPeriodicTask y admite conexiones con cualquier IP.

Autor

Erick David Carpio Hachiri

Ver también

NonPeriodicTaskService

5.26.2. Documentación del constructor y destructor

5.26.2.1. NonPeriodicTaskRestController()

```
edu.app.server.controller.NonPeriodicTaskRestController.NonPeriodicTaskRestController (
    NonPeriodicTaskService nonPeriodicTaskService )
```

Constructor que inyecta las dependencias de Spring container.

Parámetros

<i>nonPeriodicTaskService</i>	Es el servicio que sera inyectado por Spring.
-------------------------------	---

5.26.3. Documentación de las funciones miembro

5.26.3.1. deleteOne()

```
void edu.app.server.controller.NonPeriodicTaskRestController.deleteOne (
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que elimina una nonPeriodicTask por su mediante el metodo DELETE.

Parámetros

<i>id</i>	Es el identificador del nonPeriodicTask.
-----------	--

5.26.3.2. findAll()

```
ResponseEntity<List<NonPeriodicTask> > edu.app.server.controller.NonPeriodicTaskRestController.↵
findAll ( )
```

Es el servicio Rest que devuelve todos las nonPeriodicTask mediante el metodo GET.

Devuelve

Es la respuesta que dara el servicio.

5.26.3.3. findOne()

```
ResponseEntity<NonPeriodicTask> edu.app.server.controller.NonPeriodicTaskRestController.findOne (
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que busca una nonPeriodicTask por id su mediante el metodo POST.

Parámetros

<i>id</i>	Es el identificador del nonPeriodicTask.
-----------	--

Devuelve

Es la respuesta que dara el servicio.

5.26.3.4. newNonPeriodicTask()

```
ResponseEntity<NonPeriodicTask> edu.app.server.controller.NonPeriodicTaskRestController.newNonPeriodicTask (
    @Valid @RequestBody NonPeriodicTask nonPeriodicTask )
```

Es el servicio Rest que crea una nueva nonPeriodicTask mediante el metodo PUT.

Parámetros

<i>nonPeriodicTask</i>	Es el nonPeriodicTask que se guardara.
------------------------	--

Devuelve

Es la respuesta que dara el servicio.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/controller/NonPeriodicTaskRestController.java

5.27. Referencia de la Clase

edu.app.server.controller.NonPeriodicTaskRestControllerTest

Métodos públicos

- void **whenGetRequestAllNonPeriodicTasks** () throws Exception
- void **whenPostRequestGetNonPeriodicTask** () throws Exception

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/controller/NonPeriodicTaskRestControllerTest.java

5.28. Referencia de la Clase

edu.app.server.service.NonPeriodicTaskService

Métodos públicos

- [NonPeriodicTaskService](#) ([NonPeriodicTaskRepository](#) nonPeriodicTaskRepository)
- [NonPeriodicTask](#) [getById](#) (Long id)
- List< [NonPeriodicTask](#) > [getAllNonPeriodicTask](#) ()
- [NonPeriodicTask](#) [saveNonPeriodicTask](#) ([NonPeriodicTask](#) nonPeriodicTask)
- [NonPeriodicTask](#) [updateNonPeriodicTask](#) ([NonPeriodicTask](#) nonPeriodicTask)
- String [deleteNonPeriodicTask](#) ([NonPeriodicTask](#) nonPeriodicTask)
- [NonPeriodicTask](#) [getByName](#) (String name)

5.28.1. Descripción detallada

Es el modulo del servicio que implementa las reglas de negocio o transformaciones a los datos de NonPeriodicTask.

Autor

Erick David Carpio Hachiri

Ver también

[NonPeriodicTask](#)

5.28.2. Documentación del constructor y destructor

5.28.2.1. NonPeriodicTaskService()

```
edu.app.server.service.NonPeriodicTaskService.NonPeriodicTaskService (
    NonPeriodicTaskRepository nonPeriodicTaskRepository )
```

Contructor que inyecta las dependencias de Spring container.

Parámetros

<i>nonPeriodicTaskRepository</i>	Es el repositorio de NonPeriodicTask.
----------------------------------	---------------------------------------

5.28.3. Documentación de las funciones miembro

5.28.3.1. deleteNonPeriodicTask()

```
String edu.app.server.service.NonPeriodicTaskService.deleteNonPeriodicTask (
    NonPeriodicTask nonPeriodicTask )
```

Elimina la NonPeriodicTask, mediante una transacción de escritura.

Parámetros

<i>nonPeriodicTask</i>	Es la nonPeriodicTask que se desea eliminar.
------------------------	--

Devuelve

Retorna una cadena que confirma que se completo la operación.

5.28.3.2. getAllNonPeriodicTask()

```
List<NonPeriodicTask> edu.app.server.service.NonPeriodicTaskService.getAllNonPeriodicTask ( )
```

Obtiene todas las nonPeriodicTask, mediante una transacción de lectura.

Devuelve

Es una lista con todas las NonPeriodicTasks.

5.28.3.3. getById()

```
NonPeriodicTask edu.app.server.service.NonPeriodicTaskService.getById (
    Long id )
```

Obtiene la nonPeriodicTask por identificador, mediante una transacción de lectura.

Parámetros

<i>id</i>	Es el identificador de la nonPeriodicTask que se desea obtener.
-----------	---

Devuelve

Es la nonPeriodicTask que se desea obtener o un valor nulo si no se encuentra.

5.28.3.4. `getByName()`

```
NonPeriodicTask edu.app.server.service.NonPeriodicTaskService.getByName (
    String name )
```

Obtiene una `NonPeriodicTask` por su nombre, mediante una transacción de escritura.

Parámetros

<i>name</i>	Es nombre de la <code>NonPeriodicTask</code>
-------------	--

Devuelve

Es la `nonPeriodicTask` que esta en la BD o un valor nulo si no se completo la operación.

5.28.3.5. `saveNonPeriodicTask()`

```
NonPeriodicTask edu.app.server.service.NonPeriodicTaskService.saveNonPeriodicTask (
    NonPeriodicTask nonPeriodicTask )
```

Guarda la `NonPeriodicTask`, mediante una transacción de escritura.

Parámetros

<i>nonPeriodicTask</i>	Es la <code>nonPeriodicTask</code> que se desea guardar.
------------------------	--

Devuelve

Es la `nonPeriodicTask` que esta en la BD y con el id actualizado.

5.28.3.6. `updateNonPeriodicTask()`

```
NonPeriodicTask edu.app.server.service.NonPeriodicTaskService.updateNonPeriodicTask (
    NonPeriodicTask nonPeriodicTask )
```

Actualiza la `NonPeriodicTask`, mediante una transacción de escritura.

Parámetros

<i>nonPeriodicTask</i>	Es la <code>nonPeriodicTask</code> que se desea actualizar.
------------------------	---

Devuelve

Es la `nonPeriodicTask` que esta en la BD o un valor nulo si no se completo la operación.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/service/NonPeriodicTaskService.java

5.29. Referencia de la Clase edu.app.server.service.NonPeriodicTaskServiceTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/service/NonPeriodicTaskServiceTest.java

5.30. Referencia de la Clase edu.app.server.model.NonPeriodicTaskTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/model/NonPeriodicTaskTest.java

5.31. Referencia de la Clase edu.app.server.model.Periodicity

Métodos públicos

- **Periodicity** (LocalDate initDate, LocalTime initTime, [PeriodicityType](#) periodicityType, Integer quantityOf↔
PeriodicityType)

5.31.1. Descripción detallada

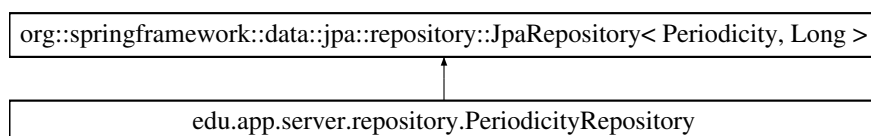
Define la periodicidad de las tareas.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/model/Periodicity.java

5.32. Referencia de la Interfaz edu.app.server.repository.PeriodicityRepository

Diagrama de herencias de edu.app.server.repository.PeriodicityRepository



5.32.1. Descripción detallada

Es el repositorio de Periodicity el cual realiza las operaciones implementadas por JpaRepository.

