

index	2
pressurestats_package	3
pressurestats_PressurestatsApplication	4
pressurestats_testController	6
pressurestats_testModel	9
pressurestats_testService	12
pressurestats_testStatistics	15
pressurestats_testThrows	18
controller_package	21
controller_Controller	22
exception_package	28
exception_CityStatisticsNotFoundException	29
exception_DayNotFoundException	32
exception_IdNotFoundException	35
exception_ItalianCityNotFoundException	38
exception_NegativeStartException	41
exception_WrongHoursPeriodException	44
exception_WrongMultiplyException	47
model_package	50
model_City	51
model_Forecast	57
service_package	62
service_Service	63
service_ServiceImpl	67
statistics_package	73
statistics_Filters	74
statistics_Statistics	77

Packages

Package	Description
<code>it.univpm.pressurestats</code>	Package principale dell'applicazione PressureStats
<code>it.univpm.pressurestats.controller</code>	Contiene il Controller, necessario per gestire le richieste degli utenti
<code>it.univpm.pressurestats.exception</code>	Contiene le eccezioni personalizzate di PressureStats
<code>it.univpm.pressurestats.model</code>	Contiene tutta la parte di modellazione delle classi
<code>it.univpm.pressurestats.service</code>	Contiene i servizi e le operazioni usate per raccogliere, salvare e mostrare i dati.
<code>it.univpm.pressurestats.statistics</code>	Contiene le statistiche e metodi per filtrarle e presentarle

Package it.univpm.pressurestats

package it.univpm.pressurestats

Package principale dell'applicazione PressureStats

Author:

Pietroniro Cristian, Settimi Diego

Class Summary	
Class	Description
PressurestatsApplication	Avvia l'applicazione Spring.
TestController	Testa il Controller.
TestModel	Testa i vari modelli.
TestService	Testa Service.
TestStatistics	Testa le statistiche.
TestThrows	Testa le eccezioni.

Package [it.univpm.pressurestats](#)

Class PressurestatsApplication

[java.lang.Object](#)
[it.univpm.pressurestats.PressurestatsApplication](#)

```
@SpringBootApplication  
public class PressurestatsApplication  
extends Object
```

Avvia l'applicazione Spring.

Author:

Pietroniro Cristian, Settimi Diego

Constructor Summary

Constructors

Constructor	Description
PressurestatsApplication()	Costruttore della classe.

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
static void	main(String[] args)	Avvia main che avvia l'applicazione SpringBoot

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[toString](#), [wait](#), [wait](#), [wait](#)

Constructor Details

[PressurestatsApplication](#)

```
public PressurestatsApplication()
```

Costruttore della classe.

Method Details

main

```
public static void main(String[] args)
```

Avvia main che avvia l'applicazione SpringBoot

Parameters:

args - argomenti linee di comando JAVA

Package [it.univpm.pressurestats](#)

Class TestController

[java.lang.Object](#)
[it.univpm.pressurestats.TestController](#)

```
@SpringBootTest  
@AutoConfigureMockMvc  
class TestController  
extends Object
```

Testa il Controller.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
private org.springframework.web.conte	context	Interfaccia di configurazione
private org.springframework.test.web.	mockMvc	Rende disponibili al test le applicazioni web senza usare una vera comunicazione HTTP.

Constructor Summary

Constructors

Constructor	Description
TestController()	Costruttore della classe;

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
(package private) void	setUp()	Imposta le variabili per i test.
(package private) void	tearDown()	Usato per rilasciare le risorse dopo che i test sono eseguiti.

```
(package private) void testController() Testa rotte del Controller.
```

Methods inherited from class java.lang.Object

```
clone✉, equals✉, finalize✉, getClass✉, hashCode✉, notify✉, notifyAll✉,  
toString✉, wait✉, wait✉, wait✉
```

Field Details

context

```
@Autowired  
private org.springframework.web.context.WebApplicationContext context
```

Interfaccia di configurazione

mockMvc

```
private org.springframework.test.web.servlet.MockMvc mockMvc
```

Rende disponibili al test le applicazioni web senza usare una vera comunicazione HTTP.

Constructor Details

TestController

```
public TestController()
```

Costruttore della classe;

Method Details

setUp

```
@BeforeEach  
void setUp()  
    throws Exception✉
```

Imposta le variabili per i test.

Throws:

Exception[✉] - Eccezione

tearDown

```
@AfterEach  
void tearDown()  
    throws Exception↗
```

Usato per rilasciare le risorse dopo che i test sono eseguiti.

Throws:

Exception[↗] - Eccezione

testController

```
@Test  
void testController()  
    throws Exception↗
```

Testa rotte del Controller.

Throws:

Exception[↗] - Eccezione

Package [it.univpm.pressurestats](#)

Class TestModel

[java.lang.Object](#)
it.univpm.pressurestats.TestModel

```
class TestModel  
extends Object
```

Testa i vari modelli.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
private City	c	Città usata per test.
private Vector < Forecast > f		Vettore di previsioni usato per test.
private ServiceImpl	s	Implementazione di Service usata per test.

Constructor Summary

Constructors

Constructor	Description
TestModel()	Costruttore della classe.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
(package private) void	setUp()	Imposta le variabili per i test.
(package private) void	tearDown()	Usato per rilasciare le risorse dopo che i test sono eseguiti.
(package private) void	testCity()	Testa metodi della classe City.
(package private) void	testForecast()	Testa metodi della classe Forecast.

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

Field Details

f

```
private Vector<Forecast> f
```

Vettore di previsioni usato per test.

c

```
private City c
```

Città usata per test.

s

```
private ServiceImpl s
```

Implementazione di Service usata per test.

Constructor Details

TestModel

```
public TestModel()
```

Costruttore della classe.

Method Details

setUp

```
@BeforeEach
void setUp()
    throws Exception
```

Imposta le variabili per i test.

Throws:

[Exception](#) - Eccezione

tearDown

```
@AfterEach  
void tearDown()  
    throws Exception
```

Usato per rilasciare le risorse dopo che i test sono eseguiti.

Throws:

[Exception](#) - Eccezione

testCity

```
@Test  
void testCity()
```

Testa metodi della classe City.

testForecast

```
@Test  
void testForecast()
```

Testa metodi della classe Forecast.

Package [it.univpm.pressurestats](#)

Class TestService

[java.lang.Object](#)
it.univpm.pressurestats.TestService

```
class TestService  
extends Object
```

Testa Service.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
(package private)	City	Città usata per test
(package private)	ServiceImpl s	Implementazione di Service usata per test

Constructor Summary

Constructors

Constructor	Description
TestService()	Costruttore della classe.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
(package private)	void setUp()	Imposta le variabili per i test.
(package private)	void tearDown()	Usato per rilasciare le risorse dopo che i test sono eseguiti.
(package private)	void testService()	Testa i metodi di Service

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),

```
toString✉, wait✉, wait✉, wait✉
```

Field Details

s

ServiceImpl s

Implementazione di Service usata per test

city

City city

Città usata per test

Constructor Details

TestService

```
public TestService()
```

Costruttore della classe.

Method Details

setUp

```
@BeforeEach  
void setUp()  
    throws Exception✉
```

Imposta le variabili per i test.

Throws:

Exception[✉] - Eccezione

tearDown

```
@AfterEach  
void tearDown()  
    throws Exception✉
```

Usato per rilasciare le risorse dopo che i test sono eseguiti.

