

Project: "This is my time"
FINAL

Generado por Doxygen 1.8.18

1 Índice jerárquico	1
1.1 Jerarquía de la clase	1
2 Índice de clases	3
2.1 Lista de clases	3
3 Documentación de las clases	5
3.1 Referencia de la Clase edu.app.server.error.ApiError	5
3.1.1 Descripción detallada	5
3.2 Referencia de la Clase edu.app.server.model.Authority	5
3.2.1 Descripción detallada	6
3.2.2 Documentación del constructor y destructor	6
3.2.2.1 Authority()	6
3.3 Referencia de la Interfaz edu.app.server.repository.AuthorityRepository	6
3.3.1 Descripción detallada	7
3.3.2 Documentación de las funciones miembro	7
3.3.2.1 findByAuthority()	7
3.4 Referencia de la Clase edu.app.server.repository.AuthorityRepositoryTest	7
3.5 Referencia de la Clase edu.app.server.controller.AuthorityRestController	8
3.5.1 Descripción detallada	8
3.5.2 Documentación del constructor y destructor	8
3.5.2.1 AuthorityRestController()	8
3.5.3 Documentación de las funciones miembro	8
3.5.3.1 deleteOne()	8
3.5.3.2 findOne()	9
3.5.3.3 getAll()	9
3.5.3.4 newAuthority()	9
3.6 Referencia de la Clase edu.app.server.controller.AuthorityRestControllerTest	10
3.7 Referencia de la Clase edu.app.server.service.AuthorityService	10
3.7.1 Descripción detallada	10
3.7.2 Documentación del constructor y destructor	10
3.7.2.1 AuthorityService()	10
3.7.3 Documentación de las funciones miembro	11
3.7.3.1 deleteAuthority()	11
3.7.3.2 getAllAuthorities()	11
3.7.3.3 getById()	11
3.7.3.4 getName()	12
3.7.3.5 saveAuthority()	12
3.7.3.6 updateAuthority()	13
3.8 Referencia de la Clase edu.app.server.service.AuthorityServiceTest	13
3.9 Referencia de la Clase edu.app.server.model.AuthorityTest	13
3.10 Referencia de la Clase edu.app.server.config.Constants	14
3.11 Referencia de la Clase edu.app.server.utils.DateTimeTest	14

3.12 Referencia de la Clase edu.app.server.error.ErrorRestController	14
3.12.1 Descripción detallada	15
3.12.2 Documentación de las funciones miembro	15
3.12.2.1 handleConstraintViolation()	15
3.13 Referencia de la Clase edu.app.server.model.GroupTask	15
3.13.1 Descripción detallada	16
3.14 Referencia de la Interfaz edu.app.server.repository.GroupTaskRepository	16
3.14.1 Descripción detallada	16
3.14.2 Documentación de las funciones miembro	16
3.14.2.1 findByName()	16
3.14.2.2 getByName()	17
3.15 Referencia de la Clase edu.app.server.repository.GroupTaskRepositoryTest	17
3.16 Referencia de la Clase edu.app.server.controller.GroupTaskRestController	18
3.16.1 Descripción detallada	18
3.16.2 Documentación del constructor y destructor	18
3.16.2.1 GroupTaskRestController()	18
3.16.3 Documentación de las funciones miembro	18
3.16.3.1 deleteOne()	18
3.16.3.2 findAll()	19
3.16.3.3 findOne()	19
3.16.3.4 newGroupTask()	19
3.17 Referencia de la Clase edu.app.server.controller.GroupTaskRestControllerTest	20
3.18 Referencia de la Clase edu.app.server.service.GroupTaskService	20
3.18.1 Descripción detallada	20
3.18.2 Documentación del constructor y destructor	20
3.18.2.1 GroupTaskService()	20
3.18.3 Documentación de las funciones miembro	21
3.18.3.1 deleteGroupTask()	21
3.18.3.2 getAllGroupTask()	21
3.18.3.3 getByName()	21
3.18.3.4 getByName()	22
3.18.3.5 saveGroupTask()	22
3.18.3.6 updateGroupTask()	23
3.19 Referencia de la Clase edu.app.server.service.GroupTaskServiceTest	23
3.20 Referencia de la Clase edu.app.server.model.GroupTaskTest	23
3.21 Referencia de la Clase edu.app.server.config.JWTAuthenticationFilter	23
3.21.1 Documentación de las funciones miembro	24
3.21.1.1 attemptAuthentication()	24
3.22 Referencia de la Clase edu.app.server.config.JWTAuthorizationFilter	24
3.23 Referencia de la Clase edu.app.server.model.NonPeriodicTask	25
3.23.1 Descripción detallada	25
3.24 Referencia de la Interfaz edu.app.server.repository.NonPeriodicTaskRepository	25

3.24.1 Descripción detallada	26
3.24.2 Documentación de las funciones miembro	26
3.24.2.1 <code>findByName()</code>	26
3.24.2.2 <code>getById()</code>	26
3.25 Referencia de la Clase <code>edu.app.server.repository.NonPeriodicTaskRepositoryTest</code>	27
3.26 Referencia de la Clase <code>edu.app.server.controller.NonPeriodicTaskRestController</code>	27
3.26.1 Descripción detallada	27
3.26.2 Documentación del constructor y destructor	28
3.26.2.1 <code>NonPeriodicTaskRestController()</code>	28
3.26.3 Documentación de las funciones miembro	28
3.26.3.1 <code>deleteOne()</code>	28
3.26.3.2 <code>findAll()</code>	28
3.26.3.3 <code>findOne()</code>	29
3.26.3.4 <code>newNonPeriodicTask()</code>	29
3.27 Referencia de la Clase <code>edu.app.server.controller.NonPeriodicTaskRestControllerTest</code>	29
3.28 Referencia de la Clase <code>edu.app.server.service.NonPeriodicTaskService</code>	30
3.28.1 Descripción detallada	30
3.28.2 Documentación del constructor y destructor	30
3.28.2.1 <code>NonPeriodicTaskService()</code>	30
3.28.3 Documentación de las funciones miembro	30
3.28.3.1 <code>deleteNonPeriodicTask()</code>	31
3.28.3.2 <code>getAllNonPeriodicTask()</code>	31
3.28.3.3 <code>getById()</code>	31
3.28.3.4 <code>getByName()</code>	32
3.28.3.5 <code>saveNonPeriodicTask()</code>	32
3.28.3.6 <code>updateNonPeriodicTask()</code>	32
3.29 Referencia de la Clase <code>edu.app.server.service.NonPeriodicTaskServiceTest</code>	33
3.30 Referencia de la Clase <code>edu.app.server.model.NonPeriodicTaskTest</code>	33
3.31 Referencia de la Clase <code>edu.app.server.model.Periodicity</code>	33
3.31.1 Descripción detallada	33
3.32 Referencia de la Interfaz <code>edu.app.server.repository.PeriodicityRepository</code>	33
3.32.1 Descripción detallada	34
3.33 Referencia de la Clase <code>edu.app.server.controller.PeriodicityRestController</code>	34
3.33.1 Descripción detallada	34
3.33.2 Documentación del constructor y destructor	34
3.33.2.1 <code>PeriodicityRestController()</code>	34
3.33.3 Documentación de las funciones miembro	35
3.33.3.1 <code>deleteOne()</code>	35
3.33.3.2 <code>findAll()</code>	35
3.33.3.3 <code>findOne()</code>	35
3.33.3.4 <code>newPeriodicity()</code>	36
3.34 Referencia de la Clase <code>edu.app.server.controller.PeriodicityRestControllerTest</code>	36