Autor

Erick David Carpio Hachiri

Ver también

JpaRepository

Periodicity

La documentación para este interfaz fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/repository/PeriodicityRepository.java

5.33. Referencia de la Clase

edu.app.server.controller.PeriodicityRestController

Métodos públicos

- [PeriodicityRestController](#) ([PeriodicityService](#) periodicityService)
- `ResponseEntity< List< Periodicity > > findAll ()`
- `ResponseEntity< Periodicity > newPeriodicity (@Valid @RequestBody Periodicity periodicity)`
- `ResponseEntity< Periodicity > findOne (@PathVariable @Min(1) Long id)`
- `void deleteOne (@PathVariable @Min(1) Long id)`

5.33.1. Descripción detallada

Es el controlador Rest de Periodicity y admite conexiones con cualquier IP.

Autor

Erick David Carpio Hachiri

Ver también

PeriodicityService

5.33.2. Documentación del constructor y destructor

5.33.2.1. PeriodicityRestController()

```
edu.app.server.controller.PeriodicityRestController.PeriodicityRestController (
    PeriodicityService periodicityService )
```

Constructor que inyecta las dependencias de Spring container.

Parámetros

<i>periodicityService</i>	Es el servicio que sera inyectado por Spring.
---------------------------	---

5.33.3. Documentación de las funciones miembro**5.33.3.1. deleteOne()**

```
void edu.app.server.controller.PeriodicityRestController.deleteOne (
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que elimina una periodicidad por su mediante el metodo DELETE.

Parámetros

<i>id</i>	Es el identificador del periodicity.
-----------	--------------------------------------

5.33.3.2. findAll()

```
ResponseEntity<List<Periodicity> > edu.app.server.controller.PeriodicityRestController.findAll ( )
```

Es el servicio Rest que devuelve todos las periodicidades mediante el metodo GET.

Devuelve

Es la respuesta que dara el servicio.

5.33.3.3. findOne()

```
ResponseEntity<Periodicity> edu.app.server.controller.PeriodicityRestController.findOne (
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que busca una periodicidad por id su mediante el metodo POST.

Parámetros

<i>id</i>	Es el identificador del periodicity.
-----------	--------------------------------------

Devuelve

Es la respuesta que dara el servicio.

5.33.3.4. newPeriodicity()

```
ResponseEntity<Periodicity> edu.app.server.controller.PeriodicityRestController.newPeriodicity
(
    @Valid @RequestBody Periodicity periodicity )
```

Es el servicio Rest que crea una nueva periodicidad mediante el metodo PUT.

Parámetros

<i>periodicity</i>	Es la periodicidad que se guardara.
--------------------	-------------------------------------

Devuelve

Es la respuesta que dara el servicio.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/controller/PeriodicityRestController.java

5.34. Referencia de la Clase**edu.app.server.controller.PeriodicityRestControllerTest****Métodos públicos**

- void **whenGetRequestAllNonPeriodicTasks** () throws Exception
- void **whenPostRequestGetNonPeriodicTask** () throws Exception

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/controller/PeriodicityRestControllerTest.java

5.35. Referencia de la Clase edu.app.server.service.PeriodicityService**Métodos públicos**

- [PeriodicityService](#) ([PeriodicityRepository](#) periodicityRepository)
- [Periodicity](#) [getById](#) (Long id)
- List< [Periodicity](#) > [getAllPeriodicity](#) ()
- [Periodicity](#) [savePeriodicity](#) ([Periodicity](#) periodicity)
- [Periodicity](#) [updatePeriodicity](#) ([Periodicity](#) periodicity)
- String [deletePeriodicity](#) ([Periodicity](#) periodicity)

5.35.1. Descripción detallada

Es el modulo de servicio que implementa las logicas de negocio o transformaciones a los datos de Periodicity.

Autor

Erick David Carpio Hachiri

Ver también

Periodicity

5.35.2. Documentación del constructor y destructor

5.35.2.1. PeriodicityService()

```
edu.app.server.service.PeriodicityService.PeriodicityService (
    PeriodicityRepository periodicityRepository )
```

Contructor que inyecta las dependencias de Spring container.

Parámetros

<i>periodicityRepository</i>	Es el repositorio que sera inyectado por Spring.
------------------------------	--

5.35.3. Documentación de las funciones miembro

5.35.3.1. deletePeriodicity()

```
String edu.app.server.service.PeriodicityService.deletePeriodicity (
    Periodicity periodicity )
```

Elimina la periodicidad, mediante una transacción de escritura.

Parámetros

<i>periodicity</i>	Es la periodicidad a eliminar.
--------------------	--------------------------------

Devuelve

Es una cadena con la confirmación.

Ver también

PeriodicityRepository

5.35.3.2. getAllPeriodicity()

```
List<Periodicity> edu.app.server.service.PeriodicityService.getAllPeriodicity ( )
```

Obtiene todas las periodicidades, mediante una transacción de solo lectura.

Devuelve

Son todas las periodicidades realizadas por el repositorio.

Ver también

PeriodicityRepository

5.35.3.3. getById()

```
Periodicity edu.app.server.service.PeriodicityService.getById (
    Long id )
```

Obtiene la Periodicity mediante su id, mediante una transacción de solo lectura.

Parámetros

<i>id</i>	Es el id del Periodicity.
-----------	---------------------------

Devuelve

Retorna el valor existe o no existe que es retornado por el repositorio.

Ver también

PeriodicityRepository

5.35.3.4. savePeriodicity()

```
Periodicity edu.app.server.service.PeriodicityService.savePeriodicity (
    Periodicity periodicity )
```

Guarda la periodicidad, mediante una transacción de escritura.

Parámetros

<i>periodicity</i>	Es la periodicidad a guardar.
--------------------	-------------------------------

Devuelve

Es la periodicidad con el id asignado en la base de datos.

Ver también

PeriodicityRepository

5.35.3.5. updatePeriodicity()

```
Periodicity edu.app.server.service.PeriodicityService.updatePeriodicity (
    Periodicity periodicity )
```

Actualiza la periodicidad, mediante una transacción de escritura.

Parámetros

<i>periodicity</i>	Es la periodicidad a actualizar.
--------------------	----------------------------------

Devuelve

Es la periodicidad con el los valores actualizados en la base de datos.

Ver también

PeriodicityRepository

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/service/PeriodicityService.java

5.36. Referencia del enum edu.app.server.model.PeriodicityType

Atributos públicos

- DAY
- WEEK
- MONTH
- YEAR

5.36.1. Descripción detallada

Define el tipo de periodicidad, si la cantidad de tiempo que se notificaran en la tarea es decir, si serán días, semanas, meses o años.

5.36.2. Documentación de los datos miembro

5.36.2.1. DAY

```
edu.app.server.model.PeriodicityType.DAY
```

Define cada cuantos días serán notificadas las tareas.