Throws:

Exception[✉] - Eccezione

testService

```
@Test  
void testService()  
    throws ItalianCityNotFoundException,  
           IdNotFoundException
```

Testa i metodi di Service

Throws:

`ItalianCityNotFoundException` - eccezione lanciata se la città non è italiana

`IdNotFoundException` - eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta

Package [it.univpm.pressurestats](#)

Class TestStatistics

[java.lang.Object](#)
it.univpm.pressurestats.TestStatistics

```
class TestStatistics  
extends Object
```

Testa le statistiche.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
private	Filters f	Filtri usati per i test

Constructor Summary

Constructors

Constructor	Description
TestStatistics()	Costruttore della classe.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
(package private)	void setUp()	Imposta le variabili per i test.
(package private)	void tearDown()	Usato per rilasciare le risorse dopo che i test sono eseguiti.
(package private)	void testFilters()	Testa metodi di Filters

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Field Details

f

```
private Filters f
```

Filtrri usati per i test

Constructor Details

TestStatistics

```
public TestStatistics()
```

Costruttore della classe.

Method Details

setUp

```
@BeforeEach  
void setUp()  
    throws Exception
```

Imposta le variabili per i test.

Throws:

`Exception` - Eccezione

tearDown

```
@AfterEach  
void tearDown()  
    throws Exception
```

Usato per rilasciare le risorse dopo che i test sono eseguiti.

Throws:

`Exception` - Eccezione

testFilters

```
@Test  
void testFilters()
```

```
throws WrongHoursPeriodException,  
        CityStatisticsNotFoundException,  
        DayNotFoundException
```

Testa metodi di Filters

Throws:

`WrongHoursPeriodException` - eccezione lanciata se il range orario non è corretto

`CityStatisticsNotFoundException` - eccezione lanciata se le statistiche della città non sono state trovate

`DayNotFoundException` - eccezione lanciata se il giorno non è stato trovato

Package [it.univpm.pressurestats](#)

Class TestThrows

[java.lang.Object](#)
it.univpm.pressurestats.TestThrows

```
class TestThrows  
extends Object
```

Testa le eccezioni.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
private ServiceImpl c		Implementazione di Service usata per test
private Filters f		Filtri usati per test

Constructor Summary

Constructors

Constructor	Description
TestThrows ()	Costruttore della classe.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
(package private) void	setUp()	Imposta le variabili per i test.
(package private) void	tearDown()	Usato per rilasciare le risorse dopo che i test sono eseguiti.
(package private) void	testException()	Testa le eccezioni personalizzate di PressureStats.

Methods inherited from class [java.lang.Object](#)

```
clone✉, equals✉, finalize✉, getClass✉, hashCode✉, notify✉, notifyAll✉,  
toString✉, wait✉, wait✉, wait✉
```

Field Details

c

```
private ServiceImpl c
```

Implementazione di Service usata per test

f

```
private Filters f
```

Filtri usati per test

Constructor Details

TestThrows

```
public TestThrows()
```

Costruttore della classe.

Method Details

setUp

```
@BeforeEach  
void setUp()  
    throws Exception✉
```

Imposta le variabili per i test.

Throws:

```
Exception✉ - Eccezione
```

tearDown

```
@AfterEach  
void tearDown()  
    throws Exception✉
```

Usato per rilasciare le risorse dopo che i test sono eseguiti.

Throws:

`Exception` - Eccezione

testException

```
@Test  
void testException()
```

Testa le eccezioni personalizzate di PressureStats.

Package it.univpm.pressurestats.controller

package it.univpm.pressurestats.controller

Contiene il Controller, necessario per gestire le richieste degli utenti

Author:

Pietroniro Cristian, Settimi Diego

Class Summary

Class	Description
Controller	Processa le varie richieste, prepara il modello e restituisce una risposta.

Package [it.univpm.pressurestats.controller](#)

Class Controller

[java.lang.Object](#)
[it.univpm.pressurestats.controller.Controller](#)

```
@RestController  
public class Controller  
extends Object
```

Processa le varie richieste, prepara il modello e restituisce una risposta.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
(package private) Service	service	Interfaccia Service usata per le operazioni di raccolta e salvataggio dati.
(package private) Statistics	statistics	Oggetto Statistics usato per le operazioni di generazione e salvataggio statistiche.

Constructor Summary

Constructors

Constructor	Description
Controller()	Costruttore della classe.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
org.springframework.ht	getForecast(String id)	Mostra le informazioni attuali su pressione e visibilità.
org.springframework.ht	getStatisticsHourly(String city, String date, int from, int to)	Restituisce e salva le statistiche per più ore di un giorno di una città.

<pre>org.springframework.ht getStatisticsMoreDays ✉> (String city, int days)</pre>	Restituisce le statistiche di pressione e visibilità di una città data, filtrate dal giorno attuale a tot giorni passati.
<pre>org.springframework.ht getStatisticsOneDay ✉> (String city, String date)</pre>	Restituisce e salva le statistiche giornaliere di una città.
<pre>org.springframework.ht saveToFileHourly ✉> (double multiplier)</pre>	Se lasciata in esecuzione, salva con una frequenza impostata dall'utente le informazioni attuali su pressione e visibilità di 5 città non scelte dall'utente.
<pre>org.springframework.ht saveToFileHourly(String id, double multiplier)</pre>	Se lasciata in esecuzione, salva con una frequenza impostata dall'utente le informazioni attuali su pressione e visibilità

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait
```

Field Details

service

```
@Autowired  
Service service
```

Interfaccia Service usata per le operazioni di raccolta e salvataggio dati.

statistics

```
Statistics statistics
```

Oggetto Statistics usato per le operazioni di generazione e salvataggio statistiche.

Constructor Details

Controller

```
public Controller()
```

Costruttore della classe.

Method Details

getForecast

```
@GetMapping("/current")
public org.springframework.http.ResponseEntity<Object> getForecast
(@RequestParam(name="id", defaultValue="3169070")
 String id)
throws
ItalianCityNotFoundException,
IdNotFoundException
```

Mostra le informazioni attuali su pressione e visibilità. I dati vengono stampati su schermo e salvati in un file di testo.

Parameters:

`id` - rappresenta la città di cui si richiedono le previsioni. Valore di default: 3169070 (Rome,IT)

Returns:

le previsioni meteo su pressione e visibilità della città richiesta e le informazioni generali sulla città.

Throws:

`ItalianCityNotFoundException` - eccezione lanciata se la città non è italiana

`IdNotFoundException` - eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta

saveToFileHourly

```
@GetMapping("/multiSave")
public org.springframework.http.ResponseEntity<Object> saveToFileHourly
(@RequestParam(name="multiplier", defaultValue="1")
 double multiplier)
throws
ItalianCityNotFoundException,
WrongMultiplyException,
NegativeStartException,
IdNotFoundException
```

Se lasciata in esecuzione, salva con una frequenza impostata dall'utente le informazioni attuali su pressione e visibilità di 5 città non scelte dall'utente.

Parameters:

`multiplier` - moltiplicatore dell'ora (Esempio: 0,5 salva i dati ogni 30 minuti, 2 ogni 2 ore)

Returns:

Un messaggio di riepilogo con la frequenza

Throws:

`ItalianCityNotFoundException` - eccezione lanciata se la città non è italiana

`WrongMultiplyException` - eccezione lanciata se il moltiplicatore non è ammesso (moltiplicatore minore o uguale a 0.02). Un moltiplicatore di 0.02 restituirebbe dati ogni minuto circa

`NegativeStartException` - eccezione lanciata se il valore di Start è inferiore a 0. (Negativo)

`IdNotFoundException` - eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta

saveToFileHourly

