

My Project

Generated by Doxygen 1.9.6

1 Namespace Index	1
1.1 Package List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 Package com.example.backend	9
5.2 Package com.example.backend.configuration	9
5.3 Package com.example.backend.controller	9
5.4 Package com.example.backend.dto	10
5.5 Package com.example.backend.entity	10
5.6 Package com.example.backend.scheduler	10
5.7 Package com.example.backend.service	10
5.8 Package com.example.backend.service.impl	10
6 Class Documentation	11
6.1 com.example.backend.controller.AlertController Class Reference	11
6.1.1 Detailed Description	11
6.1.2 Member Function Documentation	11
6.1.2.1 getAlerts()	12
6.1.2.2 postAlerts()	12
6.2 com.example.backend.entity.AlertEntity Class Reference	12
6.2.1 Detailed Description	12
6.2.2 Member Function Documentation	13
6.2.2.1 get_category()	13
6.2.2.2 get_description()	13
6.2.2.3 get_entity_id()	13
6.2.2.4 get_entity_name()	13
6.2.2.5 get_id()	13
6.2.2.6 get_pod()	14
6.2.2.7 get_status()	14
6.2.2.8 get_time()	14
6.2.2.9 set_category()	14
6.2.2.10 set_description()	15
6.2.2.11 set_entity_id()	15
6.2.2.12 set_entity_name()	15
6.2.2.13 set_id()	16

6.2.2.14 set_pod()	16
6.2.2.15 set_status()	16
6.2.2.16 set_time()	17
6.3 com.example.backend.scheduler.AlertScheduler Class Reference	17
6.3.1 Detailed Description	17
6.3.2 Member Function Documentation	18
6.3.2.1 pollingAlert()	18
6.4 com.example.backend.dto.AlertStatusDto Class Reference	18
6.4.1 Detailed Description	18
6.5 com.example.backend.dto.AllEmployeeInfoOutDto Class Reference	19
6.5.1 Detailed Description	19
6.6 com.example.backend.configuration.AppConfiguration Class Reference	19
6.6.1 Detailed Description	20
6.6.2 Member Function Documentation	20
6.6.2.1 addCorsMappings()	20
6.7 com.example.backend.dto.AttendInfo Class Reference	20
6.7.1 Detailed Description	20
6.8 com.example.backend.BackendApplication Class Reference	20
6.8.1 Detailed Description	20
6.8.2 Member Function Documentation	20
6.8.2.1 main()	21
6.9 com.example.backend.dto.EdgeDto Class Reference	21
6.9.1 Detailed Description	21
6.10 com.example.backend.controller.EmployeeController Class Reference	21
6.10.1 Detailed Description	21
6.10.2 Constructor & Destructor Documentation	21
6.10.2.1 EmployeeController()	22
6.10.3 Member Function Documentation	22
6.10.3.1 addEmployee()	22
6.10.3.2 checkToken()	22
6.10.3.3 deleteEmployee()	23
6.10.3.4 getAllEmployees() [1/2]	23
6.10.3.5 getAllEmployees() [2/2]	24
6.10.3.6 updateEmployee()	24
6.11 com.example.backend.entity.EmployeeEntity Class Reference	24
6.11.1 Detailed Description	24
6.12 com.example.backend.service.EmployeeService Interface Reference	25
6.12.1 Detailed Description	25
6.12.2 Member Function Documentation	25
6.12.2.1 addEmployee()	25
6.12.2.2 deleteEmployee()	26
6.12.2.3 getAllEmployeeInfo()	26

6.12.2.4	getOneEmployeeInfo()	26
6.12.2.5	updateEmployee()	27
6.13	com.example.backend.service.impl.EmployeeServiceImpl Class Reference	27
6.13.1	Detailed Description	28
6.13.2	Member Function Documentation	28
6.13.2.1	addEmployee()	28
6.13.2.2	deleteEmployee()	29
6.13.2.3	getAllEmployeeInfo()	29
6.13.2.4	getOneEmployeeInfo()	29
6.13.2.5	updateEmployee()	29
6.14	com.example.backend.dto.GraphDto Class Reference	30
6.14.1	Detailed Description	30
6.15	com.example.backend.controller.KnowledgeGraphController Class Reference	30
6.15.1	Detailed Description	30
6.15.2	Member Function Documentation	30
6.15.2.1	getKnoeledgeGraph()	31
6.16	com.example.backend.controller.LogController Class Reference	31
6.16.1	Detailed Description	32
6.16.2	Member Function Documentation	32
6.16.2.1	getLogs()	32
6.16.2.2	getLogsByNode()	32
6.17	com.example.backend.controller.MetricController Class Reference	32
6.17.1	Detailed Description	33
6.17.2	Member Function Documentation	33
6.17.2.1	getMetrics()	33
6.17.2.2	getMetricsByNode()	33
6.18	com.example.backend.dto.MyToken Class Reference	33
6.18.1	Detailed Description	33
6.19	com.example.backend.dto.NodeDto Class Reference	34
6.19.1	Detailed Description	34
6.19.2	Member Function Documentation	34
6.19.2.1	getId()	34
6.19.2.2	getLabel()	35
6.19.2.3	getName()	35
6.19.2.4	getNamespace()	35
6.19.2.5	getPod_node()	36
6.19.2.6	getProperty()	36
6.19.2.7	setId()	36
6.19.2.8	setLabel()	37
6.19.2.9	setName()	37
6.19.2.10	setNamespace()	37
6.19.2.11	setPod_node()	38

6.19.2.12 setProperty()	38
6.20 com.example.backend.dto.OneEmployeeInDto Class Reference	38
6.20.1 Detailed Description	38
6.21 com.example.backend.dto.OneEmployeeOutDto Class Reference	39
6.21.1 Detailed Description	39
6.22 com.example.backend.dto.PayrollInfo Class Reference	39
6.22.1 Detailed Description	39
6.23 com.example.backend.dto.PrizeInfo Class Reference	39
6.23.1 Detailed Description	39
6.24 com.example.backend.dto.TokenInfo Class Reference	39
6.24.1 Detailed Description	39
7 File Documentation	41
7.1 src/main/java/com/example/backend/BackendApplication.java File Reference	41
7.2 BackendApplication.java	41
7.3 src/main/java/com/example/backend/configuration/AppConfiguration.java File Reference	41
7.4 AppConfiguration.java	42
7.5 src/main/java/com/example/backend/controller/AlertController.java File Reference	42
7.6 AlertController.java	42
7.7 src/main/java/com/example/backend/controller/EmployeeController.java File Reference	43
7.8 EmployeeController.java	43
7.9 src/main/java/com/example/backend/controller/KnowledgeGraphController.java File Reference	45
7.10 KnowledgeGraphController.java	45
7.11 src/main/java/com/example/backend/controller/LogController.java File Reference	48
7.12 LogController.java	48
7.13 src/main/java/com/example/backend/controller/MetricController.java File Reference	51
7.14 MetricController.java	51
7.15 src/main/java/com/example/backend/dto/AlertStatusDto.java File Reference	53
7.16 AlertStatusDto.java	54
7.17 src/main/java/com/example/backend/dto/AllEmployeeInfoOutDto.java File Reference	54
7.18 AllEmployeeInfoOutDto.java	54
7.19 src/main/java/com/example/backend/dto/AttendInfo.java File Reference	54
7.20 AttendInfo.java	55
7.21 src/main/java/com/example/backend/dto/EdgeDto.java File Reference	55
7.22 EdgeDto.java	55
7.23 src/main/java/com/example/backend/dto/GraphDto.java File Reference	55
7.24 GraphDto.java	56
7.25 src/main/java/com/example/backend/dto/MyToken.java File Reference	56
7.26 MyToken.java	56
7.27 src/main/java/com/example/backend/dto/NodeDto.java File Reference	56
7.28 NodeDto.java	57
7.29 src/main/java/com/example/backend/dto/OneEmployeeInDto.java File Reference	57

7.30 OneEmployeeInDto.java	57
7.31 src/main/java/com/example/backend/dto/OneEmployeeOutDto.java File Reference	58
7.32 OneEmployeeOutDto.java	58
7.33 src/main/java/com/example/backend/dto/PayrollInfo.java File Reference	58
7.34 PayrollInfo.java	58
7.35 src/main/java/com/example/backend/dto/PrizeInfo.java File Reference	59
7.36 PrizeInfo.java	59
7.37 src/main/java/com/example/backend/dto/TokenInfo.java File Reference	59
7.38 TokenInfo.java	59
7.39 src/main/java/com/example/backend/entity/AlertEntity.java File Reference	60
7.40 AlertEntity.java	60
7.41 src/main/java/com/example/backend/entity/EmployeeEntity.java File Reference	61
7.42 EmployeeEntity.java	61
7.43 src/main/java/com/example/backend/scheduler/AlertScheduler.java File Reference	61
7.44 AlertScheduler.java	62
7.45 src/main/java/com/example/backend/service/EmployeeService.java File Reference	63
7.46 EmployeeService.java	63
7.47 src/main/java/com/example/backend/service/impl/EmployeeServiceImpl.java File Reference	63
7.48 EmployeeServiceImpl.java	64
7.49 src/test/java/com/example/backend/BackendApplicationTests.java File Reference	65
7.50 BackendApplicationTests.java	66
Index	67

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

com.example.backend	9
com.example.backend.configuration	9
com.example.backend.controller	9
com.example.backend.dto	10
com.example.backend.entity	10
com.example.backend.scheduler	10
com.example.backend.service	10
com.example.backend.service.impl	10

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

com.example.backend.controller.AlertController	11
com.example.backend.entity.AlertEntity	12
com.example.backend.scheduler.AlertScheduler	17
com.example.backend.dto.AlertStatusDto	18
com.example.backend.dto.AllEmployeeInfoOutDto	19
com.example.backend.dto.AttendInfo	20
com.example.backend.BackendApplication	20
com.example.backend.dto.EdgeDto	21
com.example.backend.controller.EmployeeController	21
com.example.backend.entity.EmployeeEntity	24
com.example.backend.service.EmployeeService	25
com.example.backend.service.impl.EmployeeServiceImpl	27
com.example.backend.dto.GraphDto	30
com.example.backend.controller.KnowledgeGraphController	30
com.example.backend.controller.LogController	31
com.example.backend.controller.MetricController	32
com.example.backend.dto.MyToken	33
com.example.backend.dto.NodeDto	34
com.example.backend.dto.OneEmployeeInDto	38
com.example.backend.dto.OneEmployeeOutDto	39
com.example.backend.dto.PayrollInfo	39
com.example.backend.dto.PrizeInfo	39
com.example.backend.dto.TokenInfo	39
WebMvcConfigurer	
com.example.backend.configuration.AppConfiguration	19

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

com.example.backend.controller.AlertController	11
com.example.backend.entity.AlertEntity	12
com.example.backend.scheduler.AlertScheduler	17
com.example.backend.dto.AlertStatusDto	18
com.example.backend.dto.AllEmployeeInfoOutDto	19
com.example.backend.configuration.AppConfiguration	19
com.example.backend.dto.AttendInfo	20
com.example.backend.BackendApplication	20
com.example.backend.dto.EdgeDto	21
com.example.backend.controller.EmployeeController	21
com.example.backend.entity.EmployeeEntity	24
com.example.backend.service.EmployeeService	25
com.example.backend.service.impl.EmployeeServiceImpl	27
com.example.backend.dto.GraphDto	30
com.example.backend.controller.KnowledgeGraphController	30
com.example.backend.controller.LogController	31
com.example.backend.controller.MetricController	32
com.example.backend.dto.MyToken	33
com.example.backend.dto.NodeDto	34
com.example.backend.dto.OneEmployeeInDto	38
com.example.backend.dto.OneEmployeeOutDto	39
com.example.backend.dto.PayrollInfo	39
com.example.backend.dto.PrizeInfo	39
com.example.backend.dto.TokenInfo	39

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

src/main/java/com/example/backend/BackendApplication.java	41
src/main/java/com/example/backend/configuration/AppConfiguration.java	41
src/main/java/com/example/backend/controller/AlertController.java	42
src/main/java/com/example/backend/controller/EmployeeController.java	43
src/main/java/com/example/backend/controller/KnowledgeGraphController.java	45
src/main/java/com/example/backend/controller/LogController.java	48
src/main/java/com/example/backend/controller/MetricController.java	51
src/main/java/com/example/backend/dto/AlertStatusDto.java	53
src/main/java/com/example/backend/dto/AllEmployeeInfoOutDto.java	54
src/main/java/com/example/backend/dto/AttendInfo.java	54
src/main/java/com/example/backend/dto/EdgeDto.java	55
src/main/java/com/example/backend/dto/GraphDto.java	55
src/main/java/com/example/backend/dto/MyToken.java	56
src/main/java/com/example/backend/dto/NodeDto.java	56
src/main/java/com/example/backend/dto/OneEmployeeInDto.java	57
src/main/java/com/example/backend/dto/OneEmployeeOutDto.java	58
src/main/java/com/example/backend/dto/PayrollInfo.java	58
src/main/java/com/example/backend/dto/PrizeInfo.java	59
src/main/java/com/example/backend/dto/TokenInfo.java	59
src/main/java/com/example/backend/entity/AlertEntity.java	60
src/main/java/com/example/backend/entity/EmployeeEntity.java	61
src/main/java/com/example/backend/scheduler/AlertScheduler.java	61
src/main/java/com/example/backend/service/EmployeeService.java	63
src/main/java/com/example/backend/service/impl/EmployeeServiceImpl.java	63
src/test/java/com/example/backend/BackendApplicationTests.java	65

Chapter 5

Namespace Documentation

5.1 Package com.example.backend

Packages

- package [configuration](#)
- package [controller](#)
- package [dto](#)
- package [entity](#)
- package [scheduler](#)
- package [service](#)

Classes

- class [BackendApplication](#)
- class **BackendApplicationTests**

5.2 Package com.example.backend.configuration

Classes

- class [AppConfiguration](#)

5.3 Package com.example.backend.controller

Classes

- class [AlertController](#)
- class [EmployeeController](#)
- class [KnowledgeGraphController](#)
- class [LogController](#)
- class [MetricController](#)

5.4 Package com.example.backend.dto

Classes

- class [AlertStatusDto](#)
- class [AllEmployeeInfoOutDto](#)
- class [AttendInfo](#)
- class [EdgeDto](#)
- class [GraphDto](#)
- class [MyToken](#)
- class [NodeDto](#)
- class [OneEmployeeInDto](#)
- class [OneEmployeeOutDto](#)
- class [PayrollInfo](#)
- class [PrizeInfo](#)
- class [TokenInfo](#)

5.5 Package com.example.backend.entity

Classes

- class [AlertEntity](#)
- class [EmployeeEntity](#)

5.6 Package com.example.backend.scheduler

Classes

- class [AlertScheduler](#)

5.7 Package com.example.backend.service

Packages

- package [impl](#)

Classes

- interface [EmployeeService](#)

5.8 Package com.example.backend.service.impl

Classes

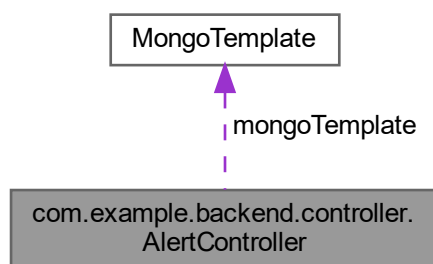
- class [EmployeeServiceImpl](#)

Chapter 6

Class Documentation

6.1 com.example.backend.controller.AlertController Class Reference

Collaboration diagram for com.example.backend.controller.AlertController:



Public Member Functions

- `ResponseEntity< String > getAlerts (@RequestParam(required=false, defaultValue="0") Long start_time, @RequestParam(required=false, defaultValue="999999999") Long end_time) throws IOException`
- `ResponseEntity postAlerts (@RequestBody AlertStatusDto body)`

6.1.1 Detailed Description

Definition at line [26](#) of file [AlertController.java](#).