5.36.2.2. MONTH

```
edu.app.server.model.PeriodicityType.MONTH
```

Defnie cada cuantos meses serán notificadas las tareas.

5.36.2.3. WEEK

```
edu.app.server.model.PeriodicityType.WEEK
```

Defnie cada cuantas semanas serán notificadas las tareas.

5.36.2.4. YEAR

```
edu.app.server.model.PeriodicityType.YEAR
```

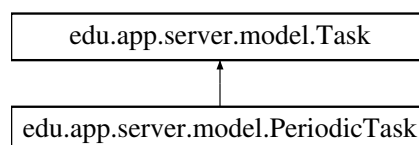
Defnie cada cuantos años serán notificadas las tareas.

La documentación para este enum ha sido generada a partir del siguiente fichero:

- `src/main/java/edu/app/server/model/PeriodicityType.java`

5.37. Referencia de la Clase `edu.app.server.model.PeriodicTask`

Diagrama de herencias de `edu.app.server.model.PeriodicTask`



Métodos públicos

- **PeriodicTask** (String [name](#), String [description](#), [State state](#), [User user](#))

Otros miembros heredados

5.37.1. Descripción detallada

Esta clase hereda de [Task](#). Son las tareas que se repiten de manera constante.

Ver también

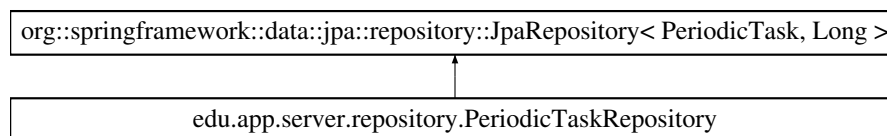
[Task](#)

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/model/PeriodicTask.java

5.38. Referencia de la Interfaz edu.app.server.repository.PeriodicTaskRepository

Diagrama de herencias de edu.app.server.repository.PeriodicTaskRepository



Métodos públicos

- Optional< [PeriodicTask](#) > [getById](#) (Long id)
- Optional< [PeriodicTask](#) > [findByName](#) (String name)

5.38.1. Descripción detallada

Es el repositorio de Periodictask el cual implementa las operaciones de JpaRepository y otras mas personalizadas.

Autor

Erick David Carpio Hachiri

Ver también

[JpaRepository](#)

[PeriodicTask](#)

5.38.2. Documentación de las funciones miembro

5.38.2.1. `findByName()`

```
Optional<PeriodicTask> edu.app.server.repository.PeriodicTaskRepository.findByName (
    String name )
```

Obtiene la PeriodicTask por su nombre.

Parámetros

<i>name</i>	Es el nombre del PeriodicTask
-------------	-------------------------------

Devuelve

Retorna un valor que puede o no puede encontrarse.

Ver también

Optional

5.38.2.2. getById()

```
Optional<PeriodicTask> edu.app.server.repository.PeriodicTaskRepository.getById (
    Long id )
```

Obtiene la PeriodicTask por su id.

Parámetros

<i>id</i>	Es la id del registro en la base de datos de la entidad PeriodicTask
-----------	--

Devuelve

Retorna un valor que puede o no puede encontrarse.

Ver también

Optional

La documentación para este interfaz fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/repository/PeriodicTaskRepository.java

5.39. Referencia de la Clase edu.app.server.repository.PeriodicTaskRepositoryTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/repository/PeriodicTaskRepositoryTest.java

5.40. Referencia de la Clase

edu.app.server.controller.PeriodicTaskRestController

Métodos públicos

- [PeriodicTaskRestController](#) ([PeriodicTaskService](#) periodicTaskService)
- ResponseEntity< List< [PeriodicTask](#) > > [findAll](#) ()
- ResponseEntity< [PeriodicTask](#) > [newPeriodicTask](#) (@Valid @RequestBody [PeriodicTask](#) periodicTask)
- ResponseEntity< [PeriodicTask](#) > [findOne](#) (@PathVariable @Min(1) Long id)
- void [deleteOne](#) (@PathVariable @Min(1) Long id)

5.40.1. Descripción detallada

Es el controlador Rest de PeriodicTask y admite conexiones con cualquier IP.

Autor

Erick David Carpio Hachiri

Ver también

[PeriodicTaskService](#)

5.40.2. Documentación del constructor y destructor

5.40.2.1. PeriodicTaskRestController()

```
edu.app.server.controller.PeriodicTaskRestController.PeriodicTaskRestController (
    PeriodicTaskService periodicTaskService )
```

Constructor que inyecta las dependencias de Spring container.

Parámetros

<i>periodicTaskService</i>	Es el servicio que sera inyectado por Spring.
----------------------------	---

5.40.3. Documentación de las funciones miembro

5.40.3.1. deleteOne()

```
void edu.app.server.controller.PeriodicTaskRestController.deleteOne (
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que elimina una PeriodicTask por su mediante el metodo DELETE.

Parámetros

<i>id</i>	Es el identificador de el periodicTask.
-----------	---

5.40.3.2. findAll()

```
ResponseEntity<List<PeriodicTask> > edu.app.server.controller.PeriodicTaskRestController.↵  
findAll ( )
```

Es el servicio Rest que devuelve todos las periodicTask mediante el metodo GET.

Devuelve

Es la respuesta que dara el servicio.

5.40.3.3. findOne()

```
ResponseEntity<PeriodicTask> edu.app.server.controller.PeriodicTaskRestController.findOne (↵  
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que busca una PeriodicTask por id su mediante el metodo POST.

Parámetros

<i>id</i>	Es el identificador de el periodicTask.
-----------	---

Devuelve

Es la respuesta que dara el servicio.

5.40.3.4. newPeriodicTask()

```
ResponseEntity<PeriodicTask> edu.app.server.controller.PeriodicTaskRestController.newPeriodic↵  
Task (↵  
    @Valid @RequestBody PeriodicTask periodicTask )
```

Es el servicio Rest que crea una nueva periodicTask mediante el metodo PUT.

Parámetros

<i>periodicTask</i>	Es el peridicTask que se guardara.
---------------------	------------------------------------

Devuelve

Es la respuesta que dara el servicio.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/controller/PeriodicTaskRestController.java

5.41. Referencia de la Clase

edu.app.server.controller.PeriodicTaskRestControllerTest

Métodos públicos

- void **whenGetRequestAllNonPeriodicTasks** () throws Exception
- void **whenPostRequestGetNonPeriodicTask** () throws Exception

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/controller/PeriodicTaskRestControllerTest.java

5.42. Referencia de la Clase edu.app.server.service.PeriodicTaskService

Métodos públicos

- [PeriodicTaskService](#) ([PeriodicTaskRepository](#) periodicTaskRepository)
- [PeriodicTask](#) [getById](#) (Long id)
- List< [PeriodicTask](#) > [getAllPeriodicTask](#) ()
- [PeriodicTask](#) [savePeriodicTask](#) ([PeriodicTask](#) periodicTask)
- [PeriodicTask](#) [updatePeriodicTask](#) ([PeriodicTask](#) periodicTask)
- String [deletePeriodicTask](#) ([PeriodicTask](#) periodicTask)
- [PeriodicTask](#) [getByName](#) (String name)

5.42.1. Descripción detallada

Es el modulo de servicio que implementa las logicas de negocio o transformaciones a los datos de PeriodicTask.

Autor

Erick David Carpio Hachiri

Ver también

[PeriodicTask](#)

5.42.2. Documentación del constructor y destructor

5.42.2.1. PeriodicTaskService()

```
edu.app.server.service.PeriodicTaskService.PeriodicTaskService (
    PeriodicTaskRepository periodicTaskRepository )
```

Constructor que inyecta las dependencias de Spring container.