3.35 Referencia de la Clase edu.app.server.service.PeriodicityService	36
3.35.1 Descripción detallada	37
3.35.2 Documentación del constructor y destructor	37
3.35.2.1 PeriodicityService()	37
3.35.3 Documentación de las funciones miembro	37
3.35.3.1 deletePeriodicity()	37
3.35.3.2 getAllPeriodicity()	38
3.35.3.3 getById()	38
3.35.3.4 savePeriodicity()	38
3.35.3.5 updatePeriodicity()	39
3.36 Referencia del enum edu.app.server.model.PeriodicityType	39
3.36.1 Descripción detallada	40
3.36.2 Documentación de los datos miembro	40
3.36.2.1 DAY	40
3.36.2.2 MONTH	40
3.36.2.3 WEEK	40
3.36.2.4 YEAR	40
3.37 Referencia de la Clase edu.app.server.model.PeriodicTask	40
3.37.1 Descripción detallada	41
3.38 Referencia de la Interfaz edu.app.server.repository.PeriodicTaskRepository	41
3.38.1 Descripción detallada	41
3.38.2 Documentación de las funciones miembro	42
3.38.2.1 findByName()	42
3.38.2.2 getById()	43
3.39 Referencia de la Clase edu.app.server.repository.PeriodicTaskRepositoryTest	43
3.40 Referencia de la Clase edu.app.server.controller.PeriodicTaskRestController	44
3.40.1 Descripción detallada	44
3.40.2 Documentación del constructor y destructor	44
3.40.2.1 PeriodicTaskRestController()	44
3.40.3 Documentación de las funciones miembro	44
3.40.3.1 deleteOne()	44
3.40.3.2 findAll()	45
3.40.3.3 findOne()	45
3.40.3.4 newPeriodicTask()	45
3.41 Referencia de la Clase edu.app.server.controller.PeriodicTaskRestControllerTest	46
3.42 Referencia de la Clase edu.app.server.service.PeriodicTaskService	46
3.42.1 Descripción detallada	46
3.42.2 Documentación del constructor y destructor	47
3.42.2.1 PeriodicTaskService()	47
3.42.3 Documentación de las funciones miembro	47
3.42.3.1 deletePeriodicTask()	47
3.42.3.2 getAllPeriodicTask()	47

3.42.3.3 getById()	48
3.42.3.4 getByName()	48
3.42.3.5 savePeriodicTask()	49
3.42.3.6 updatePeriodicTask()	49
3.43 Referencia de la Clase edu.app.server.service.PeriodicTaskServiceTest	50
3.44 Referencia de la Clase edu.app.server.model.PeriodicTaskTest	50
3.45 Referencia de la Clase edu.app.server.model.Resource	50
3.45.1 Descripción detallada	50
3.45.2 Documentación del constructor y destructor	50
3.45.2.1 Resource()	50
3.46 Referencia de la Interfaz edu.app.server.repository.ResourceRepository	51
3.46.1 Descripción detallada	51
3.46.2 Documentación de las funciones miembro	51
3.46.2.1 findByResource()	51
3.47 Referencia de la Clase edu.app.server.repository.ResourceRepositoryTest	52
3.48 Referencia de la Clase edu.app.server.service.ResourceService	52
3.48.1 Descripción detallada	52
3.48.2 Documentación del constructor y destructor	53
3.48.2.1 ResourceService()	53
3.48.3 Documentación de las funciones miembro	53
3.48.3.1 deleteResource()	53
3.48.3.2 getAllResources()	53
3.48.3.3 getById()	54
3.48.3.4 getByName()	54
3.48.3.5 saveResource()	55
3.48.3.6 updateResource()	55
3.49 Referencia de la Clase edu.app.server.service.ResourceServiceTest	56
3.50 Referencia de la Clase edu.app.server.model.ResourceTest	56
3.51 Referencia de la Clase edu.app.server.config.SecurityConfig	56
3.52 Referencia de la Clase edu.app.server.config.SecurityData	57
3.53 Referencia de la Clase edu.app.server.ServerRestApplication	57
3.54 Referencia del enum edu.app.server.model.State	57
3.54.1 Descripción detallada	57
3.54.2 Documentación de los datos miembro	57
3.54.2.1 IN_PROCESS	57
3.54.2.2 PENDING	58
3.54.2.3 SUSPENDED	58
3.54.2.4 TERMINATED	58
3.55 Referencia de la Clase edu.app.server.model.Task	58
3.55.1 Descripción detallada	58
3.55.2 Documentación del constructor y destructor	59
3.55.2.1 Task()	59

3.55.3 Documentación de los datos miembro	59
3.55.3.1 description	59
3.55.3.2 id	59
3.55.3.3 name	59
3.55.3.4 state	59
3.55.3.5 user	60
3.56 Referencia de la Clase edu.app.server.model.TaskTest	60
3.57 Referencia de la Clase edu.app.server.utils.Token	60
3.58 Referencia de la Clase edu.app.server.model.User	60
3.58.1 Descripción detallada	60
3.58.2 Documentación del constructor y destructor	60
3.58.2.1 User()	60
3.59 Referencia de la Clase edu.app.server.config.UserDetailServiceImpl	61
3.60 Referencia de la Interfaz edu.app.server.repository.UserRepository	61
3.60.1 Descripción detallada	62
3.60.2 Documentación de las funciones miembro	62
3.60.2.1 findByUsername()	62
3.61 Referencia de la Clase edu.app.server.repository.UserRepositoryTest	62
3.62 Referencia de la Clase edu.app.server.controller.UserRestController	63
3.62.1 Descripción detallada	63
3.62.2 Documentación del constructor y destructor	63
3.62.2.1 UserRestController()	63
3.62.3 Documentación de las funciones miembro	63
3.62.3.1 addUser()	63
3.62.3.2 deleteOne()	65
3.62.3.3 findAll()	65
3.63 Referencia de la Clase edu.app.server.controller.UserRestControllerTest	65
3.64 Referencia de la Clase edu.app.server.service.UserService	66
3.64.1 Descripción detallada	66
3.64.2 Documentación del constructor y destructor	66
3.64.2.1 UserService()	66
3.64.3 Documentación de las funciones miembro	66
3.64.3.1 deleteUser()	67
3.64.3.2 getAllUsers()	67
3.64.3.3 getById()	67
3.64.3.4 getByUsername()	68
3.64.3.5 saveUser()	68
3.64.3.6 updateUser()	69
3.65 Referencia de la Clase edu.app.server.service.UserServiceTest	69
3.66 Referencia de la Clase edu.app.server.model.UserTest	69
3.67 Referencia de la Clase edu.app.server.config.ValidationMessage	69