```
@GetMapping("/hourlySave")
public org.springframework.http.ResponseEntity<Object> saveToFileHourly
(@RequestParam(name="id", defaultValue="3169070")
 String id,
 @RequestParam(name="multiplier", defaultValue="1")
 double multiplier)
throws
WrongMultiplyException,
IdNotFoundException,
NegativeStartException
```

Se lasciata in esecuzione, salva con una frequenza impostata dall'utente le informazioni attuali su pressione e visibilità

Parameters:

`id` - rappresenta la città di cui si richiedono le previsioni. Valore di default: 3169070 (Rome,IT)

`multiplier` - moltiplicatore dell'ora (Esempio: 0,5 salva i dati ogni 30 minuti, 2 ogni 2 ore)

Returns:

Un messaggio di riepilogo con la frequenza e la prima misurazione

Throws:

`WrongMultiplyException` - eccezione lanciata se il moltiplicatore non è ammesso (moltiplicatore minore o uguale a 0.02). Un moltiplicatore di 0.02 restituirebbe dati ogni minuto circa

`IdNotFoundException` - eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta

`NegativeStartException` - eccezione lanciata se il valore di Start è inferiore a 0. (Negativo)

getStatisticsOneDay

```
@GetMapping("/oneDay")
public org.springframework.http.ResponseEntity<Object>
```

```
> getStatisticsOneDay(@RequestParam(name="city", defaultValue="Rome")
                      String✉ city,
                      @RequestParam(name="date")
                      String✉ date)
```

Restituisce e salva le statistiche giornaliere di una città.

Parameters:

city - rappresenta la città di cui si richiedono le statistiche. Valore di default: Rome,IT

date - Il giorno di cui si vogliono ricevere statistiche

Returns:

Le statistiche per un giorno

getStatisticsMoreDays

```
@GetMapping("/moreDays")
public org.springframework.http.ResponseEntity<Object✉
> getStatisticsMoreDays(@RequestParam(name="city", defaultValue="Rome")
                        String✉ city,
                        @RequestParam(name="days")
                        int days)
```

Restituisce le statistiche di pressione e visibilità di una città data, filtrate dal giorno attuale a tot giorni passati.

Parameters:

city - rappresenta la città di cui si richiedono le statistiche. Valore di default: Rome,IT

days - Il numero di giorni di cui si vogliono ricevere statistiche

Returns:

Le statistiche per più giorni

getStatisticsHourly

```
@GetMapping("/hourly")
public org.springframework.http.ResponseEntity<Object✉
> getStatisticsHourly(@RequestParam(name="city", defaultValue="Rome")
                       String✉ city,
                       @RequestParam(name="date")
                       String✉ date,
                       @RequestParam(name="from")
                       int from,
                       @RequestParam(name="to")
                       int to)
```

Restituisce e salva le statistiche per più ore di un giorno di una città.

Parameters:

city - rappresenta la città di cui si richiedono le statistiche. Valore di default: Rome,IT

`date` - Il giorno di cui si vogliono ricevere statistiche

`from` - prima ora

`to` - ultima ora

Returns:

Le statistiche per più ore

Package it.univpm.pressurestats.exception

package it.univpm.pressurestats.exception

Contiene le eccezioni personalizzate di PressureStats

Author:

Pietroniro Cristian, Settimi Diego

Exception Summary	
Exception	Description
CityStatisticsNotFoundException	Eccezione lanciata se le statistiche della città non sono state trovate.
DayNotFoundException	Eccezione lanciata se il giorno non è stato trovato.
IdNotFoundException	Eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta.
ItalianCityNotFoundException	Eccezione lanciata se la città non è italiana.
NegativeStartException	Eccezione lanciata se il valore di Start è inferiore a 0.
WrongHoursPeriodException	Eccezione lanciata se il range orario non è corretto.
WrongMultiplyException	Eccezione lanciata se il moltiplicatore non è ammesso (moltiplicatore minore o uguale a 0.02).

Package [it.univpm.pressurestats.exception](#)

Class CityStatisticsNotFoundException

java.lang.Object ↗
 java.lang.Throwable ↗
 java.lang.Exception ↗
 it.univpm.pressurestats.exception.CityStatisticsNotFoundException

All Implemented Interfaces:

[Serializable](#) ↗

```
public class CityStatisticsNotFoundException
extends Exception ↗
```

Eccezione lanciata se le statistiche della città non sono state trovate.

Author:

Pietroniro Cristian, Settimi Diego

See Also:

[Serialized Form](#)

Field Summary

Fields

Modifier and Type	Field	Description
private String ↗	message	Messaggio di errore
private static long	serialVersionUID	Seriale eccezione

Constructor Summary

Constructors

Constructor	Description
CityStatisticsNotFoundException (String ↗ message)	Costruttore della classe.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
String ↗	getMessage ()	Restituisce la stringa di errore dell'eccezione.

Methods inherited from class java.lang.Throwable

addSuppressed[✉], fillInStackTrace[✉], getCause[✉], getLocalizedMessage[✉],
getStackTrace[✉], getSuppressed[✉], initCause[✉], printStackTrace[✉],
printStackTrace[✉], printStackTrace[✉], setStackTrace[✉], toString[✉]

Methods inherited from class java.lang.Object

clone[✉], equals[✉], finalize[✉], getClass[✉], hashCode[✉], notify[✉], notifyAll[✉],
wait[✉], wait[✉], wait[✉]

Field Details

serialVersionUID

private static final long serialVersionUID

Seriiale eccezione

See Also:

Constant Field Values

message

private String[✉] message

Messaggio di errore

Constructor Details

CityStatisticsNotFoundException

public CityStatisticsNotFoundException(String[✉] message)

Costruttore della classe.

Parameters:

message - messaggio di errore

Method Details

getMessage

public String[✉] getMessage()

Restituisce la stringa di errore dell'eccezione.

Overrides:

[getMessage](#) in class [Throwable](#)

Returns:

il messaggio di errore

Package [it.univpm.pressurestats.exception](#)

Class DayNotFoundException

java.lang.Object ↗
 java.lang.Throwable ↗
 java.lang.Exception ↗
 it.univpm.pressurestats.exception.DayNotFoundException

All Implemented Interfaces:

[Serializable](#) ↗

```
public class DayNotFoundException
extends Exception ↗
```

Eccezione lanciata se il giorno non è stato trovato.

Author:

Pietroniro Cristian, Settimi Diego

See Also:

[Serialized Form](#)

Field Summary

Fields

Modifier and Type	Field	Description
private String ↗	message	Messaggio di errore
private static long	serialVersionUID	Seriale eccezione

Constructor Summary

Constructors

Constructor	Description
DayNotFoundException(String message)	Costruttore della classe.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
String ↗	getMessage()	Restituisce la stringa di errore dell'eccezione.

Methods inherited from class java.lang.Throwable

addSuppressed[✉], fillInStackTrace[✉], getCause[✉], getLocalizedMessage[✉],
getStackTrace[✉], getSuppressed[✉], initCause[✉], printStackTrace[✉],
printStackTrace[✉], printStackTrace[✉], setStackTrace[✉], toString[✉]

Methods inherited from class java.lang.Object

clone[✉], equals[✉], finalize[✉], getClass[✉], hashCode[✉], notify[✉], notifyAll[✉],
wait[✉], wait[✉], wait[✉]

Field Details

serialVersionUID

private static final long serialVersionUID

Seriale eccezione

See Also:

Constant Field Values

message

private String[✉] message

Messaggio di errore

Constructor Details

DayNotFoundException

public DayNotFoundException(String[✉] message)

Costruttore della classe.

Parameters:

message - messaggio di errore

Method Details

getMessage

public String[✉] getMessage()

Restituisce la stringa di errore dell'eccezione.

Overrides:

[getMessage](#) in class [Throwable](#)

Returns:

il messaggio di errore

Package [it.univpm.pressurestats.exception](#)

Class IdNotFoundException

java.lang.Object 
 java.lang.Throwable 
 java.lang.Exception 
 it.univpm.pressurestats.exception.IdNotFoundException

All Implemented Interfaces:

[Serializable](#) 