Parámetros

<i>periodicTaskRepository</i>	Es el repositorio que sera inyectado por Spring.
-------------------------------	--

5.42.3. Documentación de las funciones miembro

5.42.3.1. deletePeriodicTask()

```
String edu.app.server.service.PeriodicTaskService.deletePeriodicTask (
    PeriodicTask periodicTask )
```

Elimina la periodicTask, mediante una transacción de escritura.

Parámetros

<i>periodicTask</i>	Es la periodictTask a eliminar.
---------------------	---------------------------------

Devuelve

Es una cadena con la confirmación.

Ver también

PeriodicTaskRepository

5.42.3.2. getAllPeriodicTask()

```
List<PeriodicTask> edu.app.server.service.PeriodicTaskService.getAllPeriodicTask ( )
```

Obtiene todas las periodicTask, mediante una transacción de solo lectura.

Devuelve

Son todas las periodicTask realizadas por el repositorio.

Ver también

PeriodicTaskRepository

5.42.3.3. getById()

```
PeriodicTask edu.app.server.service.PeriodicTaskService.getById (
    Long id )
```

Obtiene la periodicTask mediante su id, mediante una transacción de solo lectura.

Parámetros

<i>id</i>	Es el id del PeriodicTask.
-----------	----------------------------

Devuelve

Retorna el valor existe o no existe que es retornado por el repositorio.

Ver también

PeriodicTaskRepository

5.42.3.4. getName()

```
PeriodicTask edu.app.server.service.PeriodicTaskService.getName (
    String name )
```

Realiza la búsqueda por nombre, mediante una transacción de solo lectura.

Parámetros

<i>name</i>	Es el nombre de la periodicTask.
-------------	----------------------------------

Devuelve

Es la periodicTask que se busco.

Ver también

PeriodicTaskRepository

5.42.3.5. savePeriodicTask()

```
PeriodicTask edu.app.server.service.PeriodicTaskService.savePeriodicTask (
    PeriodicTask periodicTask )
```

Guarda la periodicTask, mediante una transacción de escritura.

Parámetros

<i>periodicTask</i>	Es la autoridad a guardar.
---------------------	----------------------------

Devuelve

Es la periodicTask con el id asignado en la base de datos.

Ver también

PeriodicTaskRepository

5.42.3.6. updatePeriodicTask()

```
PeriodicTask edu.app.server.service.PeriodicTaskService.updatePeriodicTask (
    PeriodicTask periodicTask )
```

Actualiza la periodicTask, mediante una transacción de escritura.

Parámetros

<i>periodicTask</i>	Es la periodicTask a actualizar.
---------------------	----------------------------------

Devuelve

Es la periodicTask con el los valores actualizados en la base de datos.

Ver también

PeriodicTaskRepository

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/service/PeriodicTaskService.java

5.43. Referencia de la Clase

edu.app.server.service.PeriodicTaskServiceTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/test/java/edu/app/server/service/PeriodicTaskServiceTest.java`

5.44. Referencia de la Clase **edu.app.server.model.PeriodicTaskTest**

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/test/java/edu/app/server/model/PeriodicTaskTest.java`

5.45. Referencia de la Clase **edu.app.server.model.Resource**

Métodos públicos

- [Resource](#) (String resource, HttpMethod httpMethod, Boolean isEnabled)

5.45.1. Descripción detallada

La clases [Resource](#) es la encargada de almacenar las URL disponibles y por cual metodo HTTP esta disponible.

5.45.2. Documentación del constructor y destructor

5.45.2.1. Resource()

```
edu.app.server.model.Resource.Resource (
    String resource,
    HttpMethod httpMethod,
    Boolean isEnabled )
```

Este constructor inicializa los valores que no se inicializan de forma automatica.

Parámetros

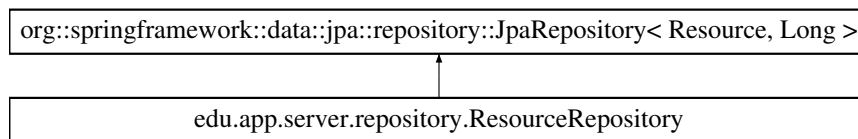
<i>httpMethod</i>	Es el metodo Http que se usa.
<i>resource</i>	Es la url del recurso.
<i>isEnabled</i>	Es el estado del recurso.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/model/Resource.java

5.46. Referencia de la Interfaz edu.app.server.repository.ResourceRepository

Diagrama de herencias de edu.app.server.repository.ResourceRepository



Métodos públicos

- Optional< [Resource](#) > [findByResource](#) (String resource)

5.46.1. Descripción detallada

Es el repositorio de Resource el cual implementa las operaciones de JpaRepository y algunas personalizadas.

Autor

Erick David Carpio Hachiri

Ver también

JpaRepository

Resource

5.46.2. Documentación de las funciones miembro

5.46.2.1. findByResource()

```
Optional<Resource> edu.app.server.repository.ResourceRepository.findByResource (
    String resource )
```

Obtiene el recurso por su nombre.

Parámetros

<i>resource</i>	Es el nombre del recurso.
-----------------	---------------------------

Devuelve

Retorna un valor que puede o no existir.

Ver también

Optional

La documentación para este interfaz fue generada a partir del siguiente fichero:

- `src/main/java/edu/app/server/repository/ResourceRepository.java`

5.47. Referencia de la Clase

`edu.app.server.repository.ResourceRepositoryTest`

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/test/java/edu/app/server/repository/ResourceRepositoryTest.java`

5.48. Referencia de la Clase `edu.app.server.service.ResourceService`

Métodos públicos

- [ResourceService](#) ([ResourceRepository](#) resourceRepository)
- [Resource](#) [getById](#) (Long id)
- List< [Resource](#) > [getAllResources](#) ()
- [Resource](#) [getByName](#) (String name)
- [Resource](#) [saveResource](#) ([Resource](#) resource)
- [Resource](#) [updateResource](#) ([Resource](#) resource)
- String [deleteResource](#) ([Resource](#) resource)

5.48.1. Descripción detallada

Es el modulo de servicio que implementa las logicas de negocio o transformaciones a los datos de Resource.

Autor

Erick David Carpio Hachiri

Ver también

Resource

5.48.2. Documentación del constructor y destructor

5.48.2.1. ResourceService()

```
edu.app.server.service.ResourceService.ResourceService (
    ResourceRepository resourceRepository )
```

Constructor que inyecta las dependencias de Spring container.

Parámetros

<code>resourceRepository</code>	Es el repositorio que sera inyectado por Spring.
---------------------------------	--

5.48.3. Documentación de las funciones miembro**5.48.3.1. deleteResource()**

```
String edu.app.server.service.ResourceService.deleteResource (
    Resource resource )
```

Elimina el recurso, mediante una transacción de escritura.

Parámetros

<code>resource</code>	Es el recurso a eliminar.
-----------------------	---------------------------

Devuelve

Es una cadena con la confirmación.

Ver también

ResourceRepository

5.48.3.2. getAllResources()

```
List<Resource> edu.app.server.service.ResourceService.getAllResources ( )
```

Obtiene todas los recursos, mediante una transacción de solo lectura.

Devuelve

Son todas los recursos realizadas por el repositorio.

Ver también

ResourceRepository

5.48.3.3. getByld()

```
Resource edu.app.server.service.ResourceService.getByld (
    Long id )
```

Obtiene el recurso mediante su id, mediante una transacción de solo lectura.