Capítulo 1

Índice jerárquico

1.1. Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

edu.app.server.error.ApiError	5
edu.app.server.repository.AuthorityRepositoryTest	7
edu.app.server.controller.AuthorityRestController	8
edu.app.server.controller.AuthorityRestControllerTest	10
edu.app.server.service.AuthorityService	10
edu.app.server.service.AuthorityServiceTest	13
edu.app.server.model.AuthorityTest	13
edu.app.server.config.Constants	14
edu.app.server.utils.DateTimeTest	14
edu.app.server.model.GroupTask	15
edu.app.server.repository.GroupTaskRepositoryTest	17
edu.app.server.controller.GroupTaskRestController	18
edu.app.server.controller.GroupTaskRestControllerTest	20
edu.app.server.service.GroupTaskService	20
edu.app.server.service.GroupTaskServiceTest	23
edu.app.server.model.GroupTaskTest	23
edu.app.server.repository.NonPeriodicTaskRepositoryTest	27
edu.app.server.controller.NonPeriodicTaskRestController	27
edu.app.server.controller.NonPeriodicTaskRestControllerTest	29
edu.app.server.service.NonPeriodicTaskService	30
edu.app.server.service.NonPeriodicTaskServiceTest	33
edu.app.server.model.NonPeriodicTaskTest	33
edu.app.server.model.Periodicity	33
edu.app.server.controller.PeriodicityRestController	34
edu.app.server.controller.PeriodicityRestControllerTest	36
edu.app.server.service.PeriodicityService	36
edu.app.server.model.PeriodicityType	39
edu.app.server.repository.PeriodicTaskRepositoryTest	43
edu.app.server.controller.PeriodicTaskRestController	44
edu.app.server.controller.PeriodicTaskRestControllerTest	46
edu.app.server.service.PeriodicTaskService	46
edu.app.server.service.PeriodicTaskServiceTest	50
edu.app.server.model.PeriodicTaskTest	50
edu.app.server.model.Resource	50
edu.app.server.repository.ResourceRepositoryTest	52

edu.app.server.service.ResourceService	52
edu.app.server.service.ResourceServiceTest	56
edu.app.server.model.ResourceTest	56
edu.app.server.config.SecurityData	57
edu.app.server.ServerRestApplication	57
edu.app.server.model.State	57
edu.app.server.model.Task	58
edu.app.server.model.NonPeriodicTask	25
edu.app.server.model.PeriodicTask	40
edu.app.server.model.TaskTest	60
edu.app.server.utils.Token	60
edu.app.server.model.User	60
edu.app.server.repository.UserRepositoryTest	62
edu.app.server.controller.UserRestController	63
edu.app.server.controller.UserRestControllerTest	65
edu.app.server.service.UserService	66
edu.app.server.service.UserServiceTest	69
edu.app.server.model.UserTest	69
edu.app.server.config.ValidationMessage	69
AbstractAuthenticationProcessingFilter	
edu.app.server.config.JWTAuthenticationFilter	23
GrantedAuthority	
edu.app.server.model.Authority	5
JpaRepository	
edu.app.server.repository.AuthorityRepository	6
edu.app.server.repository.GroupTaskRepository	16
edu.app.server.repository.NonPeriodicTaskRepository	25
edu.app.server.repository.PeriodicityRepository	33
edu.app.server.repository.PeriodicTaskRepository	41
edu.app.server.repository.ResourceRepository	51
edu.app.server.repository.UserRepository	61
OncePerRequestFilter	
edu.app.server.config.JWTAuthorizationFilter	24
ResponseEntityExceptionHandler	
edu.app.server.error.ErrorRestController	14
UserDetailsService	
edu.app.server.config.UserDetailServiceImpl	61
WebSecurityConfigurerAdapter	
edu.app.server.config.SecurityConfig	56

Capítulo 2

Índice de clases

2.1. Lista de clases

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

edu.app.server.error.ApiError	5
edu.app.server.model.Authority	5
edu.app.server.repository.AuthorityRepository	6
edu.app.server.repository.AuthorityRepositoryTest	7
edu.app.server.controller.AuthorityRestController	8
edu.app.server.controller.AuthorityRestControllerTest	10
edu.app.server.service.AuthorityService	10
edu.app.server.service.AuthorityServiceTest	13
edu.app.server.model.AuthorityTest	13
edu.app.server.config.Constants	14
edu.app.server.utils.DateTimeTest	14
edu.app.server.error.ErrorRestController	14
edu.app.server.model.GroupTask	15
edu.app.server.repository.GroupTaskRepository	16
edu.app.server.repository.GroupTaskRepositoryTest	17
edu.app.server.controller.GroupTaskRestController	18
edu.app.server.controller.GroupTaskRestControllerTest	20
edu.app.server.service.GroupTaskService	20
edu.app.server.service.GroupTaskServiceTest	23
edu.app.server.model.GroupTaskTest	23
edu.app.server.config.JWTAuthenticationFilter	23
edu.app.server.config.JWTAuthorizationFilter	24
edu.app.server.model.NonPeriodicTask	25
edu.app.server.repository.NonPeriodicTaskRepository	25
edu.app.server.repository.NonPeriodicTaskRepositoryTest	27
edu.app.server.controller.NonPeriodicTaskRestController	27
edu.app.server.controller.NonPeriodicTaskRestControllerTest	29
edu.app.server.service.NonPeriodicTaskService	30
edu.app.server.service.NonPeriodicTaskServiceTest	33
edu.app.server.model.NonPeriodicTaskTest	33
edu.app.server.model.Periodicity	33
edu.app.server.repository.PeriodicityRepository	33
edu.app.server.controller.PeriodicityRestController	34
edu.app.server.controller.PeriodicityRestControllerTest	36
edu.app.server.service.PeriodicityService	36

edu.app.server.model.PeriodicityType	39
edu.app.server.model.PeriodicTask	40
edu.app.server.repository.PeriodicTaskRepository	41
edu.app.server.repository.PeriodicTaskRepositoryTest	43
edu.app.server.controller.PeriodicTaskRestController	44
edu.app.server.controller.PeriodicTaskRestControllerTest	46
edu.app.server.service.PeriodicTaskService	46
edu.app.server.service.PeriodicTaskServiceTest	50
edu.app.server.model.PeriodicTaskTest	50
edu.app.server.model.Resource	50
edu.app.server.repository.ResourceRepository	51
edu.app.server.repository.ResourceRepositoryTest	52
edu.app.server.service.ResourceService	52
edu.app.server.service.ResourceServiceTest	56
edu.app.server.model.ResourceTest	56
edu.app.server.config.SecurityConfig	56
edu.app.server.config.SecurityData	57
edu.app.server.ServerRestApplication	57
edu.app.server.model.State	57
edu.app.server.model.Task	58
edu.app.server.model.TaskTest	60
edu.app.server.utils.Token	60
edu.app.server.model.User	60
edu.app.server.config.UserDetailServiceImpl	61
edu.app.server.repository.UserRepository	61
edu.app.server.repository.UserRepositoryTest	62
edu.app.server.controller.UserRestController	63
edu.app.server.controller.UserRestControllerTest	65
edu.app.server.service.UserService	66
edu.app.server.service.UserServiceTest	69
edu.app.server.model.UserTest	69
edu.app.server.config.ValidationMessage	69

Capítulo 3

Documentación de las clases

3.1. 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)

3.1.1. Descripción detallada

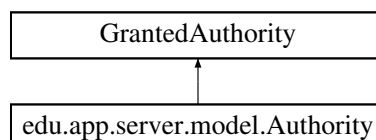
Sirve para enviar respuestas de los errores que se producen al realizar un petición del usuario.