```
public class IdNotFoundException  
extends Exception
```

Eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta.

Author:

Pietroniro Cristian, Settimi Diego

See Also:

[Serialized Form](#)

Field Summary

Fields

Modifier and Type	Field	Description
private String 	message	Messaggio di errore
private static long	serialVersionUID	Seriale eccezione

Constructor Summary

Constructors

Constructor	Description
IdNotFoundException(String message)	Costruttore della classe.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
String 	getMessage()	Restituisce la stringa di errore dell'eccezione.

Methods inherited from class java.lang.Throwable

addSuppressed[✉], fillInStackTrace[✉], getCause[✉], getLocalizedMessage[✉],
getStackTrace[✉], getSuppressed[✉], initCause[✉], printStackTrace[✉],
printStackTrace[✉], printStackTrace[✉], setStackTrace[✉], toString[✉]

Methods inherited from class java.lang.Object

clone[✉], equals[✉], finalize[✉], getClass[✉], hashCode[✉], notify[✉], notifyAll[✉],
wait[✉], wait[✉], wait[✉]

Field Details

serialVersionUID

private static final long serialVersionUID

Seriale eccezione

See Also:

Constant Field Values

message

private String[✉] message

Messaggio di errore

Constructor Details

IdNotFoundException

public IdNotFoundException(String[✉] message)

Costruttore della classe.

Parameters:

message - messaggio di errore

Method Details

getMessage

public String[✉] getMessage()

Restituisce la stringa di errore dell'eccezione.

Overrides:

[getMessage](#) in class [Throwable](#)

Returns:

il messaggio di errore

Package [it.univpm.pressurestats.exception](#)

Class ItalianCityNotFoundException

```
java.lang.Object
  java.lang.Throwable
    java.lang.Exception
      it.univpm.pressurestats.exception.ItalianCityNotFoundException
```

All Implemented Interfaces:

[Serializable](#)

```
public class ItalianCityNotFoundException
extends Exception
```

Eccezione lanciata se la città non è italiana.

Author:

Pietroniro Cristian, Settimi Diego

See Also:

[Serialized Form](#)

Field Summary

Fields

Modifier and Type	Field	Description
(package private) String <code>message</code>		Messaggio di errore
private static long	<code>serialVersionUID</code>	Seriale eccezione

Constructor Summary

Constructors

Constructor	Description
<code>ItalianCityNotFoundException(String message)</code>	Costruttore della classe.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
String	<code>getMessage()</code>	Restituisce la stringa di errore dell'eccezione.

Methods inherited from class java.lang.Throwable

addSuppressed[✉], fillInStackTrace[✉], getCause[✉], getLocalizedMessage[✉],
getStackTrace[✉], getSuppressed[✉], initCause[✉], printStackTrace[✉],
printStackTrace[✉], printStackTrace[✉], setStackTrace[✉], toString[✉]

Methods inherited from class java.lang.Object

clone[✉], equals[✉], finalize[✉], getClass[✉], hashCode[✉], notify[✉], notifyAll[✉],
wait[✉], wait[✉], wait[✉]

Field Details

serialVersionUID

private static final long serialVersionUID

Seriale eccezione

See Also:

Constant Field Values

message

String[✉] message

Messaggio di errore

Constructor Details

ItalianCityNotFoundException

public ItalianCityNotFoundException(String[✉] message)

Costruttore della classe.

Parameters:

message - messaggio di errore

Method Details

getMessage

public String[✉] getMessage()

Restituisce la stringa di errore dell'eccezione.

Overrides:

[getMessage](#) in class [Throwable](#)

Returns:

il messaggio di errore

Package [it.univpm.pressurestats.exception](#)

Class NegativeStartException

java.lang.Object 
 java.lang.Throwable 
 java.lang.Exception 
 it.univpm.pressurestats.exception.NegativeStartException

All Implemented Interfaces:

[Serializable](#) 

```
public class NegativeStartException  
extends Exception
```

Eccezione lanciata se il valore di Start è inferiore a 0. (Negativo)

Author:

Pietroniro Cristian, Settimi Diego

See Also:

[Serialized Form](#)

Field Summary

Fields

Modifier and Type	Field	Description
(package private) String  message		Messaggio di errore
private static long	serialVersionUID	Seriale eccezione

Constructor Summary

Constructors

Constructor	Description
NegativeStartException (String  message)	Costruttore della classe.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
String 	getMessage ()	Restituisce la stringa di errore dell'eccezione.

Methods inherited from class java.lang.Throwable

addSuppressed[✉], fillInStackTrace[✉], getCause[✉], getLocalizedMessage[✉],
getStackTrace[✉], getSuppressed[✉], initCause[✉], printStackTrace[✉],
printStackTrace[✉], printStackTrace[✉], setStackTrace[✉], toString[✉]

Methods inherited from class java.lang.Object

clone[✉], equals[✉], finalize[✉], getClass[✉], hashCode[✉], notify[✉], notifyAll[✉],
wait[✉], wait[✉], wait[✉]

Field Details

serialVersionUID

private static final long serialVersionUID

Seriale eccezione

See Also:

Constant Field Values

message

String[✉] message

Messaggio di errore

Constructor Details

NegativeStartException

public NegativeStartException(String[✉] message)

Costruttore della classe.

Parameters:

message - messaggio di errore

Method Details

getMessage

public String[✉] getMessage()

Restituisce la stringa di errore dell'eccezione.

Overrides:

[getMessage](#) in class [Throwable](#)

Returns:

il messaggio di errore

Package [it.univpm.pressurestats.exception](#)

Class WrongHoursPeriodException

```
java.lang.Object
  java.lang.Throwable
    java.lang.Exception
      it.univpm.pressurestats.exception.WrongHoursPeriodException
```

All Implemented Interfaces:

[Serializable](#)