6.1.2 Member Function Documentation

6.1.2.1 getAlerts()

```
ResponseEntity< String > com.example.backend.controller.AlertController.getAlerts (
    @RequestParam(required=false, defaultValue="0") Long start_time,
    @RequestParam(required=false, defaultValue="999999999") Long end_time ) throws
IOException
```

Definition at line 32 of file [AlertController.java](#).

6.1.2.2 postAlerts()

```
ResponseEntity com.example.backend.controller.AlertController.postAlerts (
    @RequestBody AlertStatusDto body )
```

Definition at line 47 of file [AlertController.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/controller/AlertController.java](#)

6.2 com.example.backend.entity.AlertEntity Class Reference

Public Member Functions

- Long [get_time](#) ()
- void [set_time](#) (Long time)
- String [get_entity_id](#) ()
- void [set_entity_id](#) (String entity_id)
- String [get_pod](#) ()
- void [set_pod](#) (String pod)
- String [get_category](#) ()
- void [set_category](#) (String category)
- String [get_description](#) ()
- void [set_description](#) (String description)
- String [get_status](#) ()
- void [set_status](#) (String status)
- String [get_entity_name](#) ()
- void [set_entity_name](#) (String entity_name)
- String [get_id](#) ()
- void [set_id](#) (String id)

6.2.1 Detailed Description

Definition at line 7 of file [AlertEntity.java](#).

6.2.2 Member Function Documentation

6.2.2.1 get_category()

```
String com.example.backend.entity.AlertEntity.get_category ( )
```

Definition at line 40 of file [AlertEntity.java](#).

6.2.2.2 get_description()

```
String com.example.backend.entity.AlertEntity.get_description ( )
```

Definition at line 44 of file [AlertEntity.java](#).

6.2.2.3 get_entity_id()

```
String com.example.backend.entity.AlertEntity.get_entity_id ( )
```

Definition at line 26 of file [AlertEntity.java](#).

6.2.2.4 get_entity_name()

```
String com.example.backend.entity.AlertEntity.get_entity_name ( )
```

Definition at line 52 of file [AlertEntity.java](#).

6.2.2.5 get_id()

```
String com.example.backend.entity.AlertEntity.get_id ( )
```

Definition at line 56 of file [AlertEntity.java](#).

6.2.2.6 get_pod()

```
String com.example.backend.entity.AlertEntity.get_pod ( )
```

Definition at line 32 of file [AlertEntity.java](#).

6.2.2.7 get_status()

```
String com.example.backend.entity.AlertEntity.get_status ( )
```

Definition at line 48 of file [AlertEntity.java](#).

6.2.2.8 get_time()

```
Long com.example.backend.entity.AlertEntity.get_time ( )
```

Definition at line 18 of file [AlertEntity.java](#).

6.2.2.9 set_category()

```
void com.example.backend.entity.AlertEntity.set_category (
    String category )
```

Definition at line 42 of file [AlertEntity.java](#).

Here is the caller graph for this function:

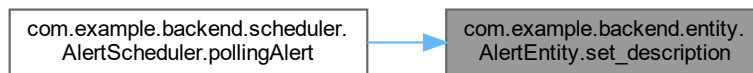


6.2.2.10 set_description()

```
void com.example.backend.entity.AlertEntity.set_description (
    String description )
```

Definition at line 46 of file [AlertEntity.java](#).

Here is the caller graph for this function:

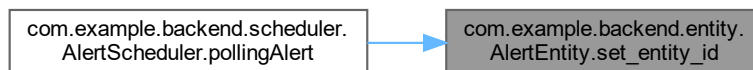


6.2.2.11 set_entity_id()

```
void com.example.backend.entity.AlertEntity.set_entity_id (
    String entity_id )
```

Definition at line 30 of file [AlertEntity.java](#).

Here is the caller graph for this function:



6.2.2.12 set_entity_name()

```
void com.example.backend.entity.AlertEntity.set_entity_name (
    String entity_name )
```

Definition at line 54 of file [AlertEntity.java](#).

Here is the caller graph for this function:



6.2.2.13 set_id()

```
void com.example.backend.entity.AlertEntity.set_id (
    String id )
```

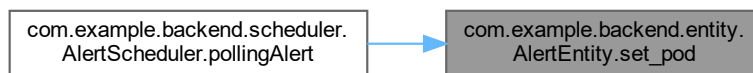
Definition at line 58 of file [AlertEntity.java](#).

6.2.2.14 set_pod()

```
void com.example.backend.entity.AlertEntity.set_pod (
    String pod )
```

Definition at line 36 of file [AlertEntity.java](#).

Here is the caller graph for this function:

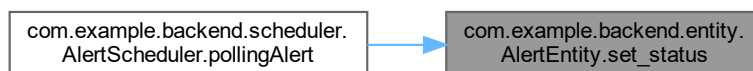


6.2.2.15 set_status()

```
void com.example.backend.entity.AlertEntity.set_status (
    String status )
```

Definition at line 50 of file [AlertEntity.java](#).

Here is the caller graph for this function:



6.2.2.16 set_time()

```
void com.example.backend.entity.AlertEntity.set_time (
    Long time )
```

Definition at line 22 of file [AlertEntity.java](#).

Here is the caller graph for this function:

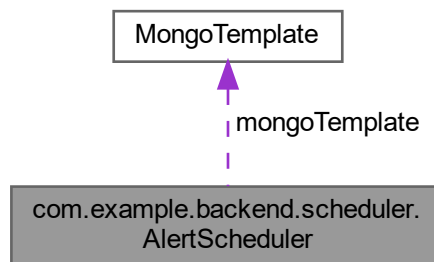


The documentation for this class was generated from the following file:

- `src/main/java/com/example/backend/entity/AlertEntity.java`

6.3 com.example.backend.scheduler.AlertScheduler Class Reference

Collaboration diagram for `com.example.backend.scheduler.AlertScheduler`:



Public Member Functions

- void [pollingAlert](#) () throws IOException

6.3.1 Detailed Description

Definition at line 21 of file [AlertScheduler.java](#).

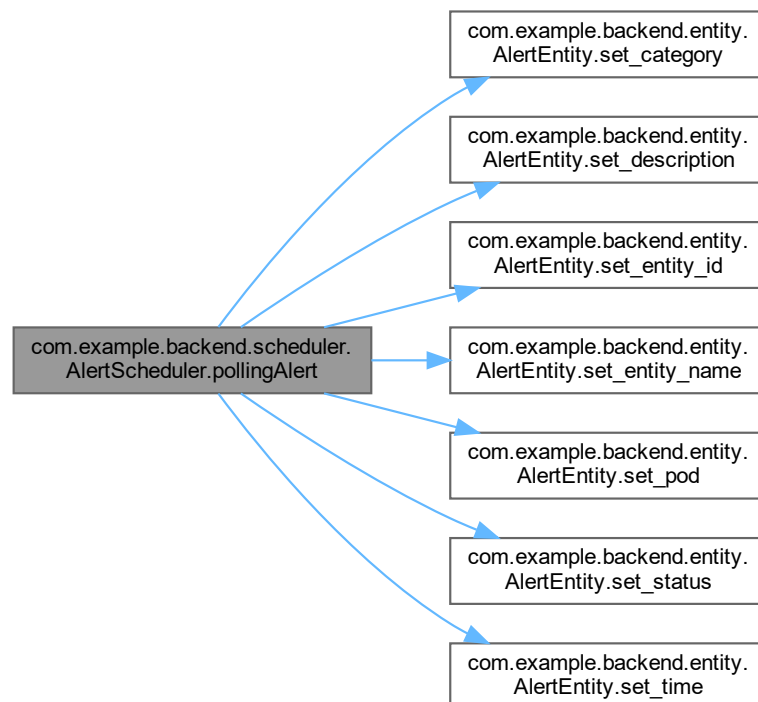
6.3.2 Member Function Documentation

6.3.2.1 pollingAlert()

```
void com.example.backend.scheduler.AlertScheduler.pollingAlert ( ) throws IOException
```

Definition at line 29 of file [AlertScheduler.java](#).

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/scheduler/AlertScheduler.java](#)

6.4 com.example.backend.dto.AlertStatusDto Class Reference

6.4.1 Detailed Description

Definition at line 6 of file [AlertStatusDto.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/AlertStatusDto.java](#)

6.5 com.example.backend.dto.AllEmployeeInfoOutDto Class Reference

6.5.1 Detailed Description

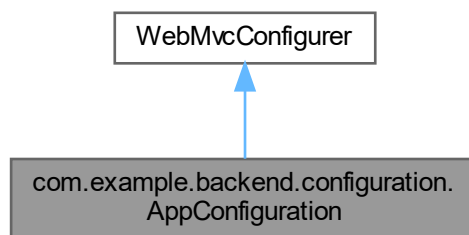
Definition at line 5 of file [AllEmployeeInfoOutDto.java](#).

The documentation for this class was generated from the following file:

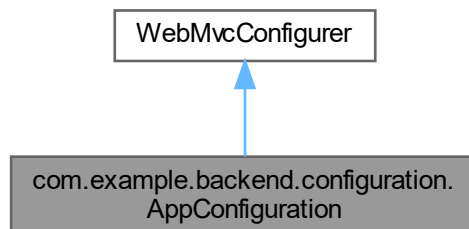
- [src/main/java/com/example/backend/dto/AllEmployeeInfoOutDto.java](#)

6.6 com.example.backend.configuration.AppConfiguration Class Reference

Inheritance diagram for com.example.backend.configuration.AppConfiguration:



Collaboration diagram for com.example.backend.configuration.AppConfiguration:



Public Member Functions

- void [addCorsMappings](#) (CorsRegistry registry)

6.6.1 Detailed Description

Definition at line 9 of file [AppConfiguration.java](#).

6.6.2 Member Function Documentation

6.6.2.1 addCorsMappings()

```
void com.example.backend.configuration.AppConfiguration.addCorsMappings (
    CorsRegistry registry )
```

Definition at line 11 of file [AppConfiguration.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/configuration/AppConfiguration.java](#)

6.7 com.example.backend.dto.AttendInfo Class Reference

6.7.1 Detailed Description

Definition at line 6 of file [AttendInfo.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/AttendInfo.java](#)

6.8 com.example.backend.BackendApplication Class Reference

Static Public Member Functions

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

6.8.1 Detailed Description

Definition at line 9 of file [BackendApplication.java](#).

6.8.2 Member Function Documentation

6.8.2.1 main()

```
static void com.example.backend.BackendApplication.main (
    String[] args ) [static]
```

Definition at line 11 of file [BackendApplication.java](#).

The documentation for this class was generated from the following file:

- src/main/java/com/example/backend/[BackendApplication.java](#)

6.9 com.example.backend.dto.EdgeDto Class Reference

6.9.1 Detailed Description

Definition at line 6 of file [EdgeDto.java](#).

The documentation for this class was generated from the following file:

- src/main/java/com/example/backend/dto/[EdgeDto.java](#)

6.10 com.example.backend.controller.EmployeeController Class Reference

Public Member Functions

- [EmployeeController](#) ([EmployeeService](#) employeeService)
- ResponseEntity< [TokenInfo](#) > [checkToken](#) (String token) throws IOException
- ResponseEntity< List< [AllEmployeeInfoOutDto](#) > > [getAllEmployees](#) (@RequestParam(value="token", required=false) String token) throws IOException
- ResponseEntity< [OneEmployeeOutDto](#) > [getAllEmployees](#) (@RequestParam(value="id", required=false) String id, @RequestParam(value="token", required=false) String token) throws IOException
- ResponseEntity [addEmployee](#) (@RequestBody [OneEmployeeInDto](#) employeeDto)
- ResponseEntity [updateEmployee](#) (@RequestBody [OneEmployeeInDto](#) employeeDto)
- ResponseEntity [deleteEmployee](#) (@RequestParam(required=true) String id, @RequestParam(required=false) String token) throws IOException

6.10.1 Detailed Description

Definition at line 20 of file [EmployeeController.java](#).

6.10.2 Constructor & Destructor Documentation

6.10.2.1 EmployeeController()

```
com.example.backend.controller.EmployeeController.EmployeeController (
    EmployeeService employeeService )
```

Definition at line 27 of file [EmployeeController.java](#).

6.10.3 Member Function Documentation

6.10.3.1 addEmployee()

```
ResponseEntity com.example.backend.controller.EmployeeController.addEmployee (
    @RequestBody OneEmployeeInDto employeeDto )
```

Definition at line 99 of file [EmployeeController.java](#).

Here is the call graph for this function:

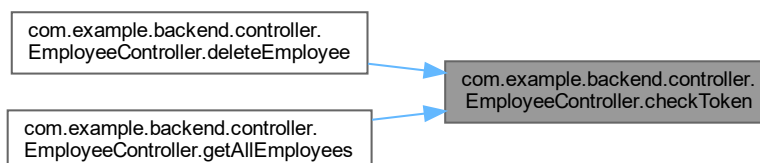


6.10.3.2 checkToken()

```
ResponseEntity< TokenInfo > com.example.backend.controller.EmployeeController.checkToken (
    String token ) throws IOException
```

Definition at line 31 of file [EmployeeController.java](#).