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/error/ApiError.java

3.2. 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)

3.2.1. Descripción detallada

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

3.2.2. Documentación del constructor y destructor

3.2.2.1. Authority()

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

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

Parámetros

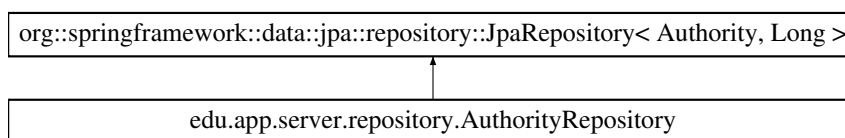
<i>authority</i>	Es el nombre de la autoridad.
<i>isEnabled</i>	Es el estado de la autoridad.

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/Authority.java

3.3. 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)

3.3.1. 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

3.3.2. Documentación de las funciones miembro

3.3.2.1. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/repository/AuthorityRepository.java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/repository/AuthorityRepository↔Test.java

3.5. 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)`

3.5.1. Descripción detallada

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

Autor

Erick David Carpio Hachiri

Ver también

[AuthorityService](#)

3.5.2. Documentación del constructor y destructor

3.5.2.1. 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.
-------------------------	---

3.5.3. Documentación de las funciones miembro

3.5.3.1. 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.
-----------	--------------------------------------

3.5.3.2. 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.

3.5.3.3. 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.

3.5.3.4. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/controller/AuthorityRestController.↔
java

3.6. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/controller/AuthorityRestController↔
Test.java

3.7. Referencia de la Clase edu.app.server.service.AuthorityService

Métodos públicos

- [AuthorityService](#) ([AuthorityRepository](#) authorityRepository)
- [Authority](#) [getById](#) (Long id)
- List< [Authority](#) > [getAllAuthorities](#) ()
- [Authority](#) [saveAuthority](#) ([Authority](#) authority)
- [Authority](#) [updateAuthority](#) ([Authority](#) authority)
- String [deleteAuthority](#) ([Authority](#) authority)
- [Authority](#) [getByName](#) (String name)

3.7.1. 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](#)

3.7.2. Documentación del constructor y destructor**3.7.2.1. 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.
----------------------------------	--

3.7.3. Documentación de las funciones miembro

3.7.3.1. 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

3.7.3.2. 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

3.7.3.3. 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

3.7.3.4. `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

3.7.3.5. `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

3.7.3.6. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/service/AuthorityService.java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/service/AuthorityServiceTest.java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/model/AuthorityTest.java

3.10. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/config/Constants.java

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

Métodos públicos

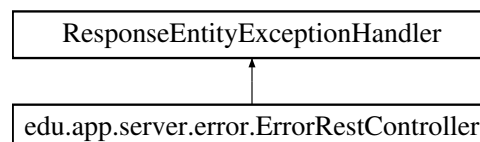
- void **dairyTest** (long day, long month, long year)

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/utils/DateTimeTest.java

3.12. 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`)

3.12.1. Descripción detallada

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

Ver también

[ApiError](#)

3.12.2. Documentación de las funciones miembro

3.12.2.1. 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:

- `C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/error/ErrorRestController.java`

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

Métodos públicos

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

3.13.1. Descripción detallada

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

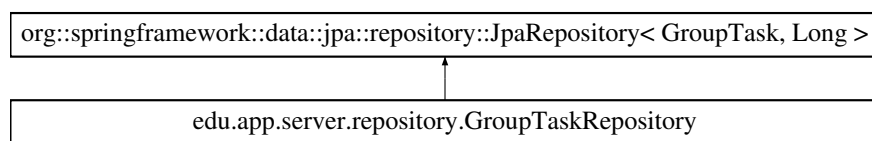
La documentación para esta clase fue generada a partir del siguiente fichero:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/GroupTask.java

3.14. 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)

3.14.1. 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

3.14.2. Documentación de las funciones miembro

3.14.2.1. 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

3.14.2.2. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/repository/GroupTaskRepository.↔
java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/repository/GroupTaskRepository↔
Test.java

3.16. 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)

3.16.1. Descripción detallada

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

Autor

Erick David Carpio Hachiri

Ver también

[GroupTaskService](#)

3.16.2. Documentación del constructor y destructor

3.16.2.1. 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.
-------------------------	---

3.16.3. Documentación de las funciones miembro

3.16.3.1. 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.
-----------	------------------------------------

3.16.3.2. 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.

3.16.3.3. 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.

3.16.3.4. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/controller/GroupTaskRestController.↵
java

3.17. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/controller/GroupTaskRestController↵
Test.java

3.18. 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)

3.18.1. 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](#)

3.18.2. Documentación del constructor y destructor**3.18.2.1. 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

3.18.3. Documentación de las funciones miembro

3.18.3.1. 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.

3.18.3.2. 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.

3.18.3.3. 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

3.18.3.4. `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.

3.18.3.5. `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.

3.18.3.6. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/service/GroupTaskService.java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/service/GroupTaskServiceTest.java

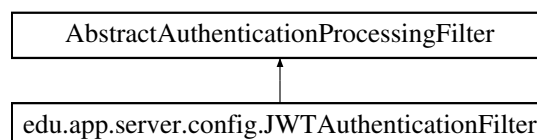
3.20. Referencia de la Clase edu.app.server.model.GroupTaskTest

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/model/GroupTaskTest.java

3.21. 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

3.21.1. Documentación de las funciones miembro

3.21.1.1. 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.