```
public class WrongHoursPeriodException
extends Exception
```

Eccezione lanciata se il range orario non è corretto.

Author:

Pietroniro Cristian, Settimi Diego

See Also:

[Serialized Form](#)

Field Summary

Fields

Modifier and Type	Field	Description
private String	message	Messaggio di errore.
private static long	serialVersionUID	Seriale eccezione

Constructor Summary

Constructors

Constructor	Description
WrongHoursPeriodException(String message)	Costruttore della classe.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
String	getMessage()	Restituisce la stringa di errore dell'eccezione.

Methods inherited from class java.lang.Throwable

addSuppressed[✉], fillInStackTrace[✉], getCause[✉], getLocalizedMessage[✉],
getStackTrace[✉], getSuppressed[✉], initCause[✉], printStackTrace[✉],
printStackTrace[✉], printStackTrace[✉], setStackTrace[✉], toString[✉]

Methods inherited from class java.lang.Object

clone[✉], equals[✉], finalize[✉], getClass[✉], hashCode[✉], notify[✉], notifyAll[✉],
wait[✉], wait[✉], wait[✉]

Field Details

serialVersionUID

private static final long serialVersionUID

Seriale eccezione

See Also:

Constant Field Values

message

private String[✉] message

Messaggio di errore.

Constructor Details

WrongHoursPeriodException

public WrongHoursPeriodException(String[✉] message)

Costruttore della classe.

Parameters:

message - messaggio di errore

Method Details

getMessage

public String[✉] getMessage()

Restituisce la stringa di errore dell'eccezione.

Overrides:

[getMessage](#) in class [Throwable](#)

Returns:

il messaggio di errore

Package [it.univpm.pressurestats.exception](#)

Class WrongMultiplyException

java.lang.Object 
 java.lang.Throwable 
 java.lang.Exception 
 it.univpm.pressurestats.exception.WrongMultiplyException

All Implemented Interfaces:

[Serializable](#) 

```
public class WrongMultiplyException  
extends Exception
```

Eccezione lanciata se il moltiplicatore non è ammesso (moltiplicatore minore o uguale a 0.02). Un moltiplicatore di 0.02 restituirebbe dati ogni minuto circa.

Author:

Pietroniro Cristian, Settimi Diego

See Also:

[Serialized Form](#)

Field Summary

Fields

Modifier and Type	Field	Description
(package private) String  message		Messaggio di errore
private static long	serialVersionUID	Seriale eccezione

Constructor Summary

Constructors

Constructor	Description
WrongMultiplyException(String  message)	Costruttore della classe.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
String 	getMessage()	Restituisce la stringa di errore dell'eccezione.

Methods inherited from class java.lang.Throwable

addSuppressed[✉], fillInStackTrace[✉], getCause[✉], getLocalizedMessage[✉],
getStackTrace[✉], getSuppressed[✉], initCause[✉], printStackTrace[✉],
printStackTrace[✉], printStackTrace[✉], setStackTrace[✉], toString[✉]

Methods inherited from class java.lang.Object

clone[✉], equals[✉], finalize[✉], getClass[✉], hashCode[✉], notify[✉], notifyAll[✉],
wait[✉], wait[✉], wait[✉]

Field Details

serialVersionUID

private static final long serialVersionUID

Seriale eccezione

See Also:

Constant Field Values

message

String[✉] message

Messaggio di errore

Constructor Details

WrongMultiplyException

public WrongMultiplyException(String[✉] message)

Costruttore della classe.

Parameters:

message - messaggio di errore

Method Details

getMessage

public String[✉] getMessage()

Restituisce la stringa di errore dell'eccezione.

Overrides:

[getMessage](#) in class [Throwable](#)

Returns:

il messaggio di errore

Package it.univpm.pressurestats.model

package it.univpm.pressurestats.model

Contiene tutta la parte di modellazione delle classi

Author:

Pietroniro Cristian, Settimi Diego

Class Summary	
Class	Description
City	Contiene le proprietà della città e le relative previsioni.
Forecast	Contiene i dati di pressione, visibilità e la data della loro misurazione.

Package [it.univpm.pressurestats.model](#)

Class City

[java.lang.Object](#)
[it.univpm.pressurestats.model.City](#)

```
public class City
extends Object
```

Contiene le proprietà della città e le relative previsioni.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
private String	country	Nazione città
private long	id	Identificativo univoco città
private double	lat	Latitudine città
private double	lon	Longitudine città
private String	name	Nome città
private Vector < Forecast >	weather	Vettore contenente varie misurazioni meteo della città

Constructor Summary

Constructors

Constructor	Description
City()	Costruttore della classe
City(String name)	Costruttore della classe con nome come parametro.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
-------------------	--------	-------------

boolean	<code>equals (Object[✉] obj)</code>	Confronta una città con un'altra
String[✉]	<code>getCountry ()</code>	Restituisce nazione della città
long	<code>getId ()</code>	Restituisce id della città
double	<code>getLat ()</code>	Restituisce latitudine della città
double	<code>getLon ()</code>	Restituisce longitudine della città
String[✉]	<code>getName ()</code>	Restituisce nome della città
Vector[✉]<Forecast>	<code>getWeather ()</code>	Restituisce previsioni meteo della città
void	<code>setCountry (String[✉] country)</code>	Imposta nuova nazione della città
void	<code>setId (long id)</code>	Imposta nuovo id della città
void	<code>setLat (double lat)</code>	Imposta nuova latitudine della città
void	<code>setLon (double lon)</code>	Imposta nuova longitudine della città
void	<code>setName (String[✉] name)</code>	Imposta nuovo nome della città
void	<code>setWeather (Vector[✉] <Forecast> weather)</code>	Imposta nuove previsioni meteo della città
String[✉]	<code>toString ()</code>	Restituisce la città sotto forma di Stringa

Methods inherited from class java.lang.Object[✉]

`clone✉, finalize✉, getClass✉, hashCode✉, notify✉, notifyAll✉, wait✉, wait✉, wait✉`

Field Details

id

`private long id`

Identificativo univoco città

name

`private String✉ name`

Nome città

country

```
private String2 country
```

Nazione città

lat

```
private double lat
```

Latitudine città

lon

```
private double lon
```

Longitudine città

weather

```
private Vector2<Forecast> weather
```

Vettore contenente varie misurazioni meteo della città

Constructor Details

City

```
public City()
```

Costruttore della classe. *

City

```
public City(String2 name)
```

Costruttore della classe con nome come parametro.

Parameters:

name - nome della città

Method Details

getId

```
public long getId()
```

Restituisce id della città

Returns:

id

setId

```
public void setId(long id)
```

Imposta nuovo id della città

Parameters:

id - nuovo id

getName

```
public String1 getName()
```

Restituisce nome della città

Returns:

name

setName

```
public void setName(String1 name)
```

Imposta nuovo nome della città

Parameters:

name - nuovo nome

getCountry

```
public String1 getCountry()
```

Restituisce nazione della città

Returns:

country

setCountry

```
public void setCountry(String1 country)
```

Imposta nuova nazione della città

Parameters:

country - nuova nazione

getLat

```
public double getLat()
```

Restituisce latitudine della città

Returns:

lat

setLat

```
public void setLat(double lat)
```

Imposta nuova latitudine della città

Parameters:

lat - nuova latitudine

getLon

```
public double getLon()
```

Restituisce longitudine della città

Returns:

lon

setLon

```
public void setLon(double lon)
```

Imposta nuova longitudine della città

Parameters:

lon - nuova longitudine

getWeather

```
public Vector<Forecast> getWeather()
```

Restituisce previsioni meteo della città

Returns:

weather

setWeather

```
public void setWeather(Vector<Forecast> weather)
```

Imposta nuove previsioni meteo della città

Parameters:

weather - nuove previsioni meteo

toString

```
public String toString()
```

Restituisce la città sotto forma di Stringa

Overrides:

[toString](#) in class [Object](#)

Returns:

String city

equals

```
public boolean equals(Object obj)
```

Confronta una città con un'altra

Overrides:

[equals](#) in class [Object](#)

Parameters:

obj - la città con cui confrontare

Returns:

true || false

Package [it.univpm.pressurestats.model](#)

Class Forecast

[java.lang.Object](#)
[it.univpm.pressurestats.model.Forecast](#)