Here is the caller graph for this function:

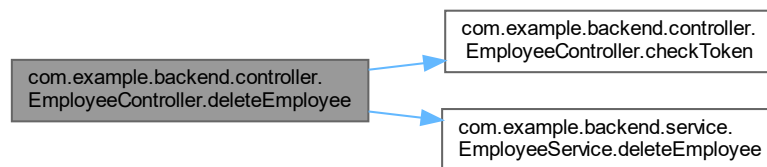


6.10.3.3 deleteEmployee()

```
ResponseEntity com.example.backend.controller.EmployeeController.deleteEmployee (
    @RequestParam(required=true) String id,
    @RequestParam(required=false) String token ) throws IOException
```

Definition at line 121 of file [EmployeeController.java](#).

Here is the call graph for this function:

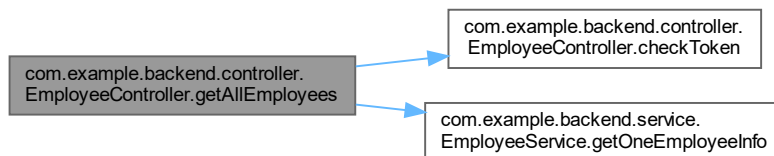


6.10.3.4 getAllEmployees() [1/2]

```
ResponseEntity< OneEmployeeOutDto > com.example.backend.controller.EmployeeController.getAllEmployees (
    @RequestParam(value="id", required=false) String id,
    @RequestParam(value="token", required=false) String token ) throws IOException
```

Definition at line 79 of file [EmployeeController.java](#).

Here is the call graph for this function:

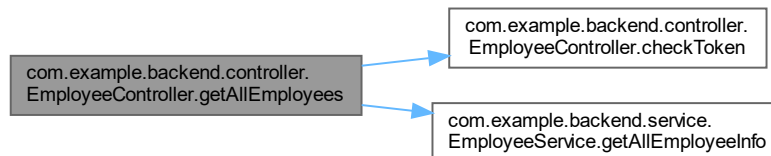


6.10.3.5 getAllEmployees() [2/2]

```
ResponseEntity< List< AllEmployeeInfoOutDto > > com.example.backend.controller.EmployeeController.getAllEmployees (
    @RequestParam(value="token", required=false) String token ) throws IOException
```

Definition at line 63 of file [EmployeeController.java](#).

Here is the call graph for this function:



6.10.3.6 updateEmployee()

```
ResponseEntity com.example.backend.controller.EmployeeController.updateEmployee (
    @RequestBody OneEmployeeInDto employeeDto )
```

Definition at line 110 of file [EmployeeController.java](#).

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/controller/EmployeeController.java](#)

6.11 com.example.backend.entity.EmployeeEntity Class Reference

6.11.1 Detailed Description

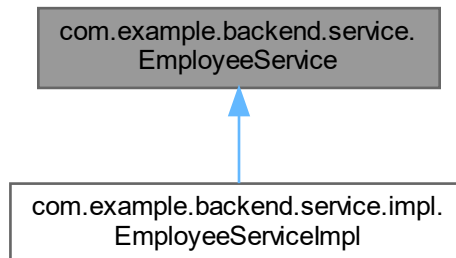
Definition at line 9 of file [EmployeeEntity.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/entity/EmployeeEntity.java](#)

6.12 com.example.backend.service.EmployeeService Interface Reference

Inheritance diagram for com.example.backend.service.EmployeeService:



Public Member Functions

- List< [AllEmployeeInfoOutDto](#) > [getAllEmployeeInfo](#) ()
- boolean [addEmployee](#) ([OneEmployeeInDto](#) employee)
- boolean [updateEmployee](#) ([OneEmployeeInDto](#) employee)
- boolean [deleteEmployee](#) (String id)
- [OneEmployeeOutDto](#) [getOneEmployeeInfo](#) (String id)

6.12.1 Detailed Description

Definition at line 10 of file [EmployeeService.java](#).

6.12.2 Member Function Documentation

6.12.2.1 addEmployee()

```
boolean com.example.backend.service.EmployeeService.addEmployee (  
    OneEmployeeInDto employee )
```

Implemented in [com.example.backend.service.impl.EmployeeServiceImpl](#).

Here is the caller graph for this function:



6.12.2.2 deleteEmployee()

```
boolean com.example.backend.service.EmployeeService.deleteEmployee (
    String id )
```

Implemented in [com.example.backend.service.impl.EmployeeServiceImpl](#).

Here is the caller graph for this function:



6.12.2.3 getAllEmployeeInfo()

```
List< AllEmployeeInfoOutDto > com.example.backend.service.EmployeeService.getAllEmployeeInfo (
)
```

Implemented in [com.example.backend.service.impl.EmployeeServiceImpl](#).

Here is the caller graph for this function:



6.12.2.4 getOneEmployeeInfo()

```
OneEmployeeOutDto com.example.backend.service.EmployeeService.getOneEmployeeInfo (
    String id )
```

Implemented in [com.example.backend.service.impl.EmployeeServiceImpl](#).

Here is the caller graph for this function:



6.12.2.5 updateEmployee()

```
boolean com.example.backend.service.EmployeeService.updateEmployee (
    OneEmployeeInDto employee )
```

Implemented in [com.example.backend.service.impl.EmployeeServiceImpl](#).

Here is the caller graph for this function:

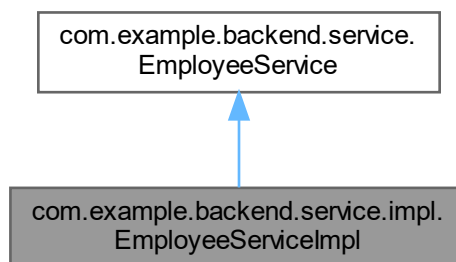


The documentation for this interface was generated from the following file:

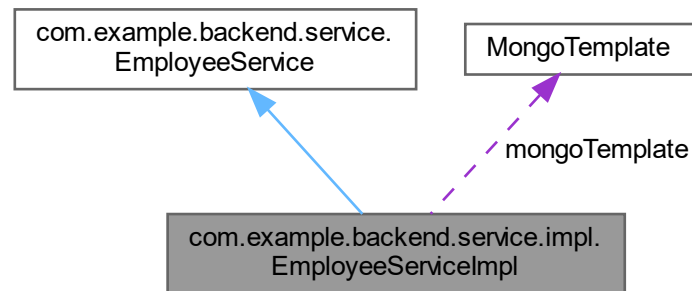
- [src/main/java/com/example/backend/service/EmployeeService.java](#)

6.13 com.example.backend.service.impl.EmployeeServiceImpl Class Reference

Inheritance diagram for `com.example.backend.service.impl.EmployeeServiceImpl`:



Collaboration diagram for `com.example.backend.service.impl.EmployeeServiceImpl`:



Public Member Functions

- `List< AllEmployeeInfoOutDto > getAllEmployeeInfo ()`
 - `boolean addEmployee (OneEmployeeInDto employeeDto)`
 - `boolean updateEmployee (OneEmployeeInDto employeeDto)`
 - `boolean deleteEmployee (String id)`
 - `OneEmployeeOutDto getOneEmployeeInfo (String id)`
-
- `List< AllEmployeeInfoOutDto > getAllEmployeeInfo ()`
 - `boolean addEmployee (OneEmployeeInDto employee)`
 - `boolean updateEmployee (OneEmployeeInDto employee)`
 - `boolean deleteEmployee (String id)`
 - `OneEmployeeOutDto getOneEmployeeInfo (String id)`

6.13.1 Detailed Description

Definition at line 26 of file [EmployeeServiceImpl.java](#).

6.13.2 Member Function Documentation

6.13.2.1 addEmployee()

```
boolean com.example.backend.service.impl.EmployeeServiceImpl.addEmployee (
    OneEmployeeInDto employeeDto )
```

Implements [com.example.backend.service.EmployeeService](#).

Definition at line 51 of file [EmployeeServiceImpl.java](#).

6.13.2.2 deleteEmployee()

```
boolean com.example.backend.service.impl.EmployeeServiceImpl.deleteEmployee (
    String id )
```

Implements [com.example.backend.service.EmployeeService](#).

Definition at line 104 of file [EmployeeServiceImpl.java](#).

6.13.2.3 getAllEmployeeInfo()

```
List< AllEmployeeInfoOutDto > com.example.backend.service.impl.EmployeeServiceImpl.getAllEmployeeInfo ( )
```

Implements [com.example.backend.service.EmployeeService](#).

Definition at line 30 of file [EmployeeServiceImpl.java](#).

6.13.2.4 getOneEmployeeInfo()

```
OneEmployeeOutDto com.example.backend.service.impl.EmployeeServiceImpl.getOneEmployeeInfo (
    String id )
```

Implements [com.example.backend.service.EmployeeService](#).

Definition at line 117 of file [EmployeeServiceImpl.java](#).

6.13.2.5 updateEmployee()

```
boolean com.example.backend.service.impl.EmployeeServiceImpl.updateEmployee (
    OneEmployeeInDto employeeDto )
```

Implements [com.example.backend.service.EmployeeService](#).

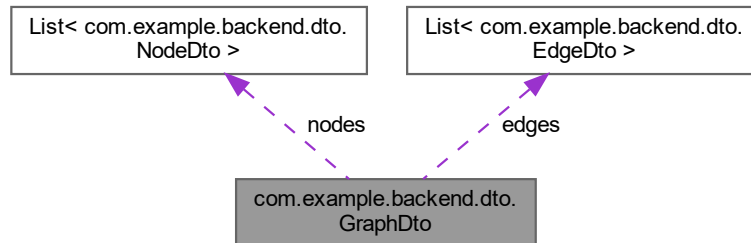
Definition at line 74 of file [EmployeeServiceImpl.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/service/impl/EmployeeServiceImpl.java](#)

6.14 com.example.backend.dto.GraphDto Class Reference

Collaboration diagram for com.example.backend.dto.GraphDto:



6.14.1 Detailed Description

Definition at line 9 of file [GraphDto.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/GraphDto.java](#)

6.15 com.example.backend.controller.KnowledgeGraphController Class Reference

Public Member Functions

- `ResponseEntity< String > getKnoeledgeGraph (@RequestParam(required=false, defaultValue="0") String time_start, @RequestParam(required=false, defaultValue="999999999") String time_end)`

6.15.1 Detailed Description

Definition at line 23 of file [KnowledgeGraphController.java](#).

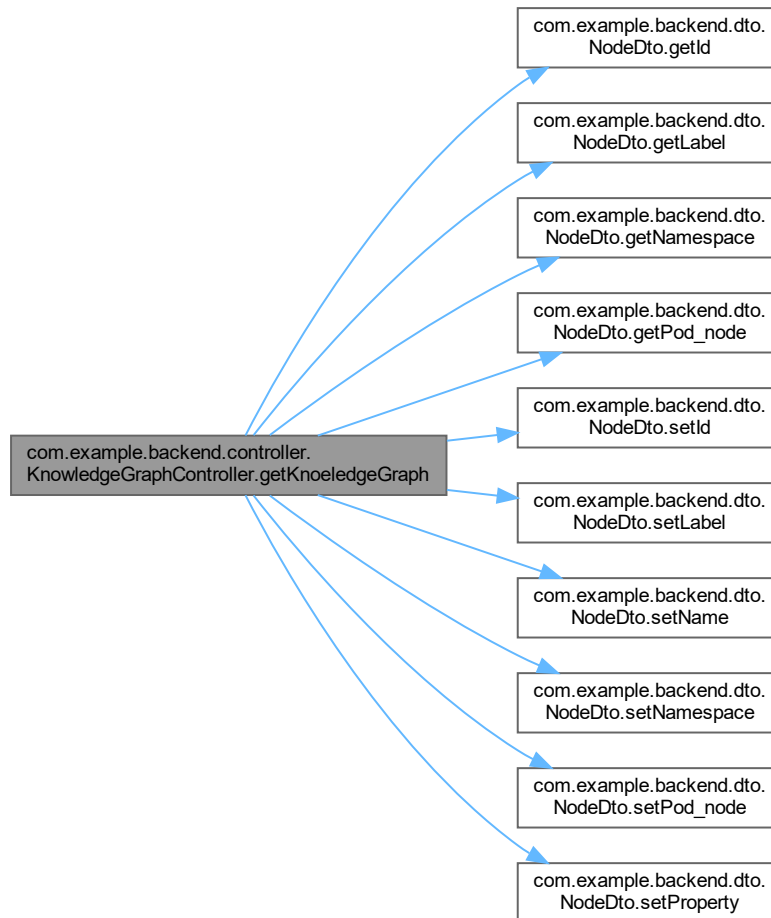
6.15.2 Member Function Documentation

6.15.2.1 getKnoeledgeGraph()

```
ResponseEntity< String > com.example.backend.controller.KnowledgeGraphController.getKnoeledgeGraph (
    @RequestParam(required=false, defaultValue="0") String time_start,
    @RequestParam(required=false, defaultValue="999999999") String time_end )
```

Definition at line 28 of file [KnowledgeGraphController.java](#).

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/controller/KnowledgeGraphController.java](#)

6.16 com.example.backend.controller.LogController Class Reference

Public Member Functions

- `ResponseEntity< String > getLogs (@RequestParam(required=false) String dataset, @RequestParam(required=false) String model, @RequestParam(required=false, defaultValue="0") Long time_start, @RequestParam(required=false, defaultValue="2147483647000") Long time_end) throws IOException`

- `ResponseEntity< String > getLogsByNode (@RequestParam(required=false, defaultValue="0") Long time_start, @RequestParam(required=false, defaultValue="2147483647000") Long time_end, @RequestParam(required=false) String entity_id) throws IOException`

6.16.1 Detailed Description

Definition at line 29 of file [LogController.java](#).

6.16.2 Member Function Documentation

6.16.2.1 getLogs()

```
ResponseEntity< String > com.example.backend.controller.LogController.getLogs (
    @RequestParam(required=false) String dataset,
    @RequestParam(required=false) String model,
    @RequestParam(required=false, defaultValue="0") Long time_start,
    @RequestParam(required=false, defaultValue="2147483647000") Long time_end ) throws
IOException
```

Definition at line 34 of file [LogController.java](#).

6.16.2.2 getLogsByNode()

```
ResponseEntity< String > com.example.backend.controller.LogController.getLogsByNode (
    @RequestParam(required=false, defaultValue="0") Long time_start,
    @RequestParam(required=false, defaultValue="2147483647000") Long time_end,
    @RequestParam(required=false) String entity_id ) throws IOException
```

Definition at line 139 of file [LogController.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/controller/LogController.java](#)

6.17 com.example.backend.controller.MetricController Class Reference

Public Member Functions

- `ResponseEntity< String > getMetrics (@RequestParam(required=false) String dataset, @RequestParam(required=false) String model, @RequestParam(required=false, defaultValue="0") Long time_start, @RequestParam(required=false, defaultValue="999999999") Long time_end) throws IOException`
- `ResponseEntity< String > getMetricsByNode (@RequestParam(required=false, defaultValue="0") Long time_start, @RequestParam(required=false, defaultValue="999999999") Long time_end, @RequestParam(required=false) String entity_id) throws IOException`

6.17.1 Detailed Description

Definition at line 21 of file [MetricController.java](#).