Parámetros

<i>id</i>	Es el id del Resource.
-----------	------------------------

Devuelve

Retorna el valor existe o no existe que es retornado por el repositorio.

Ver también

ResourceRepository

5.48.3.4. getByName()

```
Resource edu.app.server.service.ResourceService.getByName (
    String name )
```

Realiza la búsqueda por nombre, mediante una transacción de solo lectura.

Parámetros

<i>name</i>	Es el nombre del Recurso.
-------------	---------------------------

Devuelve

Es la autoridad que se busco.

Ver también

ResourceRepository

5.48.3.5. saveResource()

```
Resource edu.app.server.service.ResourceService.saveResource (
    Resource resource )
```

Guarda el recurso, mediante una transacción de escritura.

Parámetros

<i>resource</i>	Es el recurso a guardar.
-----------------	--------------------------

Devuelve

Es el recurso con el id asignado en la base de datos.

Ver también

ResourceRepository

5.48.3.6. updateResource()

```
Resource edu.app.server.service.ResourceService.updateResource (
    Resource resource )
```

Actualiza el recurso autoridad, mediante una transacción de escritura.

Parámetros

<i>resource</i>	Es el recurso a actualizar.
-----------------	-----------------------------

Devuelve

Es el recurso con el los valores actualizados en la base de datos.

Ver también

ResourceRepository

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/service/ResourceService.java

5.49. Referencia de la Clase edu.app.server.service.ResourceServiceTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/service/ResourceServiceTest.java

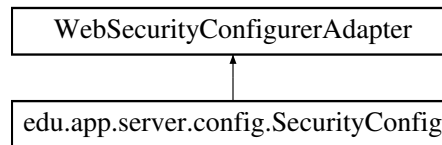
5.50. Referencia de la Clase edu.app.server.model.ResourceTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/model/ResourceTest.java

5.51. Referencia de la Clase edu.app.server.config.SecurityConfig

Diagrama de herencias de edu.app.server.config.SecurityConfig



Métodos públicos

- **SecurityConfig** (BCryptPasswordEncoder bCryptPasswordEncoder, UserDetailsServiceImpl userDetailsService, [UserService](#) userService, [AuthorityService](#) authorityService, [ResourceService](#) resourceService)
- AuthenticationManager **authenticationManagerBean** () throws Exception
- [JWTAuthorizationFilter](#) **jwtAuthorizationFilterBean** ()

Métodos protegidos

- void **configure** (AuthenticationManagerBuilder auth) throws Exception
- void **configure** (HttpSecurity http) throws Exception

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/config/SecurityConfig.java

5.52. Referencia de la Clase edu.app.server.config.SecurityData

Métodos públicos

- BCryptPasswordEncoder **bCryptPasswordEncoder** ()

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/config/SecurityData.java

5.53. Referencia de la Clase edu.app.server.ServerRestApplication

Métodos públicos estáticos

- static void **main** (String[] args)

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/ServerRestApplication.java

5.54. Referencia del enum edu.app.server.model.State

Atributos públicos

- [IN_PROCESS](#)
- [PENDING](#)
- [SUSPENDED](#)
- [TERMINATED](#)

5.54.1. Descripción detallada

Son los estados en los que pueden estar las tareas.

5.54.2. Documentación de los datos miembro

5.54.2.1. IN_PROCESS

```
edu.app.server.model.State.IN_PROCESS
```

Significa que la tarea aun esta en proceso.

5.54.2.2. PENDING

```
edu.app.server.model.State.PENDING
```

Significa que la tarea esta como pendiente.

5.54.2.3. SUSPENDED

```
edu.app.server.model.State.SUSPENDED
```

Significa que la tarea esta suspendida.

5.54.2.4. TERMINATED

```
edu.app.server.model.State.TERMINATED
```

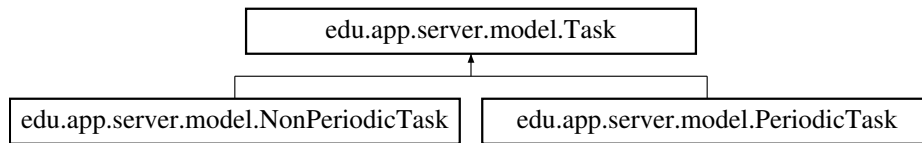
Significa que la tarea termino.

La documentación para este enum ha sido generada a partir del siguiente fichero:

- `src/main/java/edu/app/server/model/State.java`

5.55. Referencia de la Clase edu.app.server.model.Task

Diagrama de herencias de edu.app.server.model.Task



Métodos públicos

- **Task** (String [name](#), String [description](#), [State](#) [state](#), [User](#) [user](#))

Atributos protegidos

- Long [id](#)
- String [name](#)
- String [description](#)
- [State](#) [state](#)
- [User](#) [user](#)

5.55.1. Descripción detallada

Esta clase es de la cual heredan las clases [NonPeriodicTask](#) y [PeriodicTask](#),

5.55.2. Documentación del constructor y destructor

5.55.2.1. Task()

```
edu.app.server.model.Task.Task (
    String name,
    String description,
    State state,
    User user )
```

Es el constructor de la tarea.

Parámetros

<i>name</i>	Es el nombre de la tarea.
<i>description</i>	Es la descripción de la tarea.
<i>state</i>	Es el estado de la tarea.
<i>user</i>	Es el usuario propietario.

5.55.3. Documentación de los datos miembro

5.55.3.1. description

```
String edu.app.server.model.Task.description [protected]
```

Es la descripción de la tarea.

5.55.3.2. id

```
Long edu.app.server.model.Task.id [protected]
```

Es el id de [Task](#), [NonPeriodicTask](#) y [PeriodicTask](#).

5.55.3.3. name

```
String edu.app.server.model.Task.name [protected]
```

Es el nombre de la tarea.

5.55.3.4. state

```
State edu.app.server.model.Task.state [protected]
```

Es el estado de la tarea.

5.55.3.5. user

```
User edu.app.server.model.Task.user [protected]
```

Es el usuario que es propietario de la tarea.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/app/server/model/Task.java`

5.56. Referencia de la Clase `edu.app.server.model.TaskTest`

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/test/java/edu/app/server/model/TaskTest.java`

5.57. Referencia de la Clase edu.app.server.utils.Token

Métodos públicos estáticos

- static String **getToken** ([User](#) user)

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/utils/Token.java

5.58. Referencia de la Clase edu.app.server.model.User

Métodos públicos

- [User](#) (String username, String password, Boolean isEnabled)

5.58.1. Descripción detallada

Esta es la clase [User](#) la cual es la representación de la tabla [User](#) en la base de datos.

5.58.2. Documentación del constructor y destructor

5.58.2.1. User()

```
edu.app.server.model.User.User (
    String username,
    String password,
    Boolean isEnabled )
```

Es el constructor de la clase [User](#) la cual solo tiene como parametros los atributos que no se inicializan automaticamente.

Parámetros

<i>username</i>	Es el nombre de usuario.
<i>password</i>	Es la contraseña del usuario pero ya encriptada.
<i>isEnabled</i>	Es el estado del usuario.