```
public class Forecast
extends Object
```

Contiene i dati di pressione, visibilità e la data della loro misurazione.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
private	String date	Data misurazione in formato standard DD-MM-YYYY
private long	dt	Data misurazione in formato Unix epoch (long) (Secondi passati dal 1 gennaio 1970)
private long	pressure	Pressione misurata in hPa
private long	visibility	Visibilità misurata in km

Constructor Summary

Constructors

Constructor	Description
Forecast()	Costruttore della classe.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
boolean	equals(Object obj)	Confronta una previsione con un'altra
String	getDate()	Restituisce data misurazione in formato standard

long	getDt()	Restituisce data misurazione in formato Unix epoch (long) (Secondi passati dal 1 gennaio 1970)
long	getPressure()	Restituisce pressione
long	getVisibility()	Restituisce visibilità
void	 setDate (String[¶] date)	Imposta nuova data misurazione in formato standard
void	setDt (long dt)	Imposta nuova data misurazione in formato Unix epoch (long) (Secondi passati dal 1 gennaio 1970)
void	setPressure (long pressure)	Imposta nuovo valore di pressione
void	setVisibility (long visibility)	Imposta nuovo valore di visibilità
String[¶]	toString()	Restituisce le previsioni sotto forma di Stringa

Methods inherited from class java.lang.Object[¶]

`clone¶, finalize¶, getClass¶, hashCode¶, notify¶, notifyAll¶, wait¶, wait¶, wait¶`

Field Details

pressure

`private long pressure`

Pressione misurata in hPa

visibility

`private long visibility`

Visibilità misurata in km

dt

`private long dt`

Data misurazione in formato Unix epoch (long) (Secondi passati dal 1 gennaio 1970)

date

```
private String¶ date
```

Data misurazione in formato standard DD-MM-YYYY

Constructor Details

Forecast

```
public Forecast()
```

Costruttore della classe.

Method Details

getPressure

```
public long getPressure()
```

Restituisce pressione

Returns:

pressure

setPressure

```
public void setPressure(long pressure)
```

Imposta nuovo valore di pressione

Parameters:

pressure - nuovo valore di pressione

getVisibility

```
public long getVisibility()
```

Restituisce visibilità

Returns:

visibility

setVisibility

```
public void setVisibility(long visibility)
```

Imposta nuovo valore di visibilità

Parameters:

visibility - nuovo valore di visibilità

getDt

```
public long getDt()
```

Restituisce data misurazione in formato Unix epoch (long) (Secondi passanti dal 1 gennaio 1970)

Returns:

dt

setDt

```
public void setDt(long dt)
```

Imposta nuova data misurazione in formato Unix epoch (long) (Secondi passanti dal 1 gennaio 1970)

Parameters:

dt - nuova data

getDate

```
public String¶ getDate()
```

Restituisce data misurazione in formato standard

Returns:

date

setDate

```
public void setDate(String¶ date)
```

Imposta nuova data misurazione in formato standard

Parameters:

date - nuova data

toString

```
public String¶ toString()
```

Restituisce le previsioni sotto forma di Stringa

Overrides:

`toString` in class `Object`

Returns:

`String forecast`

equals

`public boolean equals(Object obj)`

Confronta una previsione con un'altra

Overrides:

`equals` in class `Object`

Parameters:

`obj` - la previsione con cui confrontare

Returns:

`true || false`

Package it.univpm.pressurestats.service

package it.univpm.pressurestats.service

Contiene i servizi e le operazioni usate per raccogliere, salvare e mostrare i dati.

Author:

Pietroniro Cristian, Settimi Diego

Interface Summary

Interface	Description
Service	Interfaccia contenente i metodi implementati da ServiceImpl e richiamati dal Controller.

Class Summary

Class	Description
ServiceImpl	Implementazione dell'interfaccia Service.

Package [it.univpm.pressurestats.service](#)

Interface Service

All Known Implementing Classes:

[ServiceImpl](#)

```
public interface Service
```

Interfaccia contenente i metodi implementati da [ServiceImpl](#) e richiamati dal Controller.

Author:

Pietroniro Cristian, Settimi Diego

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method	Description
<code>City</code>	<code>getForecast</code> (<code>org.json.simple.JSONObject</code>)	Utilizza <code>getJSONForecast</code> per prendere le informazioni di visibilità e pressione utilizzate per le statistiche.
<code>org.json.simple.JSONObject</code>	<code>getJSONForecast(String name, boolean isObject)</code>	Scarica da OpenWeather le previsioni meteo di una città data..
<code>org.json.simple.JSONObject</code>	<code>readFile(String city)</code>	Legge dati da un file salvato in precedenza
<code>String</code>	<code>saveMessage(double multiplier)</code>	A partire dal multiplier, prepara una stringa contenente la frequenza di aggiornamento del salvataggio.
<code>void</code>	<code>saveToFile(JSONObject obj)</code>	Richiama <code>getForecast(JSONObject obj)</code> e salva i dati su un file di testo (salvato in <code>/src/main/resources/</code>)
<code>void</code>	<code>saveToFileHourly(String id, double multiplier, long start)</code>	Salva ogni tot ore su un file i dati delle previsioni utili richiamando <code>getJSONForecast(String id, boolean isObject)</code> e <code>saveToFile(JSONObject object)</code>

```
org.json.simple.JSONObject toJSON(City city)
```

Converte l'oggetto di tipo City in un JSONObject.

Method Details

toJSON

```
org.json.simple.JSONObject toJSON(City city)
```

Converte l'oggetto di tipo City in un JSONObject.

Parameters:

city - città di cui si vogliono sapere le previsioni

Returns:

Un JSONObject (forecast) contenente le previsioni

getJSONForecast

```
org.json.simple.JSONObject getJSONForecast(String name,  
                                         boolean isObject)  
                           throws IdNotFoundException
```

Scarica da OpenWeather le previsioni meteo di una città data..

Parameters:

name - nome della città da cercare

isObject - controllo di tipo Boolean (COMPLETARE)

Returns:

un JSONObject contenente le previsioni meteo complete.

Throws:

`IdNotFoundException` - eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta

getForecast

```
City getForecast(org.json.simple.JSONObject forecast)  
                  throws ItalianCityNotFoundException
```

Utilizza getJSONForecast per prendere le informazioni di visibilità e pressione utilizzate per le statistiche.

Parameters:

forecast - JSONObject contenente le previsioni restituito da getJSONForecast(String id, boolean isObject)

Returns:

un oggetto di tipo City che contiene tutte le informazioni richieste e anche le informazioni sulla città.

Throws:

`ItalianCityNotFoundException` - eccezione lanciata se la città non è italiana

saveToFile

```
void saveToFile(org.json.simple.JSONObject obj)
    throws ItalianCityNotFoundException
```

Richiama `getForecast(JSONObject obj)` e salva i dati su un file di testo (salvato in `/src/main/resources/`)

Parameters:

`obj` - `JSONObject` contenente le previsioni restituito da `getJSONForecast(String id, boolean isObject)`

Throws:

`ItalianCityNotFoundException` - eccezione lanciata se la città non è italiana

saveToFileHourly