6.17.2 Member Function Documentation

6.17.2.1 getMetrics()

```
ResponseEntity< String > com.example.backend.controller.MetricController.getMetrics (
    @RequestParam(required=false) String dataset,
    @RequestParam(required=false) String model,
    @RequestParam(required=false, defaultValue="0") Long time_start,
    @RequestParam(required=false, defaultValue="999999999") Long time_end ) throws
IOException
```

Definition at line 26 of file [MetricController.java](#).

6.17.2.2 getMetricsByNode()

```
ResponseEntity< String > com.example.backend.controller.MetricController.getMetricsByNode (
    @RequestParam(required=false, defaultValue="0") Long time_start,
    @RequestParam(required=false, defaultValue="999999999") Long time_end,
    @RequestParam(required=false) String entity_id ) throws IOException
```

Definition at line 114 of file [MetricController.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/controller/MetricController.java](#)

6.18 com.example.backend.dto.MyToken Class Reference

6.18.1 Detailed Description

Definition at line 8 of file [MyToken.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/MyToken.java](#)

6.19 com.example.backend.dto.NodeDto Class Reference

Public Member Functions

- String [getId](#) ()
- void [setId](#) (String id)
- String [getLabel](#) ()
- void [setLabel](#) (String label)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getProperty](#) ()
- void [setProperty](#) (String property)
- String [getPod_node](#) ()
- void [setPod_node](#) (String pod_node)
- String [getNamespace](#) ()
- void [setNamespace](#) (String namespace)

6.19.1 Detailed Description

Definition at line 6 of file [NodeDto.java](#).

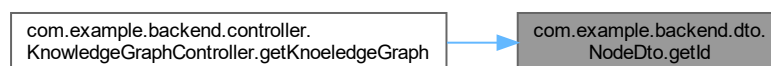
6.19.2 Member Function Documentation

6.19.2.1 getId()

```
String com.example.backend.dto.NodeDto.getId ( )
```

Definition at line 16 of file [NodeDto.java](#).

Here is the caller graph for this function:



6.19.2.2 getLabel()

```
String com.example.backend.dto.NodeDto.getLabel ( )
```

Definition at line 19 of file [NodeDto.java](#).

Here is the caller graph for this function:



6.19.2.3 getName()

```
String com.example.backend.dto.NodeDto.getName ( )
```

Definition at line 22 of file [NodeDto.java](#).

6.19.2.4 getNamespace()

```
String com.example.backend.dto.NodeDto.getNamespace ( )
```

Definition at line 31 of file [NodeDto.java](#).

Here is the caller graph for this function:

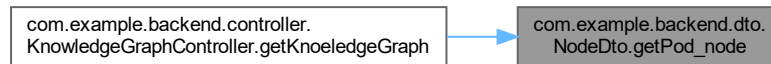


6.19.2.5 getPod_node()

```
String com.example.backend.dto.NodeDto.getPod_node ( )
```

Definition at line 28 of file [NodeDto.java](#).

Here is the caller graph for this function:



6.19.2.6 getProperty()

```
String com.example.backend.dto.NodeDto.getProperty ( )
```

Definition at line 25 of file [NodeDto.java](#).

6.19.2.7 setId()

```
void com.example.backend.dto.NodeDto.setId (
    String id )
```

Definition at line 17 of file [NodeDto.java](#).

Here is the caller graph for this function:

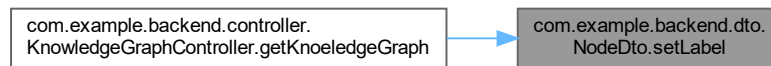


6.19.2.8 setLabel()

```
void com.example.backend.dto.NodeDto.setLabel (
    String label )
```

Definition at line 20 of file [NodeDto.java](#).

Here is the caller graph for this function:

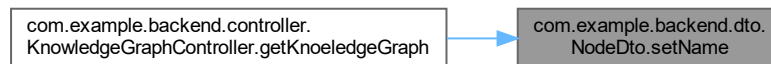


6.19.2.9 setName()

```
void com.example.backend.dto.NodeDto.setName (
    String name )
```

Definition at line 23 of file [NodeDto.java](#).

Here is the caller graph for this function:



6.19.2.10 setNamespace()

```
void com.example.backend.dto.NodeDto.setNamespace (
    String namespace )
```

Definition at line 32 of file [NodeDto.java](#).

Here is the caller graph for this function:

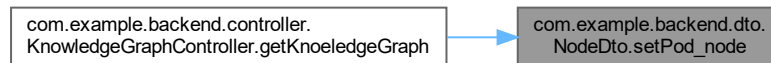


6.19.2.11 setPod_node()

```
void com.example.backend.dto.NodeDto.setPod_node (
    String pod_node )
```

Definition at line 29 of file [NodeDto.java](#).

Here is the caller graph for this function:

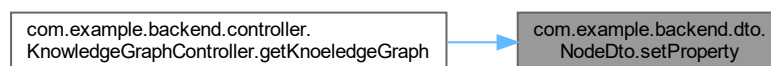


6.19.2.12 setProperty()

```
void com.example.backend.dto.NodeDto.setProperty (
    String property )
```

Definition at line 26 of file [NodeDto.java](#).

Here is the caller graph for this function:



The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/NodeDto.java](#)

6.20 com.example.backend.dto.OneEmployeeInDto Class Reference

6.20.1 Detailed Description

Definition at line 6 of file [OneEmployeeInDto.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/OneEmployeeInDto.java](#)

6.21 com.example.backend.dto.OneEmployeeOutDto Class Reference

6.21.1 Detailed Description

Definition at line 9 of file [OneEmployeeOutDto.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/OneEmployeeOutDto.java](#)

6.22 com.example.backend.dto.PayrollInfo Class Reference

6.22.1 Detailed Description

Definition at line 6 of file [PayrollInfo.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/PayrollInfo.java](#)

6.23 com.example.backend.dto.PrizeInfo Class Reference

6.23.1 Detailed Description

Definition at line 6 of file [PrizeInfo.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/PrizeInfo.java](#)

6.24 com.example.backend.dto.TokenInfo Class Reference

6.24.1 Detailed Description

Definition at line 6 of file [TokenInfo.java](#).

The documentation for this class was generated from the following file:

- [src/main/java/com/example/backend/dto/TokenInfo.java](#)

Chapter 7

File Documentation

7.1 src/main/java/com/example/backend/BackendApplication.java File Reference

Classes

- class [com.example.backend.BackendApplication](#)

Packages

- package [com.example.backend](#)

7.2 BackendApplication.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend;
00002
00003 import org.springframework.boot.SpringApplication;
00004 import org.springframework.boot.autoconfigure.SpringBootApplication;
00005 import org.springframework.scheduling.annotation.EnableScheduling;
00006
00007 @SpringBootApplication
00008 @EnableScheduling
00009 public class BackendApplication {
00010
00011     public static void main(String[] args) {
00012         SpringApplication.run(BackendApplication.class, args);
00013     }
00014
00015 }
```

7.3 src/main/java/com/example/backend/configuration/AppConfiguration.java File Reference

Classes

- class [com.example.backend.configuration.AppConfiguration](#)

Packages

- package [com.example.backend.configuration](#)

7.4 AppConfiguration.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.configuration;
00002
00003 import org.springframework.context.annotation.Bean;
00004 import org.springframework.context.annotation.Configuration;
00005 import org.springframework.web.servlet.config.annotation.CorsRegistry;
00006 import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
00007
00008 @Configuration
00009 public class AppConfiguration implements WebMvcConfigurer {
00010     @Override
00011     public void addCorsMappings(CorsRegistry registry) {
00012         registry.addMapping("/**")
00013             .allowedOriginPatterns("*")
00014             .allowedOrigins("http://localhost:3000")
00015             .allowedMethods("GET", "HEAD", "POST", "PUT", "DELETE", "OPTIONS") //
00016             .allowCredentials(true)
00017             .maxAge(3600)
00018             .allowedHeaders("*");
00019     }
00020 }
```

7.5 src/main/java/com/example/backend/controller/AlertController.java

File Reference

Classes

- class [com.example.backend.controller.AlertController](#)

Packages

- package [com.example.backend.controller](#)

7.6 AlertController.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.controller;
00002
00003 import com.alibaba.fastjson.JSON;
00004 import com.alibaba.fastjson.JSONArray;
00005 import com.alibaba.fastjson.JSONObject;
00006 import com.example.backend.dto.AlertStatusDto;
00007 import com.example.backend.entity.AlertEntity;
00008 import org.apache.catalina.connector.Response;
00009 import org.apache.commons.httpclient.HttpClient;
00010 import org.apache.commons.httpclient.methods.GetMethod;
00011 import org.springframework.beans.factory.annotation.Autowired;
00012 import org.springframework.data.mongodb.core.MongoTemplate;
00013 import org.springframework.data.mongodb.core.query.Criteria;
00014 import org.springframework.data.mongodb.core.query.Query;
00015 import org.springframework.data.mongodb.core.query.Update;
00016 import org.springframework.http.HttpStatus;
00017 import org.springframework.http.ResponseEntity;
00018 import org.springframework.web.bind.annotation.*;
00019
00020 import java.io.IOException;
```

```

00021 import java.util.ArrayList;
00022 import java.util.List;
00023
00024
00025 @RestController
00026 public class AlertController {
00027     @Autowired
00028     MongoTemplate mongoTemplate;
00030
00031     @RequestMapping(value = "alarms", method = RequestMethod.GET)
00032     public ResponseEntity<String> getAlerts(
00033         @RequestParam(required = false, defaultValue = "0") Long start_time,
00034         @RequestParam(required = false, defaultValue = "999999999") Long end_time)
00035         throws IOException
00036     {
00037         Query query = Query.query(Criteria.where("time")
00038             .lte(end_time)
00039             .gte(start_time)
00040         );
00041
00042         List<AlertEntity> resultList = mongoTemplate.find(query, AlertEntity.class);
00043         return new ResponseEntity<String>(JSON.toJSONString(resultList), HttpStatus.OK);
00044     }
00045
00046     @RequestMapping(value = "alarm", method = RequestMethod.POST)
00047     public ResponseEntity postAlerts(
00048         @RequestBody AlertStatusDto body
00049     )
00050     {
00051         Query query = Query.query(Criteria.where("id")
00052             .is(body.getId())
00053         );
00054
00055         List<AlertEntity> list = mongoTemplate.find(query, AlertEntity.class);
00056         if (list.size() == 0) {
00057             return new ResponseEntity(HttpStatus.NOT_FOUND);
00058         } else {
00059             Update update = Update.update("status", body.getStatus());
00060             mongoTemplate.updateFirst(query, update, AlertEntity.class);
00061             return new ResponseEntity(HttpStatus.OK);
00062         }
00063     }
00064 }

```

7.7 src/main/java/com/example/backend/controller/EmployeeController.java File Reference

Classes

- class [com.example.backend.controller.EmployeeController](#)

Packages

- package [com.example.backend.controller](#)

7.8 EmployeeController.java

[Go to the documentation of this file.](#)

```

00001 package com.example.backend.controller;
00002
00003 import com.alibaba.fastjson.JSONObject;
00004 import com.example.backend.dto.*;
00005 import com.example.backend.service.EmployeeService;
00006 import org.apache.commons.httpclient.HttpClient;
00007 import org.apache.commons.httpclient.methods.PostMethod;
00008 import org.modelmapper.ModelMapper;
00009 import org.springframework.beans.factory.annotation.Autowired;
00010 import org.springframework.beans.factory.annotation.Value;

```

```

00011 import org.springframework.http.HttpStatus;
00012 import org.springframework.http.ResponseEntity;
00013 import org.springframework.web.bind.annotation.*;
00014
00015 import java.awt.*;
00016 import java.io.IOException;
00017 import java.util.List;
00018
00019 @RestController
00020 public class EmployeeController {
00021     private String baseUrl = "http://39.98.47.69:8000/";
00022     private String authorization = "Basic
00023     ODAwZTA2ODg5NGFiZjVhMmQ0MTE6Y2ZmODhmYWJlMGJiMDY3Y2I1NmM2OGQ4ZmUzZjk0MWY0ODJmMmU5Yg==";
00024     private HttpClient client = new HttpClient();
00025     private EmployeeService employeeService;
00026
00027     @Autowired
00028     public EmployeeController(EmployeeService employeeService) {
00029         this.employeeService = employeeService;
00030     }
00031
00032     public ResponseEntity<TokenInfo> checkToken(String token) throws IOException {
00033         client.getHttpClientManager().getParams().setConnectionTimeout(15000);
00034         PostMethod postMethod = new PostMethod(baseUrl + "api/login/oauth/introspect");
00035         postMethod.addRequestHeader("Content-type",
00036             "application/x-www-form-urlencoded");
00037         postMethod.addRequestHeader("accept",
00038             "application/json");
00039         postMethod.addRequestHeader("Authorization",
00040             authorization);
00041
00042         postMethod.addParameter("token", token);
00043         postMethod.addParameter("token_type_hint", "access_token");
00044
00045         client.executeMethod(postMethod);
00046         byte[] responseBody = postMethod.getResponseBody();
00047         String s = new String(responseBody);
00048
00049         JSONObject response = JSONObject.parseObject(s);
00050
00051         ModelMapper modelMapper = new ModelMapper();
00052         MyToken myToken = modelMapper.map(response, MyToken.class);
00053
00054         TokenInfo info = new TokenInfo();
00055         info.setActive(myToken.isActive());
00056         info.setUsername(myToken.getUsername());
00057         return new ResponseEntity<>(info, HttpStatus.OK);
00058     }
00059
00060     // Get all employee info
00061     @RequestMapping(value = "employees", method = RequestMethod.GET)
00062     public ResponseEntity<List<AllEmployeeInfoOutDto>> getAllEmployees(
00063         @RequestParam(value = "token", required = false) String token
00064     ) throws IOException {
00065         // check token
00066         if (token == null) return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
00067         ResponseEntity<TokenInfo> check = checkToken(token);
00068         if (check.getBody() == null || !check.getBody().getActive())
00069             return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
00070
00071         List<AllEmployeeInfoOutDto> result = employeeService.getAllEmployeeInfo();
00072         if (result.size() == 0) return new ResponseEntity(HttpStatus.NO_CONTENT);
00073         return new ResponseEntity<List<AllEmployeeInfoOutDto>>(result, HttpStatus.OK);
00074     }
00075
00076     // Get one employee info
00077     @RequestMapping(value = "employee", method = RequestMethod.GET)
00078     public ResponseEntity<OneEmployeeOutDto> getOneEmployeeInfo(
00079         @RequestParam(value = "id", required = false) String id,
00080         @RequestParam(value = "token", required = false) String token
00081     ) throws IOException {
00082         // check token
00083         if (token == null) return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
00084         ResponseEntity<TokenInfo> check = checkToken(token);
00085         if (check.getBody() == null || !check.getBody().getActive())
00086             return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
00087
00088         if (id == null) id = check.getBody().getUsername();
00089         OneEmployeeOutDto result = employeeService.getOneEmployeeInfo(id);
00090         if (result == null) {
00091             return new ResponseEntity(HttpStatus.NO_CONTENT);
00092         }
00093         return new ResponseEntity<>(result, HttpStatus.OK);
00094     }
00095 }
00096

```