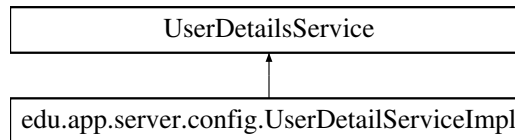
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/model/User.java

5.59. Referencia de la Clase

edu.app.server.config.UserDetailServiceImpl

Diagrama de herencias de edu.app.server.config.UserDetailServiceImpl



Métodos públicos

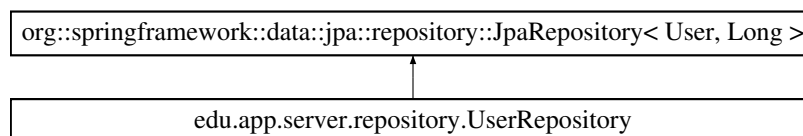
- **UserDetailServiceImpl** ([UserService](#) userService)
- UserDetails **loadUserByUsername** (String username) throws UsernameNotFoundException

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/app/server/config/UserDetailServiceImpl.java`

5.60. Referencia de la Interfaz edu.app.server.repository.UserRepository

Diagrama de herencias de edu.app.server.repository.UserRepository



Métodos públicos

- Optional< [User](#) > **findByUsername** (String username)

5.60.1. Descripción detallada

Es el repositorio de User el cual implementa las operaciones de JpaRepository y operaciones personalizadas.

Autor

Erick David Carpio Hachiri

Ver también

JpaRepository

User

5.60.2. Documentación de las funciones miembro

5.60.2.1. findByUsername()

```
Optional<User> edu.app.server.repository.UserRepository.findByUsername (
    String username )
```

Obtiene le usuario por su nombre.

Parámetros

<i>username</i>	Es el nombre del usuario.
-----------------	---------------------------

Devuelve

Retorna un objeto que puede o encontrarse en la bsae de datos.

Ver también

Optional

La documentación para este interfaz fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/repository/UserRepository.java

5.61. Referencia de la Clase edu.app.server.repository.UserRepositoryTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/repository/UserRepositoryTest.java

5.62. Referencia de la Clase edu.app.server.controller.UserRestController

Métodos públicos

- [UserRestController](#) ([UserService](#) userService)
- [ResponseEntity](#)< [List](#)< [User](#) > > [findAll](#) ()
- [ResponseEntity](#)< [User](#) > [addUser](#) (@Valid @RequestBody [User](#) user)
- void [deleteOne](#) (@PathVariable @Min(1) Long id)

5.62.1. Descripción detallada

Es el controlador Rest de User y admite conexiones con cualquier IP.

Autor

Erick David Carpio Hachiri

Ver también

UserService

5.62.2. Documentación del constructor y destructor

5.62.2.1. UserRestController()

```
edu.app.server.controller.UserRestController.UserRestController (
    UserService userService )
```

Constructor que inyecta las dependencias de Spring container.

Parámetros

<i>userService</i>	Es el servicio que sera inyectado por Spring.
--------------------	---

5.62.3. Documentación de las funciones miembro

5.62.3.1. addUser()

```
ResponseEntity<User> edu.app.server.controller.UserRestController.addUser (
    @Valid @RequestBody User user )
```

Es el servicio Rest que crea un nuevo usuario mediante el metodo PUT.

Parámetros

<i>user</i>	Es el usuario que se guardara en la base de datos.
-------------	--

Devuelve

Es la respuesta que dara el servicio.

5.62.3.2. deleteOne()

```
void edu.app.server.controller.UserRestController.deleteOne (
    @PathVariable @Min(1) Long id )
```

Es el servicio Rest que elimina un usuario, mediante el metodo DELETE.

Parámetros

<i>id</i>	Es el id del usuario que se quiere eliminar.
-----------	--

5.62.3.3. findAll()

```
ResponseEntity<List<User> > edu.app.server.controller.UserRestController.findAll ( )
```

Es el servicio Rest que devuelve todos los usuarios mediante el metodo GET.

Devuelve

Es la respuesta que dara el servicio.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/controller/UserRestController.java

5.63. Referencia de la Clase edu.app.server.controller.UserRestControllerTest

Métodos públicos

- void **testLoginWithJWT** () throws Exception
- void **whenPostRequestToUsersAndValidUserThenCorrectResponse** () throws Exception
- void **whenPostRequestToUsersAndInvalidUserThenCorrectResponse** () throws Exception

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/controller/UserRestControllerTest.java

5.64. Referencia de la Clase edu.app.server.service.UserService

Métodos públicos

- [UserService](#) ([UserRepository](#) userRepository, [BCryptPasswordEncoder](#) bCryptPasswordEncoder)
- [User](#) **saveUser** (@Validated [User](#) user)
- [User](#) **updateUser** (@Validated [User](#) user)
- [User](#) **getByUsername** (String username)
- List< [User](#) > **getAllUsers** ()
- String **deleteUser** ([User](#) user)
- [User](#) **getById** (Long id)

5.64.1. Descripción detallada

Es el modulo de servicio que implementa las logicas de negocio o transformaciones a los datos de User.

Autor

Erick David Carpio Hachiri

Ver también

User

5.64.2. Documentación del constructor y destructor

5.64.2.1. UserService()

```
edu.app.server.service.UserService.UserService (
    UserRepository userRepository,
    BCryptPasswordEncoder bCryptPasswordEncoder )
```

Constructor que inyecta las dependencias de Spring container.

Parámetros

<i>userRepository</i>	Es el repositorio que sera inyectado por Spring.
<i>bCryptPasswordEncoder</i>	Es el encriptador de datos que sera inyectado por Spring.

5.64.3. Documentación de las funciones miembro

5.64.3.1. deleteUser()

```
String edu.app.server.service.UserService.deleteUser (
    User user )
```

Elimina el usuario, mediante una transacción de escritura.

Parámetros

<i>user</i>	Es el usuario a eliminar.
-------------	---------------------------

Devuelve

Es una cadena con la confirmación.

Ver también

UserRepository

5.64.3.2. getAllUsers()

```
List<User> edu.app.server.service.UserService.getAllUsers ( )
```

Obtiene todas los usuarios, mediante una transacción de solo lectura.

Devuelve

Son todas los usuarios realizadas por el repositorio.

Ver también

UserRepository

5.64.3.3. getById()

```
User edu.app.server.service.UserService.getById (
    Long id )
```

Obtiene el usuario mediante su id, mediante una transacción de solo lectura.

Parámetros

<i>id</i>	Es el id del User.
-----------	--------------------

Devuelve

Retorna el valor existe o no existe que es retornado por el repositorio.

Ver también

UserRepository

5.64.3.4. getByUsername()

```
User edu.app.server.service.UserService.getByUsername (
    String username )
```

Realiza la búsqueda por nombre, mediante una transacción de solo lectura.

Parámetros

<i>username</i>	Es el nombre del usuario.
-----------------	---------------------------

Devuelve

Es el usuario que se busco.

Ver también

UserRepository

5.64.3.5. saveUser()

```
User edu.app.server.service.UserService.saveUser (
    @Validated User user )
```

Guarda el usuario, mediante una transacción de escritura.

Parámetros

<i>user</i>	Es el usuario a guardar.
-------------	--------------------------

Devuelve

Es el usuario con el id asignado en la base de datos.

Ver también

UserRepository

5.64.3.6. updateUser()

```
User edu.app.server.service.UserService.updateUser (
    @Validated User user )
```

Actualiza el usuario, mediante una transacción de escritura.

Parámetros

<i>user</i>	Es el usuario a actualizar.
-------------	-----------------------------

Devuelve

Es el usuario con el los valores actualizados en la base de datos.