```
void saveToFileHourly(String id,
                      double multiplier,
                      long start)
    throws ItalianCityNotFoundException,
           WrongMultiplyException,
           NegativeStartException,
           IdNotFoundException
```

Salva ogni tot ore su un file i dati delle previsioni utili richiamando `getJSONForecast(String id, boolean isObject)` e `saveToFile(JSONObject object)`

Parameters:

`id` - id della città di cui salvare i dati ogni tot ore

`multiplier` - moltiplicatore dell'ora per cambiare la frequenza di salvataggio. (Es. Impostando il valore a 2, il salvataggio avviene ogni 2 ore, a 0.5 ogni 30 minuti).

`start` - millisecondo di partenza del salvataggio

Throws:

`ItalianCityNotFoundException` - eccezione lanciata se la città non è italiana

`WrongMultiplyException` - eccezione lanciata se il moltiplicatore non è ammesso (moltiplicatore minore o uguale a 0.02). Un moltiplicatore di 0.02 restituirebbe dati ogni minuto circa.

`NegativeStartException` - eccezione lanciata se il valore di Start è inferiore a 0. (Negativo)

`IdNotFoundException` - eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta

readFile

```
org.json.simple.JSONArray readFile(String¶ city)
```

Legge dati da un file salvato in precedenza

Parameters:

city - città di cui leggere i dati

Returns:

JSONArray contenente i dati letti

saveMessage

```
String¶ saveMessage(double multiplier)
```

A partire dal multiplier, prepara una stringa contenente la frequenza di aggiornamento del salvataggio.

Parameters:

multiplier - moltiplicatore dell'ora per cambiare la frequenza di salvataggio. (Es. Impostando il valore a 2, il salvataggio avviene ogni 2 ore, a 0.5 ogni 30 minuti).

Returns:

un messaggio contenente la frequenza di aggiornamento

Package [it.univpm.pressurestats.service](#)

Class ServiceImpl

[java.lang.Object](#)
[it.univpm.pressurestats.service.ServiceImpl](#)

All Implemented Interfaces:

[Service](#)

```
@Service
public class ServiceImpl
extends Object
implements Service
```

Implementazione dell'interfaccia Service. Contiene i metodi utilizzati dal controller.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
private String	apiKey	API Key
static String	dir	Percorso di salvataggio del file
static String	f_type	Suffisso nome file ed estensione
(package private) org.json.simple.JSONObject	forecast	Rappresenta la previsione
(package private) static long	hour	Ora in millisecondi, costante

Constructor Summary

Constructors

Constructor	Description
ServiceImpl()	Costruttore della classe

Method Summary

[All Methods](#)

[Instance Methods](#)

[Concrete Methods](#)

Modifier and Type	Method	Description
City	getForecast (org.json.simple.JSONObject id, boolean isObject)	Utilizza getJSONForecast per prendere le informazioni di visibilità e pressione utilizzate per le statistiche.
org.json.simple.JSONObject	getJSONForecast (String id, boolean isObject)	Scarica da OpenWeather le previsioni meteo di una città data..
org.json.simple.JSONObject	readFile (String city)	Legge dati da un file salvato in precedenza
String	saveMessage (double multiplier)	A partire dal multiplier, prepara una stringa contenente la frequenza di aggiornamento del salvataggio.
void	saveToFile (org.json.simple.JSONObject obj)	Richiama getForecast(JSONObject obj) e salva i dati su un file di testo (salvato in /src/main/resources/)
void	saveToFileHourly (String id, double multiplier, long start)	Salva ogni tot ore su un file i dati delle previsioni utili richiamando getJSONForecast(String id, boolean isObject) e saveToFile(JSONObject object)
org.json.simple.JSONObject	toJSON (City city)	Converte l'oggetto di tipo City in un JSONObject.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Details

apiKey

private String apiKey

API Key

forecast

```
org.json.simple.JSONObject forecast
```

Rappresenta la previsione

dir

```
public static String2 dir
```

Percorso di salvataggio del file

f_type

```
public static String2 f_type
```

Suffisso nome file ed estensione

hour

```
static final long hour
```

Ora in millisecondi, costante

See Also:

[Constant Field Values](#)

Constructor Details

ServiceImpl

```
public ServiceImpl()
```

Costruttore della classe

Method Details

toJSON

```
public org.json.simple.JSONObject toJSON(City city)
```

Description copied from interface: Service

Converte l'oggetto di tipo City in un JSONObject.

Specified by:

[toJSON in interface Service](#)

Parameters:

city - città di cui si vogliono sapere le previsioni

Returns:

Un JSONObject (forecast) contenente le previsioni

getJSONForecast

```
public org.json.simple.JSONObject getJSONForecast(String id,  
                                                boolean isObject)  
throws IdNotFoundException
```

Description copied from interface: Service

Scarica da OpenWeather le previsioni meteo di una città data..

Specified by:

getJSONForecast in interface Service

Parameters:

id - nome della città da cercare

isObject - controllo di tipo Boolean (COMPLETARE)

Returns:

un JSONObject contenente le previsioni meteo complete.

Throws:

IdNotFoundException - eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta

getForecast

```
public City getForecast(org.json.simple.JSONObject obj)  
throws ItalianCityNotFoundException
```

Description copied from interface: Service

Utilizza getJSONForecast per prendere le informazioni di visibilità e pressione utilizzate per le statistiche.

Specified by:

getForecast in interface Service

Parameters:

obj - JSONObject contenente le previsioni restituito da getJSONForecast(String id, boolean isObject)

Returns:

un oggetto di tipo City che contiene tutte le informazioni richieste e anche le informazioni sulla città.

Throws:

ItalianCityNotFoundException - eccezione lanciata se la città non è italiana

saveToFile

```
public void saveToFile(org.json.simple.JSONObject object)
    throws ItalianCityNotFoundException
```

Description copied from interface: Service

Richiama getForecast(JSONObject obj) e salva i dati su un file di testo (salvato in /src/main/resources/)

Specified by:

saveToFile in interface Service

Parameters:

object - JSONObject contenente le previsioni restituito da getJSONForecast(String id, boolean isObject)

Throws:

ItalianCityNotFoundException - eccezione lanciata se la città non è italiana

saveToFileHourly

```
public void saveToFileHourly(String id,
                             double multiplier,
                             long start)
    throws ItalianCityNotFoundException,
           WrongMultiplyException,
           NegativeStartException,
           IdNotFoundException
```

Description copied from interface: Service

Salva ogni tot ore su un file i dati delle previsioni utili richiamando getJSONForecast(String id, boolean isObject) e saveToFile(JSONObject object)

Specified by:

saveToFileHourly in interface Service

Parameters:

id - id della città di cui salvare i dati ogni tot ore

multiplier - moltiplicatore dell'ora per cambiare la frequenza di salvataggio. (Es. Impostando il valore a 2, il salvataggio avviene ogni 2 ore, a 0.5 ogni 30 minuti).

start - millisecondo di partenza del salvataggio

Throws:

ItalianCityNotFoundException - eccezione lanciata se la città non è italiana

WrongMultiplyException - eccezione lanciata se il moltiplicatore non è ammesso (moltiplicatore minore o uguale a 0.02). Un moltiplicatore di 0.02 restituirebbe dati ogni minuto circa.

NegativeStartException - eccezione lanciata se il valore di Start è inferiore a 0. (Negativo)

IdNotFoundException - eccezione lanciata se non è stato trovato nessun ID corrispondente alla richiesta

readFile