```

00097 // add a new employee
00098 @RequestMapping(value = "employees", method = RequestMethod.POST)
00099 public ResponseEntity addEmployee(@RequestBody OneEmployeeInDto employeeDto)
00100 {
00101     if (employeeService.addEmployee(employeeDto)) {
00102         return new ResponseEntity(HttpStatus.CREATED);
00103     } else {
00104         return new ResponseEntity(HttpStatus.BAD_REQUEST);
00105     }
00106 }
00107
00108 // update a new employee
00109 @RequestMapping(value = "employees", method = RequestMethod.PUT)
00110 public ResponseEntity updateEmployee(@RequestBody OneEmployeeInDto employeeDto)
00111 {
00112     if (employeeService.updateEmployee(employeeDto)) {
00113         return new ResponseEntity(HttpStatus.OK);
00114     } else {
00115         return new ResponseEntity(HttpStatus.BAD_REQUEST);
00116     }
00117 }
00118
00119 // delete a new employee
00120 @RequestMapping(value = "employees", method = RequestMethod.DELETE)
00121 public ResponseEntity deleteEmployee(
00122     @RequestParam(required = true) String id,
00123     @RequestParam(required = false) String token
00124 ) throws IOException {
00125     // check token
00126     if (token == null) return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
00127     ResponseEntity<TokenInfo> check = checkToken(token);
00128     if (check.getBody() == null || !check.getBody().getActive())
00129         return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
00130
00131     if (employeeService.deleteEmployee(id)) {
00132         return new ResponseEntity(HttpStatus.OK);
00133     } else {
00134         return new ResponseEntity(HttpStatus.BAD_REQUEST);
00135     }
00136 }
00137 }

```

7.9 src/main/java/com/example/backend/controller/KnowledgeGraphController.java File Reference

Classes

- class [com.example.backend.controller.KnowledgeGraphController](#)

Packages

- package [com.example.backend.controller](#)

7.10 KnowledgeGraphController.java

[Go to the documentation of this file.](#)

```

00001 package com.example.backend.controller;
00002
00003 import com.alibaba.fastjson.JSON;
00004 import com.alibaba.fastjson.JSONArray;
00005 import com.alibaba.fastjson.JSONObject;
00006 import com.example.backend.dto.EdgeDto;
00007 import com.example.backend.dto.GraphDto;
00008 import com.example.backend.dto.NodeDto;
00009 import org.apache.commons.httpclient.HttpClient;
00010 import org.apache.commons.httpclient.methods.GetMethod;
00011 import org.slf4j.Logger;
00012 import org.slf4j.LoggerFactory;
00013 import org.springframework.http.HttpStatus;

```

```

00014 import org.springframework.http.ResponseEntity;
00015 import org.springframework.web.bind.annotation.RequestMapping;
00016 import org.springframework.web.bind.annotation.RequestMethod;
00017 import org.springframework.web.bind.annotation.RequestParam;
00018 import org.springframework.web.bind.annotation.RestController;
00019
00020 import java.util.ArrayList;
00021
00022 @RestController
00023 public class KnowledgeGraphController {
00024     private String baseUrl = "http://10.60.38.174:31013/api/v1/query_range?query=";
00025     private final Logger logger = LoggerFactory.getLogger(this.getClass());
00026
00027     @RequestMapping(value = "graph", method = RequestMethod.GET)
00028     public ResponseEntity<String> getKnoeledgeGraph(
00029         @RequestParam(required = false, defaultValue = "0") String time_start,
00030         @RequestParam(required = false, defaultValue = "999999999") String time_end
00031     )
00032     {
00033         ArrayList<String> urls = new ArrayList<>();
00034         urls.add(baseUrl + "kube_node_info" + "&start=" + time_start + "&end=" + time_end +
00035             "&step=300s");
00036         urls.add(baseUrl + "kube_deployment_labels" + "&start=" + time_start + "&end=" + time_end +
00037             "&step=300s");
00038         urls.add(baseUrl + "kube_pod_info" + "&start=" + time_start + "&end=" + time_end +
00039             "&step=300s");
00040         urls.add(baseUrl + "kube_service_info" + "&start=" + time_start + "&end=" + time_end +
00041             "&step=300s");
00042
00043         try {
00044             HttpClient httpClient = new HttpClient();
00045             httpClient.getHttpConnectionManager().getParams().setConnectionTimeout(15000);
00046
00047             GraphDto result = new GraphDto();
00048
00049             for (int i = 0; i < urls.size(); i++) {
00050                 String url = urls.get(i);
00051                 GetMethod getMethod = new GetMethod(url);
00052                 getMethod.addRequestHeader("Content-Type", "application/json");
00053                 httpClient.executeMethod(getMethod);
00054                 String response = getMethod.getResponseBodyAsString();
00055                 getMethod.releaseConnection();
00056
00057                 JSONObject object = JSON.parseObject(response);
00058                 JSONObject data = JSON.parseObject(object.getString("data"));
00059                 JSONArray data_result = JSON.parseArray(data.getString("result"));
00060                 System.out.println(data_result);
00061
00062                 for(int j = 0; j < data_result.size(); j++) {
00063                     JSONObject metric = data_result.getJSONObject(j).getJSONObject("metric");
00064                     // System.out.println(metric);
00065                     NodeDto node = new NodeDto();
00066                     switch (i) {
00067                         case 0:
00068                             node.setLabel("node");
00069                             node.setId(metric.getString("node"));
00070                             node.setName(metric.getString("__name__"));
00071                             node.setProperty(metric.toString());
00072                             break;
00073                         case 1:
00074                             node.setLabel("deployment");
00075                             node.setId(metric.getString("deployment"));
00076                             node.setNamespace(metric.getString("namespace"));
00077                             node.setName(metric.getString("__name__"));
00078                             node.setProperty(metric.toString());
00079                             break;
00080                         case 2:
00081                             node.setLabel("pod");
00082                             node.setId(metric.getString("pod"));
00083                             node.setPod_node(metric.getString("node"));
00084                             node.setNamespace(metric.getString("namespace"));
00085                             node.setName(metric.getString("__name__"));
00086                             node.setProperty(metric.toString());
00087                             break;
00088                         case 3:
00089                             node.setLabel("service");
00090                             node.setId(metric.getString("service"));
00091                             node.setNamespace(metric.getString("namespace"));
00092                             node.setName(metric.getString("__name__"));
00093                             node.setProperty(metric.toString());
00094                             break;
00095                     }
00096                     result.getNodes().add(node);
00097                 }
00098             }
00099
00100             for (int i = 0; i < result.getNodes().size(); i++) {

```

```

00097         NodeDto node = result.getNodes().get(i);
00098         for (int j = i + 1; j < result.getNodes().size(); j++) {
00099             NodeDto tem = result.getNodes().get(j);
00100             if (node.getLabel().equals("node") && tem.getLabel().equals("pod")) {
00101                 if (node.getId().equals(tem.getPod_node())) {
00102                     EdgeDto edge = new EdgeDto();
00103                     edge.setFrom_id(node.getId());
00104                     edge.setTo_id(tem.getId());
00105                     edge.setValue("node-pod");
00106                     result.getEdges().add(edge);
00107                 }
00108             }
00109
00110             if (node.getLabel().equals("pod") && tem.getLabel().equals("node")) {
00111                 if (node.getPod_node().equals(tem.getId())) {
00112                     EdgeDto edge = new EdgeDto();
00113                     edge.setFrom_id(node.getId());
00114                     edge.setTo_id(tem.getId());
00115                     edge.setValue("node-pod");
00116                     result.getEdges().add(edge);
00117                 }
00118             }
00119
00120             if (node.getLabel().equals("pod") && tem.getLabel().equals("service")) {
00121                 if (node.getNamespace().equals(tem.getNamespace())) {
00122                     EdgeDto edge = new EdgeDto();
00123                     edge.setFrom_id(node.getId());
00124                     edge.setTo_id(tem.getId());
00125                     edge.setValue("pod-service");
00126                     result.getEdges().add(edge);
00127                 }
00128             }
00129
00130             if (node.getLabel().equals("service") && tem.getLabel().equals("pod")) {
00131                 if (node.getNamespace().equals(tem.getNamespace())) {
00132                     EdgeDto edge = new EdgeDto();
00133                     edge.setFrom_id(node.getId());
00134                     edge.setTo_id(tem.getId());
00135                     edge.setValue("pod-service");
00136                     result.getEdges().add(edge);
00137                 }
00138             }
00139
00140             if (node.getLabel().equals("service") && tem.getLabel().equals("deployment")) {
00141                 if (node.getNamespace().equals(tem.getNamespace())) {
00142                     EdgeDto edge = new EdgeDto();
00143                     edge.setFrom_id(node.getId());
00144                     edge.setTo_id(tem.getId());
00145                     edge.setValue("service-deployment");
00146                     result.getEdges().add(edge);
00147                 }
00148             }
00149
00150             if (node.getLabel().equals("deployment") && tem.getLabel().equals("service")) {
00151                 if (node.getNamespace().equals(tem.getNamespace())) {
00152                     EdgeDto edge = new EdgeDto();
00153                     edge.setFrom_id(node.getId());
00154                     edge.setTo_id(tem.getId());
00155                     edge.setValue("service-deployment");
00156                     result.getEdges().add(edge);
00157                 }
00158             }
00159         }
00160     }
00161     System.out.println(result.getNodes().size());
00162     for (int i = 0; i < result.getNodes().size(); i++) {
00163         NodeDto n = result.getNodes().get(i);
00164         System.out.println(n.getLabel() + "\t" + n.getId() + "\t" + n.getPod_node() + "\t" +
n.getNamespace());
00165     }
00166     return new ResponseEntity<>(JSON.toJSONString(result), HttpStatus.OK);
00167 } catch (Exception e) {
00168     logger.info(e.toString());
00169     return new ResponseEntity<>(HttpStatus.BAD_REQUEST);
00170 }
00171 }
00172 }

```

7.11 src/main/java/com/example/backend/controller/LogController.java

File Reference

Classes

- class [com.example.backend.controller.LogController](#)

Packages

- package [com.example.backend.controller](#)

7.12 LogController.java

[Go to the documentation of this file.](#)

```

00001 package com.example.backend.controller;
00002
00003 import com.alibaba.fastjson.JSONArray;
00004 import com.alibaba.fastjson.JSONObject;
00005 import org.apache.commons.httpclient.HttpClient;
00006 import org.apache.commons.httpclient.methods.InputStreamRequestEntity;
00007 import org.apache.commons.httpclient.methods.PostMethod;
00008 import org.apache.commons.httpclient.methods.RequestEntity;
00009 import org.slf4j.Logger;
00010 import org.slf4j.LoggerFactory;
00011 import org.springframework.http.HttpStatus;
00012 import org.springframework.http.ResponseEntity;
00013 import org.springframework.web.bind.annotation.RequestMapping;
00014 import org.springframework.web.bind.annotation.RequestMethod;
00015 import org.springframework.web.bind.annotation.RequestParam;
00016 import org.springframework.web.bind.annotation.RestController;
00017
00018 import java.io.ByteArrayInputStream;
00019 import java.io.IOException;
00020 import java.io.InputStream;
00021 import java.text.SimpleDateFormat;
00022 import java.util.ArrayList;
00023 import java.util.Date;
00024 import java.util.TimeZone;
00025
00026 import static java.lang.Math.abs;
00027
00028 @RestController
00029 public class LogController {
00030     private String baseUrl = "http://10.60.38.174:31016/_search";
00031     private final Logger logger = LoggerFactory.getLogger(this.getClass());
00032
00033     @RequestMapping(value = "log", method = RequestMethod.GET)
00034     public ResponseEntity<String> getLogs(
00035         @RequestParam(required = false) String dataset,
00036         @RequestParam(required = false) String model,
00037         @RequestParam(required = false, defaultValue = "0") Long time_start,
00038         @RequestParam(required = false, defaultValue = "2147483647000") Long time_end)
00039         throws IOException {
00040         String url = baseUrl;
00041
00042         SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss");
00043         String start = sdf.format(new Date(Long.parseLong(String.valueOf(time_start * 1000))));
00044         String end = sdf.format(new Date(Long.parseLong(String.valueOf(time_end * 1000))));
00045
00046         JSONObject jsonObject = new JSONObject();
00047         jsonObject.put("query", new JSONObject());
00048         jsonObject.put("fields", new JSONArray());
00049         jsonObject.getJSONObject("query").put("range", new JSONObject());
00050         jsonObject.getJSONObject("query").getJSONObject("range").put("@timestamp", new JSONObject());
00051         jsonObject.getJSONObject("query").getJSONObject("range").getJSONObject("@timestamp").put("gte",
00052             start);
00053         jsonObject.getJSONObject("query").getJSONObject("range").getJSONObject("@timestamp").put("lt",
00054             end);
00055         JSONObject jsonObject1 = new JSONObject();
00056         jsonObject1.put("field", "@timestamp");
00057         jsonObject.getJSONArray("fields").add(jsonObject1);

```