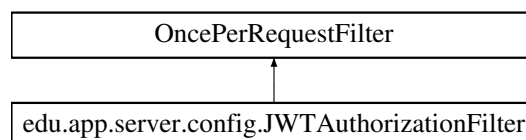
La documentación para esta clase fue generada a partir del siguiente fichero:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/config/JWTAuthenticationFilter.java

3.22. 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

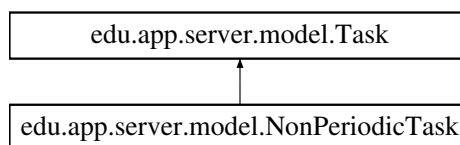
- void **doFilterInternal** (HttpServletRequest req, HttpServletResponse res, FilterChain chain) throws IOException, ServletException

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/config/JWTAuthorizationFilter.java

3.23. 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

3.23.1. 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

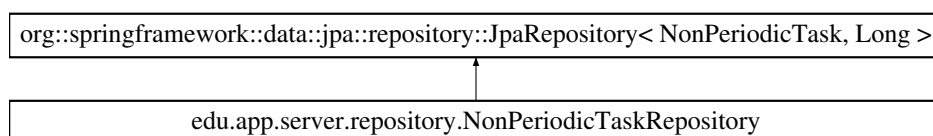
[Task](#)

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/NonPeriodicTask.java

3.24. 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)

3.24.1. 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

3.24.2. Documentación de las funciones miembro

3.24.2.1. 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

3.24.2.2. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/repository/NonPeriodicTaskRepository.java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/repository/NonPeriodicTaskRepositoryTest.java

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

Métodos públicos

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

3.26.1. Descripción detallada

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

Autor

Erick David Carpio Hachiri

Ver también

NonPeriodicTaskService

3.26.2. Documentación del constructor y destructor

3.26.2.1. 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.
-------------------------------	---

3.26.3. Documentación de las funciones miembro

3.26.3.1. 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.
-----------	--

3.26.3.2. 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.

3.26.3.3. 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.

3.26.3.4. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/controller/NonPeriodicTaskRestController.java

3.27. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/controller/NonPeriodicTaskRestControllerTest.java

3.28. 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)

3.28.1. 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](#)

3.28.2. Documentación del constructor y destructor

3.28.2.1. 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.
----------------------------------	---------------------------------------

3.28.3. Documentación de las funciones miembro

3.28.3.1. 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.

3.28.3.2. 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.

3.28.3.3. 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.

3.28.3.4. `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.

3.28.3.5. `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.

3.28.3.6. `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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/service/NonPeriodicTaskService.↵
java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/service/NonPeriodicTaskService↵
Test.java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/model/NonPeriodicTaskTest.java

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

Métodos públicos

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

3.31.1. Descripción detallada

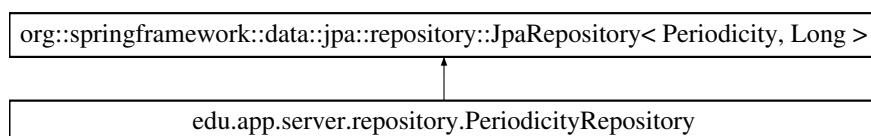
Define la periodicidad de las tareas.

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/Periodicity.java

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

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



3.32.1. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/repository/PeriodicityRepository.↔
java

3.33. 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)

3.33.1. Descripción detallada

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

Autor

Erick David Carpio Hachiri

Ver también

PeriodicityService

3.33.2. Documentación del constructor y destructor

3.33.2.1. 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.
---------------------------	---

3.33.3. Documentación de las funciones miembro**3.33.3.1. 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.
-----------	--------------------------------------

3.33.3.2. 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.

3.33.3.3. 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.

3.33.3.4. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/controller/PeriodicityRestController.↔
java

3.34. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/controller/PeriodicityRestController↔
Test.java

3.35. 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)

3.35.1. 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

3.35.2. Documentación del constructor y destructor

3.35.2.1. 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.
------------------------------	--

3.35.3. Documentación de las funciones miembro

3.35.3.1. 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

3.35.3.2. 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

3.35.3.3. 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

3.35.3.4. 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

3.35.3.5. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/service/PeriodicityService.java

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

Atributos públicos

- DAY
- WEEK
- MONTH
- YEAR

3.36.1. 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.

3.36.2. Documentación de los datos miembro

3.36.2.1. DAY

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

Define cada cuantos días serán notificadas las tareas.

3.36.2.2. MONTH

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

Defnie cada cuantos meses serán notificadas las tareas.

3.36.2.3. WEEK

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

Defnie cada cuantas semanas serán notificadas las tareas.

3.36.2.4. YEAR

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

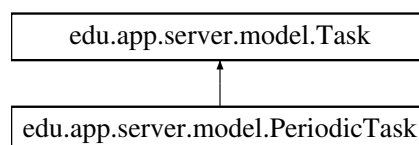
Defnie cada cuantos años serán notificadas las tareas.

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/PeriodicityType.java

3.37. 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

3.37.1. Descripción detallada

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

Ver también

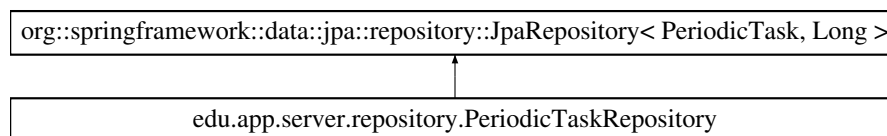
[Task](#)

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/PeriodicTask.java

3.38. 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)

3.38.1. 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](#)

3.38.2. Documentación de las funciones miembro

3.38.2.1. `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

3.38.2.2. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/repository/PeriodicTaskRepository.java↵

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/repository/PeriodicTaskRepositoryTest.java↵

3.40. 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)

3.40.1. Descripción detallada

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

Autor

Erick David Carpio Hachiri

Ver también

[PeriodicTaskService](#)

3.40.2. Documentación del constructor y destructor

3.40.2.1. 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.
----------------------------	---

3.40.3. Documentación de las funciones miembro

3.40.3.1. 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.
-----------	---

3.40.3.2. 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.

3.40.3.3. 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.

3.40.3.4. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/controller/PeriodicTaskRestController.java

3.41. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/controller/PeriodicTaskRestControllerTest.java

3.42. 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)

3.42.1. 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](#)

3.42.2. Documentación del constructor y destructor

3.42.2.1. PeriodicTaskService()

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

Contructor que inyecta las dependencias de Spring container.