Ver también

UserRepository

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/service/UserService.java

5.65. Referencia de la Clase edu.app.server.service.UserServiceTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/service/UserServiceTest.java

5.66. Referencia de la Clase edu.app.server.model.UserTest

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/edu/app/server/model/UserTest.java

5.67. Referencia de la Clase edu.app.server.config.ValidationMessage

Métodos públicos

- MessageSource **messageSource** ()
- LocalValidatorFactoryBean **validator** (MessageSource messageSource)

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/app/server/config/ValidationMessage.java

Índice alfabético

- addUser
 - edu.app.server.controller.UserRestController, 70
- attemptAuthentication
 - edu.app.server.config.JWTAuthenticationFilter, 30
- Authority
 - edu.app.server.model.Authority, 12
- AuthorityRestController
 - edu.app.server.controller.AuthorityRestController, 14
- AuthorityService
 - edu.app.server.service.AuthorityService, 16
- DAY
 - edu.app.server.model.PeriodicityType, 46
- deleteAuthority
 - edu.app.server.service.AuthorityService, 17
- deleteGroupTask
 - edu.app.server.service.GroupTaskService, 27
- deleteNonPeriodicTask
 - edu.app.server.service.NonPeriodicTaskService, 36
- deleteOne
 - edu.app.server.controller.AuthorityRestController, 14
 - edu.app.server.controller.GroupTaskRestController, 24
 - edu.app.server.controller.NonPeriodicTaskRestController, 34
 - edu.app.server.controller.PeriodicityRestController, 41
 - edu.app.server.controller.PeriodicTaskRestController, 50
 - edu.app.server.controller.UserRestController, 70
- deletePeriodicity
 - edu.app.server.service.PeriodicityService, 43
- deletePeriodicTask
 - edu.app.server.service.PeriodicTaskService, 53
- deleteResource
 - edu.app.server.service.ResourceService, 60
- deleteUser
 - edu.app.server.service.UserService, 72
- description
 - edu.app.server.model.Task, 66
- edu.app.server, 7
- edu.app.server.config, 7
- edu.app.server.config.Constants, 20
- edu.app.server.config.JWTAuthenticationFilter, 29
 - attemptAuthentication, 30
- edu.app.server.config.JWTAuthorizationFilter, 30
- edu.app.server.config.SecurityConfig, 63
- edu.app.server.config.SecurityData, 63
- edu.app.server.config.UserDetailServiceImpl, 68
- edu.app.server.config.ValidationMessage, 75
- edu.app.server.controller, 8
- edu.app.server.controller.AuthorityRestController, 14
 - AuthorityRestController, 14
 - deleteOne, 14
 - findOne, 15
 - getAll, 15
 - newAuthority, 15
- edu.app.server.controller.AuthorityRestControllerTest, 16
- edu.app.server.controller.GroupTaskRestController, 24
 - deleteOne, 24
 - findAll, 25
 - findOne, 25
 - GroupTaskRestController, 24
 - newGroupTask, 25
- edu.app.server.controller.GroupTaskRestControllerTest, 26
- edu.app.server.controller.NonPeriodicTaskRestController, 33
 - deleteOne, 34
 - findAll, 34
 - findOne, 34
 - newNonPeriodicTask, 35
 - NonPeriodicTaskRestController, 34
- edu.app.server.controller.NonPeriodicTaskRestControllerTest, 35
- edu.app.server.controller.PeriodicityRestController, 40
 - deleteOne, 41
 - findAll, 41
 - findOne, 41
 - newPeriodicity, 42
 - PeriodicityRestController, 40
- edu.app.server.controller.PeriodicityRestControllerTest, 42
- edu.app.server.controller.PeriodicTaskRestController, 50
 - deleteOne, 50
 - findAll, 51
 - findOne, 51
 - newPeriodicTask, 51
 - PeriodicTaskRestController, 50
- edu.app.server.controller.PeriodicTaskRestControllerTest, 52
- edu.app.server.controller.UserRestController, 69
 - addUser, 70

- deleteOne, 70
- findAll, 71
- UserRestController, 70
- edu.app.server.controller.UserRestControllerTest, 71
- edu.app.server.error, 8
- edu.app.server.error.ApiError, 11
- edu.app.server.error.ErrorRestController, 20
 - handleConstraintViolation, 21
- edu.app.server.model, 9
- edu.app.server.model.Authority, 11
 - Authority, 12
- edu.app.server.model.AuthorityTest, 19
- edu.app.server.model.GroupTask, 21
- edu.app.server.model.GroupTaskTest, 29
- edu.app.server.model.NonPeriodicTask, 31
- edu.app.server.model.NonPeriodicTaskTest, 39
- edu.app.server.model.Periodicity, 39
- edu.app.server.model.PeriodicityType, 45
 - DAY, 46
 - MONTH, 46
 - WEEK, 46
 - YEAR, 46
- edu.app.server.model.PeriodicTask, 46
- edu.app.server.model.PeriodicTaskTest, 56
- edu.app.server.model.Resource, 56
 - Resource, 56
- edu.app.server.model.ResourceTest, 62
- edu.app.server.model.State, 64
 - IN_PROCESS, 64
 - PENDING, 64
 - SUSPENDED, 64
 - TERMINATED, 64
- edu.app.server.model.Task, 65
 - description, 66
 - id, 66
 - name, 66
 - state, 66
 - Task, 65
 - user, 66
- edu.app.server.model.TaskTest, 66
- edu.app.server.model.User, 67
 - User, 67
- edu.app.server.model.UserTest, 75
- edu.app.server.repository, 9
- edu.app.server.repository.AuthorityRepository, 12
 - findByAuthority, 13
- edu.app.server.repository.AuthorityRepositoryTest, 13
- edu.app.server.repository.GroupTaskRepository, 22
 - findByName, 22
 - getById, 23
- edu.app.server.repository.GroupTaskRepositoryTest, 23
- edu.app.server.repository.NonPeriodicTaskRepository, 31
 - findByName, 32
 - getById, 32
- edu.app.server.repository.NonPeriodicTaskRepositoryTest, 33
- edu.app.server.repository.PeriodicityRepository, 39
- edu.app.server.repository.PeriodicTaskRepository, 47
 - findByName, 48
 - getById, 49
- edu.app.server.repository.PeriodicTaskRepositoryTest, 49
- edu.app.server.repository.ResourceRepository, 57
 - findByResource, 57
- edu.app.server.repository.ResourceRepositoryTest, 58
- edu.app.server.repository.UserRepository, 68
 - findByUsername, 69
- edu.app.server.repository.UserRepositoryTest, 69
- edu.app.server.ServerRestApplication, 63
- edu.app.server.service, 10
- edu.app.server.service.AuthorityService, 16
 - AuthorityService, 16
 - deleteAuthority, 17
 - getAllAuthorities, 17
 - getById, 17
 - getByName, 18
 - saveAuthority, 18
 - updateAuthority, 19
- edu.app.server.service.AuthorityServiceTest, 19
- edu.app.server.service.GroupTaskService, 26
 - deleteGroupTask, 27
 - getAllGroupTask, 27
 - getById, 27
 - getByName, 28
 - GroupTaskService, 26
 - saveGroupTask, 28
 - updateGroupTask, 28
- edu.app.server.service.GroupTaskServiceTest, 29
- edu.app.server.service.NonPeriodicTaskService, 36
 - deleteNonPeriodicTask, 36
 - getAllNonPeriodicTask, 37
 - getById, 37
 - getByName, 37
 - NonPeriodicTaskService, 36
 - saveNonPeriodicTask, 38
 - updateNonPeriodicTask, 38
- edu.app.server.service.NonPeriodicTaskServiceTest, 39
- edu.app.server.service.PeriodicityService, 42
 - deletePeriodicity, 43
 - getAllPeriodicity, 44
 - getById, 44
 - PeriodicityService, 43
 - savePeriodicity, 44
 - updatePeriodicity, 45
- edu.app.server.service.PeriodicTaskService, 52
 - deletePeriodicTask, 53
 - getAllPeriodicTask, 53
 - getById, 54
 - getByName, 54
 - PeriodicTaskService, 53
 - savePeriodicTask, 55
 - updatePeriodicTask, 55
- edu.app.server.service.PeriodicTaskServiceTest, 56
- edu.app.server.service.ResourceService, 58
 - deleteResource, 60