```

00056
00057     String jsonData = jsonObject.toString();
00058     //     logger.info(jsonData);
00059
00060     try {
00061         HttpClient httpClient = new HttpClient();
00062         httpClient.getHttpConnectionManager().getParams().setConnectionTimeout(15000);
00063         PostMethod postMethod = new PostMethod(url);
00064         postMethod.addRequestHeader("Content-type", "application/json; charset=utf-8");
00065         byte[] requestBytes = jsonData.getBytes("utf-8");
00066         InputStream inputStream = new ByteArrayInputStream(requestBytes, 0, requestBytes.length);
00067
00068         RequestEntity requestEntity = new InputStreamRequestEntity(inputStream,
requestBytes.length, "application/json; charset=utf-8");
00069         postMethod.setRequestEntity(requestEntity);
00070         int i = httpClient.executeMethod(postMethod);
00071
00072         byte[] responseBody = postMethod.getResponseBody();
00073         String s = new String(responseBody);
00074
00075         JSONObject response = JSONObject.parseObject(s);
00076         //     System.out.println(response);
00077
00078         JSONObject result = new JSONObject();
00079         long tot = 0;
00080         int num = 0;
00081         ArrayList<Integer> arrayList = new ArrayList<Integer>();
00082         result.put("logging", new JSONArray());
00083
00084         JSONArray infoArray = response.getJSONObject("hits").getJSONArray("hits");
00085         for (int j = 0; j < infoArray.size(); j++) {
00086             JSONObject source = infoArray.getJSONObject(j).getJSONObject("_source");
00087             JSONObject tem = new JSONObject();
00088
00089             String pattern = "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'";
00090             SimpleDateFormat simpleDateFormat = new SimpleDateFormat(pattern);
00091             simpleDateFormat.setTimeZone(TimeZone.getTimeZone("UTC"));
00092             String date = source.getString("@timestamp");
00093             String message = source.getString("message");
00094
00095             tem.put("time", simpleDateFormat.parse(date));
00096             tem.put("content", message);
00097
00098             String message_hash = message + dataset + model;
00099             int score = abs(message_hash.hashCode()) % 100 + 1;
00100             tot += score;
00101             num++;
00102             arrayList.add(score);
00103             tem.put("score", score);
00104
00105             result.getJSONArray("logging").add(tem);
00106         }
00107
00108         Double probability;
00109         Double threshold;
00110         if (num > 0) {
00111             Double mean = Double.valueOf(tot) / Double.valueOf(num);
00112             Double sum_square_diff = 0.0;
00113             for (int sco : arrayList) {
00114                 sum_square_diff += Double.valueOf((sco - mean) * (sco - mean));
00115             }
00116             Double sta = Math.sqrt(sum_square_diff / num);
00117             threshold = mean + 1 * sta;
00118
00119             Double count = 0.0;
00120             for (int sco : arrayList) {
00121                 if (sco > threshold) count++;
00122             }
00123             probability = count / num;
00124         } else {
00125             probability = 0.0;
00126             threshold = 0.0;
00127         }
00128
00129         result.put("probability", probability);
00130         result.put("threshold", threshold);
00131         return new ResponseEntity<String>(result.toString(), HttpStatus.OK);
00132     } catch (Exception e) {
00133         e.printStackTrace();
00134         return new ResponseEntity<>(HttpStatus.GATEWAY_TIMEOUT);
00135     }
00136 }
00137
00138 @RequestMapping(value = "log/node", method = RequestMethod.GET)
00139 public ResponseEntity<String> getLogsByNode(
00140     @RequestParam(required = false, defaultValue = "0") Long time_start,
00141     @RequestParam(required = false, defaultValue = "2147483647000") Long time_end,

```

```

00142         @RequestParam(required = false) String entity_id)
00143         throws IOException {
00144     String url = baseUrl;
00145
00146     SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss");
00147     String start = sdf.format(new Date(Long.parseLong(String.valueOf(time_start * 1000))));
00148     String end = sdf.format(new Date(Long.parseLong(String.valueOf(time_end * 1000))));
00149
00150     JSONObject jsonObject = new JSONObject();
00151     jsonObject.put("query", new JSONObject());
00152     jsonObject.put("fields", new JSONArray());
00153     jsonObject.getJSONObject("query").put("range", new JSONObject());
00154     jsonObject.getJSONObject("query").getJSONObject("range").put("@timestamp", new JSONObject());
00155     jsonObject.getJSONObject("query").getJSONObject("range").getJSONObject("@timestamp").put("gte",
start);
00156     jsonObject.getJSONObject("query").getJSONObject("range").getJSONObject("@timestamp").put("lt",
end);
00157     JSONObject jsonObject1 = new JSONObject();
00158     jsonObject1.put("field", "@timestamp");
00159     jsonObject.getJSONArray("fields").add(jsonObject1);
00160
00161     String jsonData = jsonObject.toString();
00162     // logger.info(jsonData);
00163
00164     try {
00165         HttpClient httpClient = new HttpClient();
00166         httpClient.getHttpConnectionManager().getParams().setConnectionTimeout(15000);
00167         PostMethod postMethod = new PostMethod(url);
00168         postMethod.addRequestHeader("Content-type", "application/json; charset=utf-8");
00169         byte[] requestBytes = jsonData.getBytes("utf-8");
00170         InputStream inputStream = new ByteArrayInputStream(requestBytes, 0, requestBytes.length);
00171
00172         RequestEntity requestEntity = new InputStreamRequestEntity(inputStream,
requestBytes.length, "application/json; charset=utf-8");
00173         postMethod.setRequestEntity(requestEntity);
00174         int i = httpClient.executeMethod(postMethod);
00175
00176         byte[] responseBody = postMethod.getResponseBody();
00177         String s = new String(responseBody);
00178
00179         JSONObject response = JSONObject.parseObject(s);
00180         // System.out.println(response);
00181
00182         JSONArray result = new JSONArray();
00183         JSONArray infoArray = response.getJSONObject("hits").getJSONArray("hits");
00184         for (int j = 0; j < infoArray.size(); j++) {
00185             JSONObject host =
infoArray.getJSONObject(j).getJSONObject("_source").getJSONObject("host");
00186             if (host == null && entity_id != null) continue;
00187             String id = host.getString("name");
00188
00189             if (entity_id != null && !id.equals(entity_id)) continue;
00190             JSONObject source = infoArray.getJSONObject(j).getJSONObject("_source");
00191             JSONObject tem = new JSONObject();
00192
00193             String pattern = "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'";
00194             SimpleDateFormat simpleDateFormat = new SimpleDateFormat(pattern);
00195             simpleDateFormat.setTimeZone(TimeZone.getTimeZone("UTC"));
00196             String date = source.getString("@timestamp");
00197             String message = source.getString("message");
00198
00199             tem.put("time", simpleDateFormat.parse(date));
00200             tem.put("content", message);
00201
00202             result.add(tem);
00203         }
00204
00205         return new ResponseEntity<String>(result.toString(), HttpStatus.OK);
00206     } catch (Exception e) {
00207         e.printStackTrace();
00208         return new ResponseEntity<>(HttpStatus.GATEWAY_TIMEOUT);
00209     }
00210 }
00211 }

```

7.13 src/main/java/com/example/backend/controller/MetricController.java File Reference

Classes

- class [com.example.backend.controller.MetricController](#)

Packages

- package [com.example.backend.controller](#)

7.14 MetricController.java

[Go to the documentation of this file.](#)

```

00001 package com.example.backend.controller;
00002
00003 import com.alibaba.fastjson.JSON;
00004 import com.alibaba.fastjson.JSONArray;
00005 import com.alibaba.fastjson.JSONObject;
00006 import org.apache.commons.httpclient.HttpClient;
00007 import org.apache.commons.httpclient.methods.GetMethod;
00008 import org.slf4j.Logger;
00009 import org.slf4j.LoggerFactory;
00010 import org.springframework.http.HttpStatus;
00011 import org.springframework.http.ResponseEntity;
00012 import org.springframework.web.bind.annotation.RequestMapping;
00013 import org.springframework.web.bind.annotation.RequestMethod;
00014 import org.springframework.web.bind.annotation.RequestParam;
00015 import org.springframework.web.bind.annotation.RestController;
00016
00017 import java.io.IOException;
00018 import java.util.ArrayList;
00019
00020 @RestController
00021 public class MetricController {
00022     private final Logger logger = LoggerFactory.getLogger(this.getClass());
00023     private String baseUrl = "http://10.60.38.174:31013/api/v1/query_range?query=";
00024
00025     @RequestMapping(value = "timeseries", method = RequestMethod.GET)
00026     public ResponseEntity<String> getMetrics(
00027         @RequestParam(required = false) String dataset,
00028         @RequestParam(required = false) String model,
00029         @RequestParam(required = false, defaultValue = "0") Long time_start,
00030         @RequestParam(required = false, defaultValue = "999999999") Long time_end)
00031         throws IOException
00032     {
00033         ArrayList<String> urls = new ArrayList<>();
00034
00035         urls.add(baseUrl + "kube_node_info" + "&start=" + time_start.toString() + "&end=" +
00036             time_end.toString() + "&step=300s");
00037         urls.add(baseUrl + "kube_deployment_labels" + "&start=" + time_start.toString() + "&end=" +
00038             time_end.toString() + "&step=300s");
00039         urls.add(baseUrl + "kube_pod_info" + "&start=" + time_start.toString() + "&end=" +
00040             time_end.toString() + "&step=300s");
00041         urls.add(baseUrl + "kube_service_info" + "&start=" + time_start.toString() + "&end=" +
00042             time_end.toString() + "&step=300s");
00043
00044         try {
00045             HttpClient httpClient = new HttpClient();
00046             httpClient.getHttpConnectionManager().getParams().setConnectionTimeout(15000);
00047
00048             JSONObject result = new JSONObject();
00049             result.put("series", new JSONArray());
00050             long tot = 0;
00051             int num = 0;
00052             ArrayList<Integer> arrayList = new ArrayList<Integer>();
00053
00054             for (String url : urls) {
00055                 GetMethod getMethod = new GetMethod(url);
00056                 getMethod.addRequestHeader("Content-Type", "application/json");
00057                 logger.info(url);
00058                 httpClient.executeMethod(getMethod);

```

```

00055         String response = getMethod.getResponseBodyAsString();
00056         getMethod.releaseConnection();
00057
00058         JSONObject object = JSON.parseObject(response);
00059         JSONObject data = JSON.parseObject(object.getString("data"));
00060         JSONArray data_result = JSON.parseArray(data.getString("result"));
00061         logger.info(data_result.toString());
00062
00063         for (int i = 0; i < data_result.size(); i++) {
00064             JSONObject value = data_result.getJSONObject(i);
00065             JSONArray arr = value.getJSONArray("values");
00066
00067             for (int j = 0; j < arr.size(); j++) {
00068                 JSONArray subArr = arr.getJSONArray(j);
00069                 JSONObject tem = new JSONObject();
00070                 tem.put("time", subArr.getString(0));
00071                 long v = (subArr.getString(1) + model + dataset).hashCode();
00072                 tem.put("value", Long.valueOf(subArr.getString(1)));
00073                 tem.put("score", v % 101);
00074
00075                 result.getJSONArray("series").add(tem);
00076
00077                 tot += v % 101;
00078                 num++;
00079                 arrayList.add((int) (v % 101));
00080             }
00081         }
00082     }
00083     Double probability, threshold;
00084     if (num > 0) {
00085         Double mean = Double.valueOf(tot) / Double.valueOf(num);
00086         Double sum_square_diff = 0.0;
00087         for (int sco : arrayList) {
00088             sum_square_diff += Double.valueOf((sco - mean) * (sco - mean));
00089         }
00090         Double sta = Math.sqrt(sum_square_diff / Double.valueOf(num));
00091         threshold = mean + 1 * sta;
00092
00093         Double count = 0.0;
00094         for (int sco : arrayList) {
00095             if (sco > threshold) count++;
00096         }
00097         probability = count / Double.valueOf(num);
00098         logger.info(mean.toString());
00099         logger.info(threshold.toString());
00100     } else {
00101         probability = 0.0;
00102     }
00103
00104     result.put("probability", probability);
00105
00106     return new ResponseEntity<String>(result.toString(), HttpStatus.OK);
00107 } catch (Exception e) {
00108     logger.info(e.toString());
00109     return new ResponseEntity<>(HttpStatus.BAD_REQUEST);
00110 }
00111 }
00112
00113 @RequestMapping(value = "timeseries/node", method = RequestMethod.GET)
00114 public ResponseEntity<String> getMetricsByNode(
00115     @RequestParam(required = false, defaultValue = "0") Long time_start,
00116     @RequestParam(required = false, defaultValue = "999999999") Long time_end,
00117     @RequestParam(required = false) String entity_id)
00118     throws IOException
00119 {
00120     ArrayList<String> urls = new ArrayList<>();
00121
00122     urls.add(baseUrl + "kube_node_info" + "&start=" + time_start.toString() + "&end=" +
time_end.toString() + "&step=300s");
00123     urls.add(baseUrl + "kube_deployment_labels" + "&start=" + time_start.toString() + "&end=" +
time_end.toString() + "&step=300s");
00124     urls.add(baseUrl + "kube_pod_info" + "&start=" + time_start.toString() + "&end=" +
time_end.toString() + "&step=300s");
00125     urls.add(baseUrl + "kube_service_info" + "&start=" + time_start.toString() + "&end=" +
time_end.toString() + "&step=300s");
00126
00127     try {
00128         HttpClient httpClient = new HttpClient();
00129         httpClient.getHttpConnectionManager().getParams().setConnectionTimeout(15000);
00130
00131         JSONArray result = new JSONArray();
00132         ArrayList<Integer> arrayList = new ArrayList<Integer>();
00133
00134         for (int urlNum = 0; urlNum < urls.size(); urlNum++) {
00135             String url = urls.get(urlNum);
00136             GetMethod getMethod = new GetMethod(url);
00137             getMethod.setRequestHeader("Content-Type", "application/json");

```

```

00138         logger.info(url);
00139         httpClient.executeMethod(getMethod);
00140         String response = getMethod.getResponseBodyAsString();
00141         getMethod.releaseConnection();
00142
00143         JSONObject object = JSON.parseObject(response);
00144         JSONObject data = JSON.parseObject(object.getString("data"));
00145         JSONArray data_result = JSON.parseArray(data.getString("result"));
00146         logger.info(data_result.toString());
00147
00148         for (int i = 0; i < data_result.size(); i++) {
00149             JSONObject value = data_result.getJSONObject(i);
00150             if (entity_id != null) {
00151                 JSONObject t = value.getJSONObject("metric");
00152                 String entity;
00153                 switch(urlNum) {
00154                     case 0:
00155                         entity = "node";
00156                         break;
00157                     case 1:
00158                         entity = "deployment";
00159                         break;
00160                     case 2:
00161                         entity = "pod";
00162                         break;
00163                     default:
00164                         entity = "service";
00165                         break;
00166                 }
00167
00168                 String id = t.getString(entity);
00169                 if (!entity_id.equals(id)) continue;
00170             }
00171
00172             JSONArray arr = value.getJSONArray("values");
00173
00174             for (int j = 0; j < arr.size(); j++) {
00175                 JSONArray subArr = arr.getJSONArray(j);
00176                 JSONObject tem = new JSONObject();
00177                 tem.put("time", subArr.getString(0));
00178                 long v = Long.valueOf(subArr.getString(1));
00179                 tem.put("value", v);
00180
00181                 result.add(tem);
00182             }
00183         }
00184     }
00185     return new ResponseEntity<String>(result.toString(), HttpStatus.OK);
00186 } catch (Exception e) {
00187     logger.info(e.toString());
00188     return new ResponseEntity<>(HttpStatus.BAD_REQUEST);
00189 }
00190 }
00191
00192 }

```

7.15 src/main/java/com/example/backend/dto/AlertStatusDto.java File Reference

Classes

- class [com.example.backend.dto.AlertStatusDto](#)

Packages

- package [com.example.backend.dto](#)

7.16 AlertStatusDto.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 @Data
00006 public class AlertStatusDto {
00007     String id;
00008     String status;
00009 }
```