Parámetros

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

3.42.3. Documentación de las funciones miembro

3.42.3.1. 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

3.42.3.2. 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

3.42.3.3. 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

3.42.3.4. getByName()

```
PeriodicTask edu.app.server.service.PeriodicTaskService.getByName (
    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

3.42.3.5. 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

3.42.3.6. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/service/PeriodicTaskService.java

3.43. Referencia de la Clase `edu.app.server.service.PeriodicTaskServiceTest`

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

- `C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/service/PeriodicTaskServiceTest.`↵
`java`

3.44. Referencia de la Clase `edu.app.server.model.PeriodicTaskTest`

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

- `C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/model/PeriodicTaskTest.java`

3.45. Referencia de la Clase `edu.app.server.model.Resource`

Métodos públicos

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

3.45.1. Descripción detallada

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

3.45.2. Documentación del constructor y destructor

3.45.2.1. 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

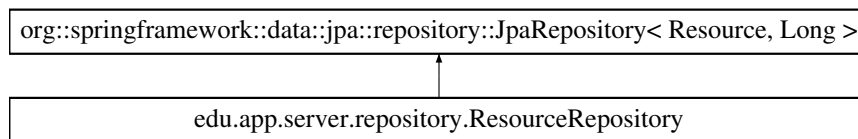
<i>httpMethod</i>	Es el metodo Http que se usa.
<i>resource</i>	Es la url del recurso.
<i>isEnabled</i>	Es el estado del recurso.

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/Resource.java

3.46. 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)

3.46.1. 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

3.46.2. Documentación de las funciones miembro

3.46.2.1. findByResource()

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

Obtiene el recurso por su nombre.

Parámetros

<code>resource</code>	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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/repository/ResourceRepository.java↔

3.47. Referencia de la Clase

edu.app.server.repository.ResourceRepositoryTest

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/repository/ResourceRepositoryTest.java↔

3.48. 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)

3.48.1. 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

3.48.2. Documentación del constructor y destructor

3.48.2.1. ResourceService()

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

Constructor que inyecta las dependencias de Spring container.

Parámetros

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

3.48.3. Documentación de las funciones miembro

3.48.3.1. deleteResource()

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

Elimina el recurso, mediante una transacción de escritura.

Parámetros

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

Devuelve

Es una cadena con la confirmación.

Ver también

[ResourceRepository](#)

3.48.3.2. 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

3.48.3.3. getById()

```
Resource edu.app.server.service.ResourceService.getById (
    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