```
public org.json.simple.JSONArray readFile(String¶ city)
```

Description copied from interface: Service

Legge dati da un file salvato in precedenza

Specified by:

`readFile` in interface `Service`

Parameters:

`city` - città di cui leggere i dati

Returns:

`JSONArray` contenente i dati letti

saveMessage

```
public String¶ saveMessage(double multiplier)
```

Description copied from interface: Service

A partire dal `multiplier`, prepara una stringa contenente la frequenza di aggiornamento del salvataggio.

Specified by:

`saveMessage` in interface `Service`

Parameters:

`multiplier` - moltiplicatore dell'ora per cambiare la frequenza di salvataggio. (Es. Impostando il valore a 2, il salvataggio avviene ogni 2 ore, a 0.5 ogni 30 minuti).

Returns:

un messaggio contenente la frequenza di aggiornamento

Package it.univpm.pressurestats.statistics

package it.univpm.pressurestats.statistics

Contiene le statistiche e metodi per filtrarle e presentarle

Author:

Pietroniro Cristian, Settimi Diego

Class Summary	
Class	Description
Filters	Gestisce i filtri.
Statistics	Permette di calcolare e salvare le statistiche.

Package [it.univpm.pressurestats.statistics](#)

Class Filters

[java.lang.Object](#)
[it.univpm.pressurestats.statistics.Filters](#)

```
public class Filters
extends Object
```

Gestisce i filtri.

Author:

Pietroniro Cristian, Settimi Diego

Constructor Summary

Constructors

Constructor	Description
Filters ()	Costruttore della classe.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
org.json.simple.JSONArray	hourly (String city, String day, int from, int to)	Prepara un JSONArray delle statistiche orarie di un giorno di una città.
org.json.simple.JSONArray	moreDayWeather (String city, int numDays)	Prepara un JSONArray delle statistiche filtrate dal giorno attuale a tot giorni passati.
org.json.simple.JSONArray	oneDayWeather (String city, String day)	Prepara un JSONArray delle statistiche di un solo giorno di una città.

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[toString](#), [wait](#), [wait](#), [wait](#)

Constructor Details

Filters

```
public Filters()
```

Costruttore della classe.

Method Details

oneDayWeather

```
public org.json.simple.JSONArray oneDayWeather(String city,  
                                              String day)  
throws  
DayNotFoundException,  
CityStatisticsNotFoundException
```

Prepara un JSONArray delle statistiche di un solo giorno di una città.

Parameters:

city - città scelta

day - giorno di cui filtrare statistiche

Returns:

Le statistiche di un giorno in un JSONArray.

Throws:

`DayNotFoundException` - eccezione lanciata se il giorno non è stato trovato

`CityStatisticsNotFoundException` - eccezione lanciata se le statistiche della città non sono state trovate

moreDayWeather

```
public org.json.simple.JSONArray moreDayWeather(String city,  
                                                int numDays)  
throws  
CityStatisticsNotFoundException
```

Prepara un JSONArray delle statistiche filtrate dal giorno attuale a tot giorni passati.

Parameters:

city - città scelta

numDays - numeri di giorni di cui filtrare statistiche

Returns:

Le statistiche di più giorni in un JSONArray.

Throws:

`CityStatisticsNotFoundException` - eccezione lanciata se le statistiche della città non sono

state trovate

hourly

```
public org.json.simple.JSONArray hourly(String2 city,  
                                         String2 day,  
                                         int from,  
                                         int to)  
throws WrongHoursPeriodException,  
      CityStatisticsNotFoundException,  
      DayNotFoundException
```

Prepara un JSONArray delle statistiche orarie di un giorno di una città.

Parameters:

city - città scelta

day - giorno di cui filtrare statistiche

from - prima ora

to - ultima ora

Returns:

Le statistiche di più ore in un JSONArray.

Throws:

WrongHoursPeriodException - eccezione lanciata se il range orario non è corretto

CityStatisticsNotFoundException - eccezione lanciata se le statistiche della città non sono state trovate

DayNotFoundException - eccezione lanciata se il giorno non è stato trovato

Package [it.univpm.pressurestats.statistics](#)

Class Statistics

[java.lang.Object](#)
[it.univpm.pressurestats.statistics.Statistics](#)

```
public class Statistics  
extends Object
```

Permette di calcolare e salvare le statistiche.

Author:

Pietroniro Cristian, Settimi Diego

Field Summary

Fields

Modifier and Type	Field	Description
private String	city	Nome città di cui calcolare statistiche.
private String	date	Data di cui calcolare statistiche.
private Filters	od	Permette di usare i filtri e le loro operazioni.
private org.json.simple.JSONArray	stats	JSONArray in cui vengono salvate le statistiche

Constructor Summary

Constructors

Constructor	Description
Statistics(String city, int numDays)	Costruttore della classe.
Statistics(String city, String day)	Costruttore della classe.
Statistics(String city, String day, int from, int to)	Costruttore della classe.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
-------------------	--------	-------------

void	saveToFile	Salva le statistiche calcolate su (org.json.simple.JSONObject) un file di testo.
org.json.simple.JSONObject	stats()	Calcola le statistiche.

Methods inherited from class java.lang.Object

clone[✉], equals[✉], finalize[✉], getClass[✉], hashCode[✉], notify[✉], notifyAll[✉],
toString[✉], wait[✉], wait[✉], wait[✉]

Field Details

stats

private org.json.simple.JSONArray stats

JSONArray in cui vengono salvate le statistiche

od

private Filters od

Permette di usare i filtri e le loro operazioni.

city

private String[✉] city

Nome città di cui calcolare statistiche.

date

private String[✉] date

Data di cui calcolare statistiche.

Constructor Details

Statistics

```
public Statistics(String✉ city,  
                  String✉ day)  
    throws DayNotFoundException,  
          CityStatisticsNotFoundException
```

Costruttore della classe.

Parameters:

city - città di cui calcolare statistiche

day - giorno di cui calcolare statistiche

Throws:

DayNotFoundException - eccezione lanciata se il giorno non è stato trovato

CityStatisticsNotFoundException - eccezione lanciata se le statistiche della città non sono state trovate

Statistics

```
public Statistics(String¶ city,
                  String¶ day,
                  int from,
                  int to)
                  throws WrongHoursPeriodException,
                         CityStatisticsNotFoundException,
                         DayNotFoundException
```

Costruttore della classe.

Parameters:

city - città di cui calcolare statistiche

day - giorno di cui calcolare statistiche

from - ora da cui partire

to - ora a cui arrivare

Throws:

WrongHoursPeriodException - eccezione lanciata se il range orario non è corretto

CityStatisticsNotFoundException - eccezione lanciata se le statistiche della città non sono state trovate.

DayNotFoundException - eccezione lanciata se il giorno non è stato trovato

Statistics

```
public Statistics(String¶ city,
                  int numDays)
                  throws CityStatisticsNotFoundException
```

Costruttore della classe.

Parameters:

city - città di cui calcolare statistiche

numDays - numero di giorni di cui calcolare statistiche

Throws:

`CityStatisticsNotFoundException` - eccezione lanciata se le statistiche della città non sono state trovate.

Method Details

stats

```
public org.json.simple.JSONObject stats()
```

Calcola le statistiche.

Returns:

Un JSONObject contenente statistiche

saveToFile

```
public void saveToFile(org.json.simple.JSONObject object)
```

Salva le statistiche calcolate su un file di testo.

Parameters:

`object` - JSONObject contenente i dati da salvare