7.17 src/main/java/com/example/backend/dto/AllEmployeeInfoOutDto.java File Reference

Classes

- class [com.example.backend.dto.AllEmployeeInfoOutDto](#)

Packages

- package [com.example.backend.dto](#)

7.18 AllEmployeeInfoOutDto.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002 import lombok.Data;
00003
00004 @Data
00005 public class AllEmployeeInfoOutDto {
00006     private String id;
00007     private String name;
00008     private String gender;
00009     private String occupation;
00010     private String birthday;
00011     private Double attendance_rate;
00012     private Double award_times;
00013     private String avatar;
00014 }
```

7.19 src/main/java/com/example/backend/dto/AttendInfo.java File Reference

Classes

- class [com.example.backend.dto.AttendInfo](#)

Packages

- package [com.example.backend.dto](#)

7.20 AttendInfo.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 @Data
00006 public class AttendInfo {
00007     private String plan_id;
00008     private String time_start;
00009     private String time_end;
00010     private String place;
00011     private boolean attendance;
00012 }
```

7.21 src/main/java/com/example/backend/dto/EdgeDto.java File Reference

Classes

- class [com.example.backend.dto.EdgeDto](#)

Packages

- package [com.example.backend.dto](#)

7.22 EdgeDto.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 @Data
00006 public class EdgeDto {
00007     String from_id;
00008     String to_id;
00009     String value;
00010 }
```

7.23 src/main/java/com/example/backend/dto/GraphDto.java File Reference

Classes

- class [com.example.backend.dto.GraphDto](#)

Packages

- package [com.example.backend.dto](#)

7.24 GraphDto.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 import java.util.ArrayList;
00006 import java.util.List;
00007
00008 @Data
00009 public class GraphDto {
00010     List<NodeDto> nodes = new ArrayList<>();
00011     List<EdgeDto> edges = new ArrayList<>();
00012 }
```

7.25 src/main/java/com/example/backend/dto/MyToken.java File Reference

Classes

- class [com.example.backend.dto.MyToken](#)

Packages

- package [com.example.backend.dto](#)

7.26 MyToken.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 import java.util.List;
00006
00007 @Data
00008 public class MyToken {
00009     private boolean active;
00010     private String client_id;
00011     private String username;
00012     private String token_type;
00013     private int exp;
00014     private int iat;
00015     private int nbf;
00016     private String sub;
00017     private String iss;
00018     private List<String> aud;
00019 }
```

7.27 src/main/java/com/example/backend/dto/NodeDto.java File Reference

Classes

- class [com.example.backend.dto.NodeDto](#)

Packages

- package [com.example.backend.dto](#)

7.28 NodeDto.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003
00004 import com.fasterxml.jackson.annotation.JsonIgnore;
00005
00006 public class NodeDto {
00007     String id;
00008     String label;
00009     String name;
00010     String property;
00011     @JsonIgnore
00012     String pod_node;
00013     @JsonIgnore
00014     String namespace;
00015
00016     public String getId() { return id; }
00017     public void setId(String id) { this.id = id; }
00018
00019     public String getLabel() { return label; }
00020     public void setLabel(String label) { this.label = label; }
00021
00022     public String getName() { return name; }
00023     public void setName(String name) { this.name = name; }
00024
00025     public String getProperty() { return property; }
00026     public void setProperty(String property) { this.property = property; }
00027
00028     public String getPod_node() { return pod_node; }
00029     public void setPod_node(String pod_node) { this.pod_node = pod_node; }
00030
00031     public String getNamespace() { return namespace; }
00032     public void setNamespace(String namespace) { this.namespace = namespace; }
00033 }
```

7.29 src/main/java/com/example/backend/dto/OneEmployeeInDto.java

File Reference

Classes

- class [com.example.backend.dto.OneEmployeeInDto](#)

Packages

- package [com.example.backend.dto](#)

7.30 OneEmployeeInDto.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 @Data
00006 public class OneEmployeeInDto {
00007     private String id;
00008     private String name;
00009     private String gender;
00010     private String occupation;
00011     private String birthday;
00012     private String avatar;
00013     private String cover;
00014     private String token;
00015 }
```

7.31 src/main/java/com/example/backend/dto/OneEmployeeOutDto.java File Reference

Classes

- class [com.example.backend.dto.OneEmployeeOutDto](#)

Packages

- package [com.example.backend.dto](#)

7.32 OneEmployeeOutDto.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 import java.util.Collection;
00006 import java.util.HashSet;
00007
00008 @Data
00009 public class OneEmployeeOutDto {
00010     private String id;
00011     private String name;
00012     private String gender;
00013     private String occupation;
00014     private String birthday;
00015     private String avatar;
00016     private String cover;
00017     private Collection<AttendInfo> attends;
00018     private Collection<PrizeInfo> prizes;
00019     private Collection<PayrollInfo> payrolls;
00020 }
```

7.33 src/main/java/com/example/backend/dto/PayrollInfo.java File Reference

Classes

- class [com.example.backend.dto.PayrollInfo](#)

Packages

- package [com.example.backend.dto](#)

7.34 PayrollInfo.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 @Data
00006 public class PayrollInfo {
00007     private String pay_datetime;
00008     private Double amount;
00009 }
```

7.35 src/main/java/com/example/backend/dto/PrizeInfo.java File Reference

Classes

- class [com.example.backend.dto.PrizeInfo](#)

Packages

- package [com.example.backend.dto](#)

7.36 PrizeInfo.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 @Data
00006 public class PrizeInfo {
00007     private String prize_datetime;
00008     private String level;
00009     private Double amount;
00010 }
```

7.37 src/main/java/com/example/backend/dto/TokenInfo.java File Reference

Classes

- class [com.example.backend.dto.TokenInfo](#)

Packages

- package [com.example.backend.dto](#)

7.38 TokenInfo.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.dto;
00002
00003 import lombok.Data;
00004
00005 @Data
00006 public class TokenInfo {
00007     String username;
00008     Boolean active;
00009 }
```

7.39 src/main/java/com/example/backend/entity/AlertEntity.java File Reference

Classes

- class [com.example.backend.entity.AlertEntity](#)

Packages

- package [com.example.backend.entity](#)

7.40 AlertEntity.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.entity;
00002
00003 import com.alibaba.fastjson.annotation.JSONField;
00004 import org.springframework.data.mongodb.core.mapping.Document;
00005
00006 @Document(collection = "Test")
00007 public class AlertEntity {
00008     private String id;
00009     @JSONField(serialize=false)
00010     private String pod;
00011     private Long time;
00012     private String entity_id;
00013     private String category;
00014     private String description;
00015     private String status;
00016     private String entity_name;
00017
00018     public Long get_time() {
00019         return time;
00020     }
00021
00022     public void set_time(Long time) {
00023         this.time = time;
00024     }
00025
00026     public String get_entity_id() {
00027         return entity_id;
00028     }
00029
00030     public void set_entity_id(String entity_id) {this.entity_id = entity_id;}
00031
00032     public String get_pod() {
00033         return pod;
00034     }
00035
00036     public void set_pod(String pod) {
00037         this.pod = pod;
00038     }
00039
00040     public String get_category() {return category;}
00041
00042     public void set_category(String category) {this.category = category;}
00043
00044     public String get_description() {return description;}
00045
00046     public void set_description(String description) {this.description = description;}
00047
00048     public String get_status() {return status;}
00049
00050     public void set_status(String status) {this.status = status;}
00051
00052     public String get_entity_name() {return entity_name;}
00053
00054     public void set_entity_name(String entity_name) {this.entity_name = entity_name;}
00055
00056     public String get_id() {return id;}
00057
00058     public void set_id(String id) {this.id = id;}
00059 }
```

7.41 src/main/java/com/example/backend/entity/EmployeeEntity.java File Reference

Classes

- class [com.example.backend.entity.EmployeeEntity](#)

Packages

- package [com.example.backend.entity](#)

7.42 EmployeeEntity.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.entity;
00002
00003 import lombok.Data;
00004
00005 import java.math.BigInteger;
00006 import java.sql.Date;
00007
00008 @Data
00009 public class EmployeeEntity {
00010     private BigInteger id;
00011     private String name;
00012     private String gender;
00013     private String occupation;
00014     private String birthday;
00015     private String avatar;
00016     private String cover;
00017 }
```

7.43 src/main/java/com/example/backend/scheduler/AlertScheduler.java File Reference

Classes

- class [com.example.backend.scheduler.AlertScheduler](#)

Packages

- package [com.example.backend.scheduler](#)

7.44 AlertScheduler.java

[Go to the documentation of this file.](#)

```

00001 package com.example.backend.scheduler;
00002
00003 import com.alibaba.fastjson.JSON;
00004 import com.alibaba.fastjson.JSONArray;
00005 import com.alibaba.fastjson.JSONObject;
00006 import com.example.backend.entity.AlertEntity;
00007 import org.apache.commons.httpclient.HttpClient;
00008 import org.apache.commons.httpclient.methods.GetMethod;
00009 import org.slf4j.Logger;
00010 import org.slf4j.LoggerFactory;
00011 import org.springframework.beans.factory.annotation.Autowired;
00012 import org.springframework.data.mongodb.core.MongoTemplate;
00013 import org.springframework.data.mongodb.core.query.Criteria;
00014 import org.springframework.data.mongodb.core.query.Query;
00015 import org.springframework.scheduling.annotation.Scheduled;
00016 import org.springframework.stereotype.Component;
00017
00018 import java.io.IOException;
00019
00020 @Component
00021 public class AlertScheduler {
00022     @Autowired
00023     MongoTemplate mongoTemplate;
00024
00025     private final Logger logger = LoggerFactory.getLogger(this.getClass());
00026     private String baseUrl = "http://10.60.38.174:31013/api/v1/query_range?query=ALERTS";
00027
00028     @Scheduled(initialDelay=1000, fixedRate=1000 * 60 * 30)
00029     public void pollingAlert() throws IOException {
00030         long cur = System.currentTimeMillis() / 1000;
00031
00032         String url = baseUrl + "&start=" + String.valueOf(cur - 60 * 30);
00033         url += "&end=" + String.valueOf(cur) + "&step=180s";
00034
00035         logger.info(url);
00036
00037         HttpClient httpClient = new HttpClient();
00038         httpClient.getHttpConnectionManager().getParams().setConnectionTimeout(15000);
00039         GetMethod getMethod = new GetMethod(url);
00040         getMethod.setRequestHeader("Content-Type", "application/json");
00041
00042         httpClient.executeMethod(getMethod);
00043
00044         String result = getMethod.getResponseBodyAsString();
00045         getMethod.releaseConnection();
00046
00047         JSONObject object = JSON.parseObject(result);
00048         JSONObject data = JSON.parseObject(object.getString("data"));
00049         JSONArray data_result = JSON.parseArray(data.getString("result"));
00050
00051         logger.info(data_result.toString());
00052
00053         for(int i = 0; i < data_result.size(); i++) {
00054             JSONObject obj = data_result.getJSONObject(i);
00055
00056             JSONObject metric = JSON.parseObject(obj.getString("metric"));
00057             JSONArray values = JSON.parseArray(obj.getString("values"));
00058
00059             String pod = metric.getString("pod");
00060             String entity_id = metric.getString("namespace");
00061             String category = metric.getString("alertstate");
00062             String description = metric.getString("alertname");
00063             String status = "unread";
00064             String entity_name = metric.getString("container");
00065
00066             for (int j = 0; j < values.size(); j++) {
00067                 JSONArray array = values.getJSONArray(j);
00068
00069                 AlertEntity alertEntity = new AlertEntity();
00070                 alertEntity.set_pod(pod);
00071                 alertEntity.set_time(Long.valueOf(array.getString(0)));
00072                 alertEntity.set_entity_id(entity_id);
00073                 alertEntity.set_category(category);
00074                 alertEntity.set_description(description);
00075                 alertEntity.set_status(status);
00076                 alertEntity.set_entity_name(entity_name);
00077                 mongoTemplate.insert(alertEntity, "Test");
00078             }
00079         }
00080
00081         Query query = new Query(Criteria.where("time").lt(String.valueOf(cur - 60 * 30)));
00082         mongoTemplate.remove(query, AlertEntity.class, "Test");

```

```
00083     }  
00084  
00085 }
```

7.45 src/main/java/com/example/backend/service/EmployeeService.java File Reference

Classes

- interface [com.example.backend.service.EmployeeService](#)

Packages

- package [com.example.backend.service](#)

7.46 EmployeeService.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend.service;  
00002  
00003 import com.example.backend.dto.AllEmployeeInfoOutDto;  
00004 import com.example.backend.dto.OneEmployeeInDto;  
00005 import com.example.backend.dto.OneEmployeeOutDto;  
00006  
00007 import java.math.BigInteger;  
00008 import java.util.List;  
00009  
00010 public interface EmployeeService {  
00011     public List<AllEmployeeInfoOutDto> getAllEmployeeInfo();  
00012     public boolean addEmployee(OneEmployeeInDto employee);  
00013     public boolean updateEmployee(OneEmployeeInDto employee);  
00014     public boolean deleteEmployee(String id);  
00015     public OneEmployeeOutDto getOneEmployeeInfo(String id);  
00016 }
```

7.47 src/main/java/com/example/backend/service/impl/EmployeeServiceImpl.java File Reference

Classes

- class [com.example.backend.service.impl.EmployeeServiceImpl](#)

Packages

- package [com.example.backend.service.impl](#)