3.48.3.4. getName()

```
Resource edu.app.server.service.ResourceService.getName (
    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

3.48.3.5. 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

3.48.3.6. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/service/ResourceService.java

3.49. Referencia de la Clase `edu.app.server.service.ResourceServiceTest`

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

- `C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/service/ResourceServiceTest.java`

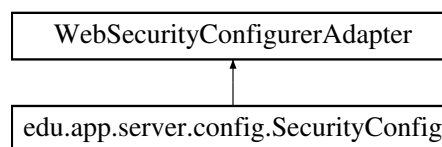
3.50. Referencia de la Clase `edu.app.server.model.ResourceTest`

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

- `C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/model/ResourceTest.java`

3.51. Referencia de la Clase `edu.app.server.config.SecurityConfig`

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



Métodos públicos

- **SecurityConfig** (`BCryptPasswordEncoder bCryptPasswordEncoder`, `UserDetailsService userDetailService`, `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:

- `C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/config/SecurityConfig.java`

3.52. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/config/SecurityData.java

3.53. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/ServerRestApplication.java

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

Atributos públicos

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

3.54.1. Descripción detallada

Son los estados en los que pueden estar las tareas.

3.54.2. Documentación de los datos miembro

3.54.2.1. IN_PROCESS

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

Significa que la tarea aun esta en proceso.

3.54.2.2. PENDING

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

Significa que la tarea esta como pendiente.

3.54.2.3. SUSPENDED

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

Significa que la tarea esta suspendida.

3.54.2.4. TERMINATED

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

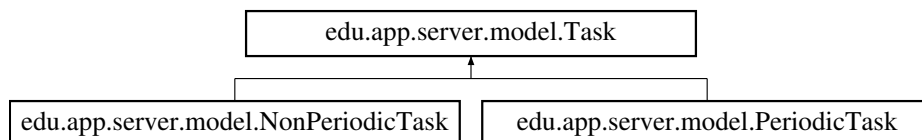
Significa que la tarea termino.

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/State.java

3.55. 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

3.55.1. Descripción detallada

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

3.55.2. Documentación del constructor y destructor

3.55.2.1. 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.

3.55.3. Documentación de los datos miembro

3.55.3.1. description

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

Es la descripción de la tarea.

3.55.3.2. id

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

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

3.55.3.3. name

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

Es el nombre de la tarea.

3.55.3.4. state

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

Es el estado de la tarea.

3.55.3.5. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/Task.java

3.56. Referencia de la Clase edu.app.server.model.TaskTest

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/model/TaskTest.java

3.57. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/utils/Token.java

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

Métodos públicos

- `User` (String username, String password, Boolean isEnabled)

3.58.1. Descripción detallada

Esta es la clase `User` la cual es la representación de la tabla `User` en la base de datos.

3.58.2. Documentación del constructor y destructor

3.58.2.1. 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

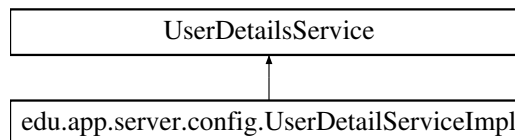
<i>username</i>	Es el nombre de usuario.
<i>password</i>	Es la contraseña del usuario pero ya encriptada.
<i>isEnabled</i>	Es el estado del usuario.

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/model/User.java

3.59. Referencia de la Clase edu.app.server.config.UserDetailServiceImpl

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



Métodos públicos

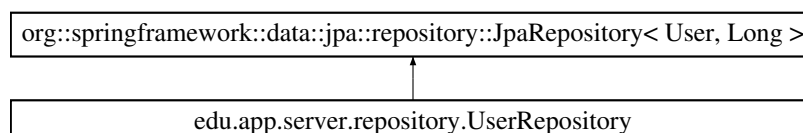
- **UserDetailServiceImpl** ([UserService](#) userService)
- UserDetailsService **loadUserByUsername** (String username) throws UsernameNotFoundException

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/config/UserDetailServiceImpl.java

3.60. 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)

3.60.1. 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

3.60.2. Documentación de las funciones miembro

3.60.2.1. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/repository/UserRepository.java

3.61. Referencia de la Clase

edu.app.server.repository.UserRepositoryTest

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/repository/UserRepositoryTest.java

3.62. 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)

3.62.1. Descripción detallada

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

Autor

Erick David Carpio Hachiri

Ver también

`UserService`

3.62.2. Documentación del constructor y destructor

3.62.2.1. UserRestController()

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

Constructor que inyecta las dependencias de Spring container.

Parámetros

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

3.62.3. Documentación de las funciones miembro

3.62.3.1. 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.

3.62.3.2. 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.
-----------	--

3.62.3.3. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/controller/UserRestController.java

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

Métodos públicos

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/controller/UserRestControllerTest.↵
java

3.64. 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)

3.64.1. 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](#)

3.64.2. Documentación del constructor y destructor

3.64.2.1. 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.

3.64.3. Documentación de las funciones miembro

3.64.3.1. 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

3.64.3.2. 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

3.64.3.3. 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

3.64.3.4. 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

3.64.3.5. 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

3.64.3.6. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/service/UserService.java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/service/UserServiceTest.java

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

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

- C:/Users/erick/IdeaProjects/this-is-my-time/src/test/java/edu/app/server/model/UserTest.java

3.67. 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:

- C:/Users/erick/IdeaProjects/this-is-my-time/src/main/java/edu/app/server/config/ValidationMessage.java

Índice alfabético

- addUser
 - edu.app.server.controller.UserRestController, 63
- attemptAuthentication
 - edu.app.server.config.JWTAuthenticationFilter, 24
- Authority
 - edu.app.server.model.Authority, 6
- AuthorityRestController
 - edu.app.server.controller.AuthorityRestController, 8
- AuthorityService
 - edu.app.server.service.AuthorityService, 10
- DAY
 - edu.app.server.model.PeriodicityType, 40
- deleteAuthority
 - edu.app.server.service.AuthorityService, 11
- deleteGroupTask
 - edu.app.server.service.GroupTaskService, 21
- deleteNonPeriodicTask
 - edu.app.server.service.NonPeriodicTaskService, 30
- deleteOne
 - edu.app.server.controller.AuthorityRestController, 8
 - edu.app.server.controller.GroupTaskRestController, 18
 - edu.app.server.controller.NonPeriodicTaskRestController, 28
 - edu.app.server.controller.PeriodicityRestController, 35
 - edu.app.server.controller.PeriodicTaskRestController, 44
 - edu.app.server.controller.UserRestController, 65
- deletePeriodicity
 - edu.app.server.service.PeriodicityService, 37
- deletePeriodicTask
 - edu.app.server.service.PeriodicTaskService, 47
- deleteResource
 - edu.app.server.service.ResourceService, 53
- deleteUser
 - edu.app.server.service.UserService, 66
- description
 - edu.app.server.model.Task, 59
- edu.app.server.config.Constants, 14
- edu.app.server.config.JWTAuthenticationFilter, 23
 - attemptAuthentication, 24
- edu.app.server.config.JWTAuthorizationFilter, 24
- edu.app.server.config.SecurityConfig, 56
- edu.app.server.config.SecurityData, 57
- edu.app.server.config.UserDetailServiceImpl, 61
- edu.app.server.config.ValidationMessage, 69
- edu.app.server.controller.AuthorityRestController, 8
 - AuthorityRestController, 8
 - deleteOne, 8
 - findOne, 9
 - getAll, 9
 - newAuthority, 9
- edu.app.server.controller.AuthorityRestControllerTest, 10
- edu.app.server.controller.GroupTaskRestController, 18
 - deleteOne, 18
 - findAll, 19
 - findOne, 19
 - GroupTaskRestController, 18
 - newGroupTask, 19
- edu.app.server.controller.GroupTaskRestControllerTest, 20
- edu.app.server.controller.NonPeriodicTaskRestController, 27
 - deleteOne, 28
 - findAll, 28
 - findOne, 28
 - newNonPeriodicTask, 29
 - NonPeriodicTaskRestController, 28
- edu.app.server.controller.NonPeriodicTaskRestControllerTest, 29
- edu.app.server.controller.PeriodicityRestController, 34
 - deleteOne, 35
 - findAll, 35
 - findOne, 35
 - newPeriodicity, 36
 - PeriodicityRestController, 34
- edu.app.server.controller.PeriodicityRestControllerTest, 36
