

be.kdg.se3.proxy

Class IncidentServiceProxy

java.lang.Object
be.kdg.se3.proxy.IncidentServiceProxy

```
public class IncidentServiceProxy  
extends java.lang.Object
```

Synchronous proxy for an HTTP based incident simulation service used by ICT systems in the Antwerp harbour
The proxy has two functions each with it's own URL: generation of random incidents and status of incidents that have 'occured'

Simulation

A call to `www.services4se3.com/incidentservice/simulate/shipid` will return a JSON with incident info regarding the specified shipid (a 7 digit number)
(example: `www.services4se3.com/incidentservice/simulate/1234567` will return an incident for the the ship with IMO number "IMO1234567")
The normal JSON response has the following structure

```
{  
  IMO: the IMO number of the ship (= the string 'IMO' followed by the 7 digits),  
  incidentType: the type of incident (string) ,  
}
```

An error JSON response indicates that the call succeeded but no sensible data could be returned. It has the following format

```
{  
  error: the type of error e.g. "Unknown shipid" (string),  
  description: a description of the error (string)  
}
```

Status

A call to `www.services4se3.com/incidentservice/status/shipid` will return a JSON with info regarding the status of the incident for the shipid (a 7 digit number)
(example: `www.services4se3.com/incidentservice/status/1234567` will return if the incident for the the ship with IMO number "IMO1234567" is still going on or not)
The normal JSON response has the following structure

```
{  
  IMO: the IMO number of the ship (= the string 'IMO' followed by the 7 digits),  
  incidentStatus: true if the incident is ongoing, false if it is solved ,  
}
```

An error JSON response indicates that the call succeeded but no sensible data could be returned. It has the following format

```
{  
  error: the type of error e.g. "Unknown shipid" (string),  
  description: a description of the error (string)  
}
```

This proxy is meant for testing purposes:

- the **URL does not actually exist** and no real HTTP call is made; the proxy just holds the calling thread for 200 ms to simulate network latency
- the service returns random incident types
- when asked for the status of an incident, the services returns random true or false except for ids 1111111, 1111112 and 1111113 where 9 times true followed by a false is returned
- for shipid 1111111 the incident will not be random but always 'schade'
- for shipid 1111112 the incident will not be random but always 'schade'

- for shipid 1111113 the incident will not be random but always 'man over board'
- shipid 2222222 wil generate a JSON error result ('Unknown shipid')
- shipid 3333333 will force an IOException to simulate a communication error

Constructor Summary

Constructors

Constructor and Description

<code>IncidentServiceProxy()</code>

Method Summary

Methods

Modifier and Type	Method and Description
java.lang.String	get (java.lang.String url) Invokes the remote service through a HTTP GET using the supplied URL.

Methods inherited from class java.lang.Object

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

Constructor Detail

IncidentServiceProxy

```
public IncidentServiceProxy()
```

Method Detail

get

```
public java.lang.String get(java.lang.String url)
                        throws java.io.IOException
```

Invokes the remote service through a HTTP GET using the supplied URL. The HTTP response is unpacked and the body is returned as a JSON string

Parameters:

url - the endpoint of the call (see class documentation)

Returns:

the data in the correct JSON format (see class documentation)

Throws:

`java.io.IOException` - if invocation fails (wrong URL or unexpected communication/service failure)

[Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

[Summary](#): [Nested](#) | [Field](#) | [Constr](#) | [Method](#) [Detail](#): [Field](#) | [Constr](#) | [Method](#)