- getAllResources, [60](#)
 - getById, [60](#)
 - getByName, [61](#)
 - ResourceService, [59](#)
 - saveResource, [61](#)
 - updateResource, [62](#)
- edu.app.server.service.ResourceServiceTest, [62](#)
- edu.app.server.service.UserService, [71](#)
 - deleteUser, [72](#)
 - getAllUsers, [73](#)
 - getById, [73](#)
 - getByUsername, [73](#)
 - saveUser, [74](#)
 - updateUser, [74](#)
 - UserService, [72](#)
- edu.app.server.service.UserServiceTest, [75](#)
- edu.app.server.utils.DateTimeTest, [20](#)
- edu.app.server.utils.Token, [67](#)
- findAll
 - edu.app.server.controller.GroupTaskRestController, [25](#)
 - edu.app.server.controller.NonPeriodicTaskRestController, [34](#)
 - edu.app.server.controller.PeriodicityRestController, [41](#)
 - edu.app.server.controller.PeriodicTaskRestController, [51](#)
 - edu.app.server.controller.UserRestController, [71](#)
- findByAuthority
 - edu.app.server.repository.AuthorityRepository, [13](#)
- findByName
 - edu.app.server.repository.GroupTaskRepository, [22](#)
 - edu.app.server.repository.NonPeriodicTaskRepository, [32](#)
 - edu.app.server.repository.PeriodicTaskRepository, [48](#)
- findByResource
 - edu.app.server.repository.ResourceRepository, [57](#)
- findByUsername
 - edu.app.server.repository.UserRepository, [69](#)
- findOne
 - edu.app.server.controller.AuthorityRestController, [15](#)
 - edu.app.server.controller.GroupTaskRestController, [25](#)
 - edu.app.server.controller.NonPeriodicTaskRestController, [34](#)
 - edu.app.server.controller.PeriodicityRestController, [41](#)
 - edu.app.server.controller.PeriodicTaskRestController, [51](#)
- getAll
 - edu.app.server.controller.AuthorityRestController, [15](#)
- getAllAuthorities
 - edu.app.server.service.AuthorityService, [17](#)
- getAllGroupTask
 - edu.app.server.service.GroupTaskService, [27](#)
- getAllNonPeriodicTask
 - edu.app.server.service.NonPeriodicTaskService, [37](#)
- getAllPeriodicity
 - edu.app.server.service.PeriodicityService, [44](#)
- getAllPeriodicTask
 - edu.app.server.service.PeriodicTaskService, [53](#)
- getAllResources
 - edu.app.server.service.ResourceService, [60](#)
- getAllUsers
 - edu.app.server.service.UserService, [73](#)
- getById
 - edu.app.server.repository.GroupTaskRepository, [23](#)
 - edu.app.server.repository.NonPeriodicTaskRepository, [32](#)
 - edu.app.server.repository.PeriodicTaskRepository, [49](#)
 - edu.app.server.service.AuthorityService, [17](#)
 - edu.app.server.service.GroupTaskService, [27](#)
 - edu.app.server.service.NonPeriodicTaskService, [37](#)
 - edu.app.server.service.PeriodicityService, [44](#)
 - edu.app.server.service.PeriodicTaskService, [54](#)
 - edu.app.server.service.ResourceService, [60](#)
 - edu.app.server.service.UserService, [73](#)
- getByName
 - edu.app.server.service.AuthorityService, [18](#)
 - edu.app.server.service.GroupTaskService, [28](#)
 - edu.app.server.service.NonPeriodicTaskService, [37](#)
 - edu.app.server.service.PeriodicTaskService, [54](#)
 - edu.app.server.service.ResourceService, [61](#)
- getByUsername
 - edu.app.server.service.UserService, [73](#)
- GroupTaskRestController
 - edu.app.server.controller.GroupTaskRestController, [24](#)
- GroupTaskService
 - edu.app.server.service.GroupTaskService, [26](#)
- handleConstraintViolation
 - edu.app.server.error.ErrorRestController, [21](#)
- id
 - edu.app.server.model.Task, [66](#)
- IN_PROCESS
 - edu.app.server.model.State, [64](#)
- MONTH
 - edu.app.server.model.PeriodicityType, [46](#)
- name
 - edu.app.server.model.Task, [66](#)
- newAuthority
 - edu.app.server.controller.AuthorityRestController, [15](#)

newGroupTask
 edu.app.server.controller.GroupTaskRestController, [25](#)
 newNonPeriodicTask
 edu.app.server.controller.NonPeriodicTaskRestController, [35](#)
 newPeriodicity
 edu.app.server.controller.PeriodicityRestController, [42](#)
 newPeriodicTask
 edu.app.server.controller.PeriodicTaskRestController, [51](#)
 NonPeriodicTaskRestController
 edu.app.server.controller.NonPeriodicTaskRestController, [34](#)
 NonPeriodicTaskService
 edu.app.server.service.NonPeriodicTaskService, [36](#)
 PENDING
 edu.app.server.model.State, [64](#)
 PeriodicityRestController
 edu.app.server.controller.PeriodicityRestController, [40](#)
 PeriodicityService
 edu.app.server.service.PeriodicityService, [43](#)
 PeriodicTaskRestController
 edu.app.server.controller.PeriodicTaskRestController, [50](#)
 PeriodicTaskService
 edu.app.server.service.PeriodicTaskService, [53](#)
 Resource
 edu.app.server.model.Resource, [56](#)
 ResourceService
 edu.app.server.service.ResourceService, [59](#)
 saveAuthority
 edu.app.server.service.AuthorityService, [18](#)
 saveGroupTask
 edu.app.server.service.GroupTaskService, [28](#)
 saveNonPeriodicTask
 edu.app.server.service.NonPeriodicTaskService, [38](#)
 savePeriodicity
 edu.app.server.service.PeriodicityService, [44](#)
 savePeriodicTask
 edu.app.server.service.PeriodicTaskService, [55](#)
 saveResource
 edu.app.server.service.ResourceService, [61](#)
 saveUser
 edu.app.server.service.UserService, [74](#)
 state
 edu.app.server.model.Task, [66](#)
 SUSPENDED
 edu.app.server.model.State, [64](#)
 Task
 edu.app.server.model.Task, [65](#)
 TERMINATED
 edu.app.server.model.State, [64](#)
 updateAuthority
 edu.app.server.service.AuthorityService, [19](#)
 updateGroupTask
 edu.app.server.service.GroupTaskService, [28](#)
 updateNonPeriodicTask
 edu.app.server.service.NonPeriodicTaskService, [38](#)
 updatePeriodicity
 edu.app.server.service.PeriodicityService, [45](#)
 updatePeriodicTask
 edu.app.server.service.PeriodicTaskService, [55](#)
 updateResource
 edu.app.server.service.ResourceService, [62](#)
 updateUser
 edu.app.server.service.UserService, [74](#)
 User
 edu.app.server.model.User, [67](#)
 user
 edu.app.server.model.Task, [66](#)
 UserRestController
 edu.app.server.controller.UserRestController, [70](#)
 UserService
 edu.app.server.service.UserService, [72](#)
 WEEK
 edu.app.server.model.PeriodicityType, [46](#)
 YEAR
 edu.app.server.model.PeriodicityType, [46](#)