- edu.app.server.controller.PeriodicTaskRestController, 44
 - deleteOne, 44
 - findAll, 45
 - findOne, 45
 - newPeriodicTask, 45
 - PeriodicTaskRestController, 44
- edu.app.server.controller.PeriodicTaskRestControllerTest, 46
- edu.app.server.controller.UserRestController, 63
 - addUser, 63
 - deleteOne, 65
 - findAll, 65
 - UserRestController, 63

- edu.app.server.controller.UserRestControllerTest, 65
- edu.app.server.error.ApiError, 5
- edu.app.server.error.ErrorRestController, 14
 - handleConstraintViolation, 15
- edu.app.server.model.Authority, 5
 - Authority, 6
- edu.app.server.model.AuthorityTest, 13
- edu.app.server.model.GroupTask, 15
- edu.app.server.model.GroupTaskTest, 23
- edu.app.server.model.NonPeriodicTask, 25
- edu.app.server.model.NonPeriodicTaskTest, 33
- edu.app.server.model.Periodicity, 33
- edu.app.server.model.PeriodicityType, 39
 - DAY, 40
 - MONTH, 40
 - WEEK, 40
 - YEAR, 40
- edu.app.server.model.PeriodicTask, 40
- edu.app.server.model.PeriodicTaskTest, 50
- edu.app.server.model.Resource, 50
 - Resource, 50
- edu.app.server.model.ResourceTest, 56
- edu.app.server.model.State, 57
 - IN_PROCESS, 57
 - PENDING, 57
 - SUSPENDED, 58
 - TERMINATED, 58
- edu.app.server.model.Task, 58
 - description, 59
 - id, 59
 - name, 59
 - state, 59
 - Task, 59
 - user, 59
- edu.app.server.model.TaskTest, 60
- edu.app.server.model.User, 60
 - User, 60
- edu.app.server.model.UserTest, 69
- edu.app.server.repository.AuthorityRepository, 6
 - findByAuthority, 7
- edu.app.server.repository.AuthorityRepositoryTest, 7
- edu.app.server.repository.GroupTaskRepository, 16
 - findByName, 16
 - getById, 17
- edu.app.server.repository.GroupTaskRepositoryTest, 17
- edu.app.server.repository.NonPeriodicTaskRepository, 25
 - findByName, 26
 - getById, 26
- edu.app.server.repository.NonPeriodicTaskRepositoryTest, 27
- edu.app.server.repository.PeriodicityRepository, 33
- edu.app.server.repository.PeriodicTaskRepository, 41
 - findByName, 42
 - getById, 43
- edu.app.server.repository.PeriodicTaskRepositoryTest, 43
- edu.app.server.repository.ResourceRepository, 51
 - findByResource, 51
- edu.app.server.repository.ResourceRepositoryTest, 52
- edu.app.server.repository.UserRepository, 61
 - findByUsername, 62
- edu.app.server.repository.UserRepositoryTest, 62
- edu.app.server.ServerRestApplication, 57
- edu.app.server.service.AuthorityService, 10
 - AuthorityService, 10
 - deleteAuthority, 11
 - getAllAuthorities, 11
 - getById, 11
 - getByName, 12
 - saveAuthority, 12
 - updateAuthority, 13
- edu.app.server.service.AuthorityServiceTest, 13
- edu.app.server.service.GroupTaskService, 20
 - deleteGroupTask, 21
 - getAllGroupTask, 21
 - getById, 21
 - getByName, 22
 - GroupTaskService, 20
 - saveGroupTask, 22
 - updateGroupTask, 22
- edu.app.server.service.GroupTaskServiceTest, 23
- edu.app.server.service.NonPeriodicTaskService, 30
 - deleteNonPeriodicTask, 30
 - getAllNonPeriodicTask, 31
 - getById, 31
 - getByName, 31
 - NonPeriodicTaskService, 30
 - saveNonPeriodicTask, 32
 - updateNonPeriodicTask, 32
- edu.app.server.service.NonPeriodicTaskServiceTest, 33
- edu.app.server.service.PeriodicityService, 36
 - deletePeriodicity, 37
 - getAllPeriodicity, 38
 - getById, 38
 - PeriodicityService, 37
 - savePeriodicity, 38
 - updatePeriodicity, 39
- edu.app.server.service.PeriodicTaskService, 46
 - deletePeriodicTask, 47
 - getAllPeriodicTask, 47
 - getById, 48
 - getByName, 48
 - PeriodicTaskService, 47
 - savePeriodicTask, 49
 - updatePeriodicTask, 49
- edu.app.server.service.PeriodicTaskServiceTest, 50
- edu.app.server.service.ResourceService, 52
 - deleteResource, 53
 - getAllResources, 53
 - getById, 54
 - getByName, 54
 - ResourceService, 53
 - saveResource, 55
 - updateResource, 55
- edu.app.server.service.ResourceServiceTest, 56

- edu.app.server.service.UserService, 66
 - deleteUser, 66
 - getAllUsers, 67
 - getById, 67
 - getByUsername, 68
 - saveUser, 68
 - updateUser, 69
 - UserService, 66
- edu.app.server.service.UserServiceTest, 69
- edu.app.server.utils.DateTimeTest, 14
- edu.app.server.utils.Token, 60
- findAll
 - edu.app.server.controller.GroupTaskRestController, 19
 - edu.app.server.controller.NonPeriodicTaskRestController, 28
 - edu.app.server.controller.PeriodicityRestController, 35
 - edu.app.server.controller.PeriodicTaskRestController, 45
 - edu.app.server.controller.UserRestController, 65
- findByAuthority
 - edu.app.server.repository.AuthorityRepository, 7
- findByName
 - edu.app.server.repository.GroupTaskRepository, 16
 - edu.app.server.repository.NonPeriodicTaskRepository, 26
 - edu.app.server.repository.PeriodicTaskRepository, 42
- findByResource
 - edu.app.server.repository.ResourceRepository, 51
- findByUsername
 - edu.app.server.repository.UserRepository, 62
- findOne
 - edu.app.server.controller.AuthorityRestController, 9
 - edu.app.server.controller.GroupTaskRestController, 19
 - edu.app.server.controller.NonPeriodicTaskRestController, 28
 - edu.app.server.controller.PeriodicityRestController, 35
 - edu.app.server.controller.PeriodicTaskRestController, 45
- getAll
 - edu.app.server.controller.AuthorityRestController, 9
- getAllAuthorities
 - edu.app.server.service.AuthorityService, 11
- getAllGroupTask
 - edu.app.server.service.GroupTaskService, 21
- getAllNonPeriodicTask
 - edu.app.server.service.NonPeriodicTaskService, 31
- getAllPeriodicity
 - edu.app.server.service.PeriodicityService, 38
- getAllPeriodicTask
 - edu.app.server.service.PeriodicTaskService, 47
- getAllResources
 - edu.app.server.service.ResourceService, 53
- getAllUsers
 - edu.app.server.service.UserService, 67
- getById
 - edu.app.server.repository.GroupTaskRepository, 17
 - edu.app.server.repository.NonPeriodicTaskRepository, 26
 - edu.app.server.repository.PeriodicTaskRepository, 43
 - edu.app.server.service.AuthorityService, 11
 - edu.app.server.service.GroupTaskService, 21
 - edu.app.server.service.NonPeriodicTaskService, 31
 - edu.app.server.service.PeriodicityService, 38
 - edu.app.server.service.PeriodicTaskService, 48
 - edu.app.server.service.ResourceService, 54
 - edu.app.server.service.UserService, 67
- getByName
 - edu.app.server.service.AuthorityService, 12
 - edu.app.server.service.GroupTaskService, 22
 - edu.app.server.service.NonPeriodicTaskService, 31
 - edu.app.server.service.PeriodicTaskService, 48
 - edu.app.server.service.ResourceService, 54
- getByUsername
 - edu.app.server.service.UserService, 68
- GroupTaskRestController
 - edu.app.server.controller.GroupTaskRestController, 18
- GroupTaskService
 - edu.app.server.service.GroupTaskService, 20
- handleConstraintViolation
 - edu.app.server.error.ErrorRestController, 15
- id
 - edu.app.server.model.Task, 59
- IN_PROCESS
 - edu.app.server.model.State, 57
- MONTH
 - edu.app.server.model.PeriodicityType, 40
- name
 - edu.app.server.model.Task, 59
- newAuthority
 - edu.app.server.controller.AuthorityRestController, 9
- newGroupTask
 - edu.app.server.controller.GroupTaskRestController, 19
- newNonPeriodicTask
 - edu.app.server.controller.NonPeriodicTaskRestController, 29
- newPeriodicity

edu.app.server.controller.PeriodicityRestController, [36](#)
 newPeriodicTask
 edu.app.server.controller.PeriodicTaskRestController, [45](#)
 NonPeriodicTaskRestController
 edu.app.server.controller.NonPeriodicTaskRestController, [28](#)
 NonPeriodicTaskService
 edu.app.server.service.NonPeriodicTaskService, [30](#)
 PENDING
 edu.app.server.model.State, [57](#)
 PeriodicityRestController
 edu.app.server.controller.PeriodicityRestController, [34](#)
 PeriodicityService
 edu.app.server.service.PeriodicityService, [37](#)
 PeriodicTaskRestController
 edu.app.server.controller.PeriodicTaskRestController, [44](#)
 PeriodicTaskService
 edu.app.server.service.PeriodicTaskService, [47](#)
 Resource
 edu.app.server.model.Resource, [50](#)
 ResourceService
 edu.app.server.service.ResourceService, [53](#)
 saveAuthority
 edu.app.server.service.AuthorityService, [12](#)
 saveGroupTask
 edu.app.server.service.GroupTaskService, [22](#)
 saveNonPeriodicTask
 edu.app.server.service.NonPeriodicTaskService, [32](#)
 savePeriodicity
 edu.app.server.service.PeriodicityService, [38](#)
 savePeriodicTask
 edu.app.server.service.PeriodicTaskService, [49](#)
 saveResource
 edu.app.server.service.ResourceService, [55](#)
 saveUser
 edu.app.server.service.UserService, [68](#)
 state
 edu.app.server.model.Task, [59](#)
 SUSPENDED
 edu.app.server.model.State, [58](#)
 Task
 edu.app.server.model.Task, [59](#)
 TERMINATED
 edu.app.server.model.State, [58](#)
 updateAuthority
 edu.app.server.service.AuthorityService, [13](#)
 updateGroupTask
 edu.app.server.service.GroupTaskService, [22](#)
 updateNonPeriodicTask
 edu.app.server.service.NonPeriodicTaskService, [32](#)
 updatePeriodicity
 edu.app.server.service.PeriodicityService, [39](#)
 updatePeriodicTask
 edu.app.server.service.PeriodicTaskService, [49](#)
 updateResource
 edu.app.server.service.ResourceService, [55](#)
 updateUser
 edu.app.server.service.UserService, [69](#)
 User
 edu.app.server.model.User, [60](#)
 user
 edu.app.server.model.Task, [59](#)
 UserRestController
 edu.app.server.controller.UserRestController, [63](#)
 UserService
 edu.app.server.service.UserService, [66](#)
 WEEK
 edu.app.server.model.PeriodicityType, [40](#)
 YEAR
 edu.app.server.model.PeriodicityType, [40](#)