7.48 EmployeeServiceImpl.java

[Go to the documentation of this file.](#)

```

00001 package com.example.backend.service.impl;
00002
00003 import com.example.backend.dto.AllEmployeeInfoOutDto;
00004 import com.example.backend.dto.OneEmployeeInDto;
00005 import com.example.backend.dto.OneEmployeeOutDto;
00006 import com.example.backend.entity.AlertEntity;
00007 import com.example.backend.entity.EmployeeEntity;
00008 import com.example.backend.service.EmployeeService;
00009 import com.mongodb.client.result.UpdateResult;
00010 import org.springframework.beans.factory.annotation.Autowired;
00011 import org.springframework.beans.factory.annotation.Value;
00012 import org.springframework.data.mongodb.core.MongoTemplate;
00013 import org.springframework.data.mongodb.core.query.Criteria;
00014 import org.springframework.data.mongodb.core.query.Query;
00015 import org.springframework.data.mongodb.core.query.Update;
00016 import org.springframework.stereotype.Service;
00017 import org.modelmapper.ModelMapper;
00018
00019 import javax.annotation.Resource;
00020 import java.io.FileOutputStream;
00021 import java.io.OutputStream;
00022 import java.math.BigInteger;
00023 import java.util.*;
00024
00025 @Service
00026 public class EmployeeServiceImpl implements EmployeeService {
00027     @Autowired
00028     MongoTemplate mongoTemplate;
00029     @Override
00030     public List<AllEmployeeInfoOutDto> getAllEmployeeInfo() {
00031         List<EmployeeEntity> employeeEntities = mongoTemplate.findAll(EmployeeEntity.class, "User");
00032         List<AllEmployeeInfoOutDto> result = new ArrayList<>();
00033         ModelMapper modelMapper = new ModelMapper();
00034
00035         try {
00036             for (EmployeeEntity employeeEntity : employeeEntities) {
00037                 AllEmployeeInfoOutDto allEmployeeInfoOutDto = modelMapper.map(employeeEntity,
00038                     AllEmployeeInfoOutDto.class);
00039                 Integer hash = (employeeEntity.getId() + "hash_id").hashCode();
00040                 allEmployeeInfoOutDto.setAttendance_rate(Double.valueOf(hash % 100));
00041                 allEmployeeInfoOutDto.setAward_times(Double.valueOf(hash % 10));
00042                 result.add(allEmployeeInfoOutDto);
00043             }
00044         } catch (Exception e) {
00045             e.printStackTrace();
00046             return null;
00047         }
00048         return result;
00049     }
00050     @Override
00051     public boolean addEmployee(OneEmployeeInDto employeeDto) {
00052         List<EmployeeEntity> employees = mongoTemplate.findAll(EmployeeEntity.class, "User");
00053         BigInteger id = BigInteger.ZERO;
00054         for (EmployeeEntity e : employees) {
00055             if (e.getId().compareTo(id) == 1) id = e.getId();
00056         }
00057         id = id.add(BigInteger.ONE);
00058         employeeDto.setId(id.toString());
00059
00060         ModelMapper modelMapper = new ModelMapper();
00061         EmployeeEntity employee = modelMapper.map(employeeDto, EmployeeEntity.class);
00062
00063         try {
00064             mongoTemplate.insert(employee, "User");
00065         } catch (Exception e) {
00066             System.out.println(e.toString());
00067             return false;
00068         }
00069
00070         return true;
00071     }
00072     @Override
00073     public boolean updateEmployee(OneEmployeeInDto employeeDto) {
00074         try {
00075             EmployeeEntity employee = mongoTemplate.findById(employeeDto.getId(),
00076                 EmployeeEntity.class, "User");
00077             employee.setAvatar(employeeDto.getAvatar());
00078             employee.setName(employeeDto.getName());
00079             employee.setBirthday(employeeDto.getBirthday());
00080             employee.setGender(employeeDto.getGender());

```



```

00081         employee.setCover(employeeDto.getCover());
00082         employee.setOccupation(employeeDto.getOccupation());
00083
00084         Query query = new Query(Criteria.where("_id").is(employeeDto.getId()));
00085         Update update = new Update();
00086
00087         update.set("name", employee.getName());
00088         update.set("avatar", employee.getAvatar());
00089         update.set("birthday", employee.getBirthday());
00090         update.set("gender", employee.getGender());
00091         update.set("cover", employee.getCover());
00092         update.set("occupation", employee.getOccupation());
00093
00094         UpdateResult upsert = mongoTemplate.upsert(query, update, EmployeeEntity.class, "User");
00095         return true;
00096     }
00097     catch (Exception e) {
00098         e.printStackTrace();
00099         return false;
00100     }
00101 }
00102
00103 @Override
00104 public boolean deleteEmployee(String id) {
00105     try {
00106         Query query = new Query(Criteria.where("_id").is(id));
00107         mongoTemplate.remove(query, EmployeeEntity.class, "User");
00108         return true;
00109     }
00110     catch (Exception e) {
00111         e.printStackTrace();
00112         return false;
00113     }
00114 }
00115
00116 @Override
00117 public OneEmployeeOutDto getOneEmployeeInfo(String id) {
00118     try {
00119         EmployeeEntity employee = mongoTemplate.findById(id, EmployeeEntity.class, "User");
00120         if (employee == null) return null;
00121
00122         OneEmployeeOutDto out = new OneEmployeeOutDto();
00123         out.setAttends(null);
00124         out.setPrizes(null);
00125         out.setPayrolls(null);
00126         out.setId(id);
00127         out.setGender(employee.getGender());
00128         out.setAvatar(employee.getAvatar());
00129         out.setBirthday(employee.getBirthday());
00130         out.setName(employee.getName());
00131         out.setCover(employee.getCover());
00132         out.setOccupation(employee.getOccupation());
00133         return out;
00134     }
00135     catch (Exception e) {
00136         e.printStackTrace();
00137         return null;
00138     }
00139 }
00140 }

```

7.49 src/test/java/com/example/backend/BackendApplicationTests.java File Reference

Classes

- class `com.example.backend.BackendApplicationTests`

Packages

- package `com.example.backend`

7.50 BackendApplicationTests.java

[Go to the documentation of this file.](#)

```
00001 package com.example.backend;
00002
00003 import org.junit.jupiter.api.Test;
00004 import org.springframework.boot.test.context.SpringBootTest;
00005
00006 @SpringBootTest
00007 class BackendApplicationTests {
00008
00009     @Test
00010     void contextLoads() {
00011     }
00012
00013 }
```

Index

addCorsMappings
 com.example.backend.configuration.AppConfiguration, 20

addEmployee
 com.example.backend.controller.EmployeeController, 22
 com.example.backend.service.EmployeeService, 25
 com.example.backend.service.impl.EmployeeServiceImpl, 28

checkToken
 com.example.backend.controller.EmployeeController, 22

com.example.backend, 9

com.example.backend.BackendApplication, 20
 main, 20

com.example.backend.configuration, 9

com.example.backend.configuration.AppConfiguration, 19
 addCorsMappings, 20

com.example.backend.controller, 9

com.example.backend.controller.AlertController, 11
 getAlerts, 11
 postAlerts, 12

com.example.backend.controller.EmployeeController, 21
 addEmployee, 22
 checkToken, 22
 deleteEmployee, 22
 EmployeeController, 21
 getAllEmployees, 23
 updateEmployee, 24

com.example.backend.controller.KnowledgeGraphController, 30
 getKnoeledgeGraph, 30

com.example.backend.controller.LogController, 31
 getLogs, 32
 getLogsByNode, 32

com.example.backend.controller.MetricController, 32
 getMetrics, 33
 getMetricsByNode, 33

com.example.backend.dto, 10

com.example.backend.dto.AlertStatusDto, 18

com.example.backend.dto.AllEmployeeInfoOutDto, 19

com.example.backend.dto.AttendInfo, 20

com.example.backend.dto.EdgeDto, 21

com.example.backend.dto.GraphDto, 30

com.example.backend.dto.MyToken, 33

com.example.backend.dto.NodeDto, 34

getId, 34

getLabel, 34

getName, 35

getNamespace, 35

getPod_node, 35

getProperty, 36

setId, 36

setLabel, 36

setName, 37

setNamespace, 37

setPod_node, 37

setProperty, 38

com.example.backend.dto.OneEmployeeInDto, 38

com.example.backend.dto.OneEmployeeOutDto, 39

com.example.backend.dto.PayrollInfo, 39

com.example.backend.dto.PrizeInfo, 39

com.example.backend.dto.TokenInfo, 39

com.example.backend.entity, 10

com.example.backend.entity.AlertEntity, 12
 get_category, 13
 get_description, 13
 get_entity_id, 13
 get_entity_name, 13
 get_id, 13
 get_pod, 13
 get_status, 14
 get_time, 14
 set_category, 14
 set_description, 14
 set_entity_id, 15
 set_entity_name, 15
 set_id, 15
 set_pod, 16
 set_status, 16
 set_time, 16

com.example.backend.entity.EmployeeEntity, 24

com.example.backend.scheduler, 10

com.example.backend.scheduler.AlertScheduler, 17
 pollingAlert, 18

com.example.backend.service, 10

com.example.backend.service.EmployeeService, 25
 addEmployee, 25
 deleteEmployee, 25
 getAllEmployeeInfo, 26
 getOneEmployeeInfo, 26
 updateEmployee, 26

com.example.backend.service.impl, 10

com.example.backend.service.impl.EmployeeServiceImpl, 27

- addEmployee, 28
- deleteEmployee, 28
- getAllEmployeeInfo, 29
- getOneEmployeeInfo, 29
- updateEmployee, 29
- deleteEmployee
 - com.example.backend.controller.EmployeeController, 22
 - com.example.backend.service.EmployeeService, 25
 - com.example.backend.service.impl.EmployeeServiceImpl, 28
- EmployeeController
 - com.example.backend.controller.EmployeeController, 21
- get_category
 - com.example.backend.entity.AlertEntity, 13
- get_description
 - com.example.backend.entity.AlertEntity, 13
- get_entity_id
 - com.example.backend.entity.AlertEntity, 13
- get_entity_name
 - com.example.backend.entity.AlertEntity, 13
- get_id
 - com.example.backend.entity.AlertEntity, 13
- get_pod
 - com.example.backend.entity.AlertEntity, 13
- get_status
 - com.example.backend.entity.AlertEntity, 14
- get_time
 - com.example.backend.entity.AlertEntity, 14
- getAlerts
 - com.example.backend.controller.AlertController, 11
- getAllEmployeeInfo
 - com.example.backend.service.EmployeeService, 26
 - com.example.backend.service.impl.EmployeeServiceImpl, 29
- getAllEmployees
 - com.example.backend.controller.EmployeeController, 23
- getId
 - com.example.backend.dto.NodeDto, 34
- getKnoledgeGraph
 - com.example.backend.controller.KnowledgeGraphController, 30
- getLabel
 - com.example.backend.dto.NodeDto, 34
- getLogs
 - com.example.backend.controller.LogController, 32
- getLogsByNode
 - com.example.backend.controller.LogController, 32
- getMetrics
 - com.example.backend.controller.MetricController, 33
- getMetricsByNode
 - com.example.backend.controller.MetricController, 33
- getName
 - com.example.backend.dto.NodeDto, 35
- getNamespace
 - com.example.backend.dto.NodeDto, 35
- getOneEmployeeInfo
 - com.example.backend.service.EmployeeService, 26
 - com.example.backend.service.impl.EmployeeServiceImpl, 29
- getPod_node
 - com.example.backend.dto.NodeDto, 35
- getProperty
 - com.example.backend.dto.NodeDto, 36
- main
 - com.example.backend.BackendApplication, 20
- pollingAlert
 - com.example.backend.scheduler.AlertScheduler, 18
- postAlerts
 - com.example.backend.controller.AlertController, 12
- set_category
 - com.example.backend.entity.AlertEntity, 14
- set_description
 - com.example.backend.entity.AlertEntity, 14
- set_entity_id
 - com.example.backend.entity.AlertEntity, 15
- set_entity_name
 - com.example.backend.entity.AlertEntity, 15
- set_id
 - com.example.backend.entity.AlertEntity, 15
- set_pod
 - com.example.backend.entity.AlertEntity, 16
- set_status
 - com.example.backend.entity.AlertEntity, 16
- set_time
 - com.example.backend.entity.AlertEntity, 16
- setId
 - com.example.backend.dto.NodeDto, 36
- setLabel
 - com.example.backend.dto.NodeDto, 36
- setName
 - com.example.backend.dto.NodeDto, 37
- setNamespace
 - com.example.backend.dto.NodeDto, 37
- setPod_node
 - com.example.backend.dto.NodeDto, 37
- setProperty
 - com.example.backend.dto.NodeDto, 38
- src/main/java/com/example/backend/BackendApplication.java, 41
- src/main/java/com/example/backend/configuration/AppConfiguration.java, 41, 42

src/main/java/com/example/backend/controller/AlertController.java,
42

src/main/java/com/example/backend/controller/EmployeeController.java,
43

src/main/java/com/example/backend/controller/KnowledgeGraphController.java,
45

src/main/java/com/example/backend/controller/LogController.java,
48

src/main/java/com/example/backend/controller/MetricController.java,
51

src/main/java/com/example/backend/dto/AlertStatusDto.java,
53, 54

src/main/java/com/example/backend/dto/AllEmployeeInfoOutDto.java,
54

src/main/java/com/example/backend/dto/AttendInfo.java,
54, 55

src/main/java/com/example/backend/dto/EdgeDto.java,
55

src/main/java/com/example/backend/dto/GraphDto.java,
55, 56

src/main/java/com/example/backend/dto/MyToken.java,
56

src/main/java/com/example/backend/dto/NodeDto.java,
56, 57

src/main/java/com/example/backend/dto/OneEmployeeInDto.java,
57

src/main/java/com/example/backend/dto/OneEmployeeOutDto.java,
58

src/main/java/com/example/backend/dto/PayrollInfo.java,
58

src/main/java/com/example/backend/dto/PrizeInfo.java,
59

src/main/java/com/example/backend/dto/TokenInfo.java,
59

src/main/java/com/example/backend/entity/AlertEntity.java,
60

src/main/java/com/example/backend/entity/EmployeeEntity.java,
61

src/main/java/com/example/backend/scheduler/AlertScheduler.java,
61, 62

src/main/java/com/example/backend/service/EmployeeService.java,
63

src/main/java/com/example/backend/service/impl/EmployeeServiceImpl.java,
63, 64

src/test/java/com/example/backend/BackendApplicationTests.java,
65, 66

updateEmployee

- com.example.backend.controller.EmployeeController,
24
- com.example.backend.service.EmployeeService,
26
- com.example.backend.service.impl.EmployeeServiceImpl,
29