Contract Management System

Util Class
——Exception Handling

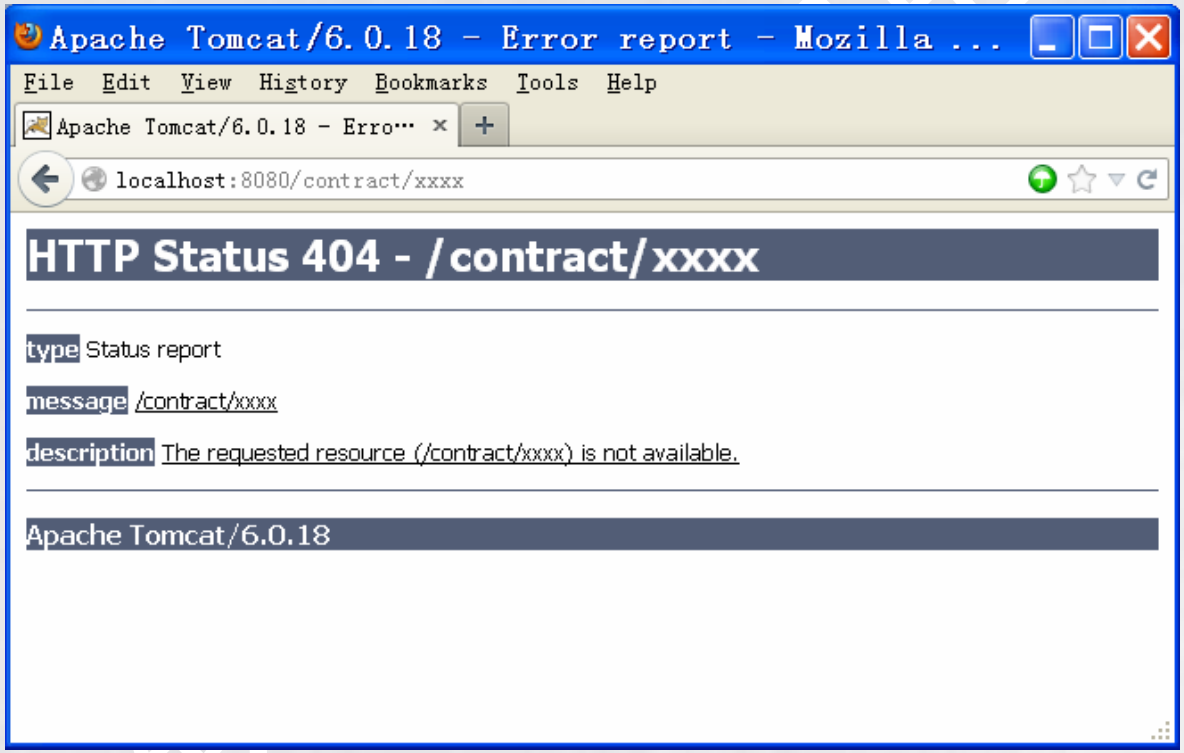
Contents

- Function
- Design
- Implementation

Copyright Declaration

Function

In the process of project development, some uncertain condition might be the cause of runtime error. For example, you get a 0 divisor after a series of computation; database is disconnected due to a sudden network breakdown, etc. If such error has not been promptly handled, it could cause program exception or breakdown. Just like the interface below, it's quite unfriendly to the user.



Function

In order to solve such problem and to catch runtime error, a user-defined exception is used to describe and transfer errant message. The purpose is to locate error and debug program.

If an error occurs, it is processed in view layer and the page turns to an exceptional page, error.jsp, with exceptional message posted.

Define a exception class AppException under project to carry and transfer exceptional message. The processing is designed in view layer.

1. Format of exception message

Format is exception code+ a complete class path of method that takes error.function name.

Example, 0 com.ruanko.service.UserService.login.

2. AppException class

(1) The attributes of private data member are exception code and message.

```
private int exceptionCode; //Exception code
private String message;     //Exception message
```

(2) Constructor that takes arguments sets exception message and code.

(3) Data members return exception message and code.

Design

The program generates exceptions of various type during runtime. If the exceptions are all thrown to the upper layer, the upper layer gathers exceptions of multiple type. That makes the upper layer even rely on the definition of concrete exception classes.

If method in data access layer has error in its SQL statement, SQLException will be generated. If the exception is transferred to business logic layer for processing, you need to import SQLException class. That's what dependency means.

In order to make it easy for the system to manage and locate exception, and to describe exceptional message, we use the following ways.

(1) Exception description mechanism. code + exception message.

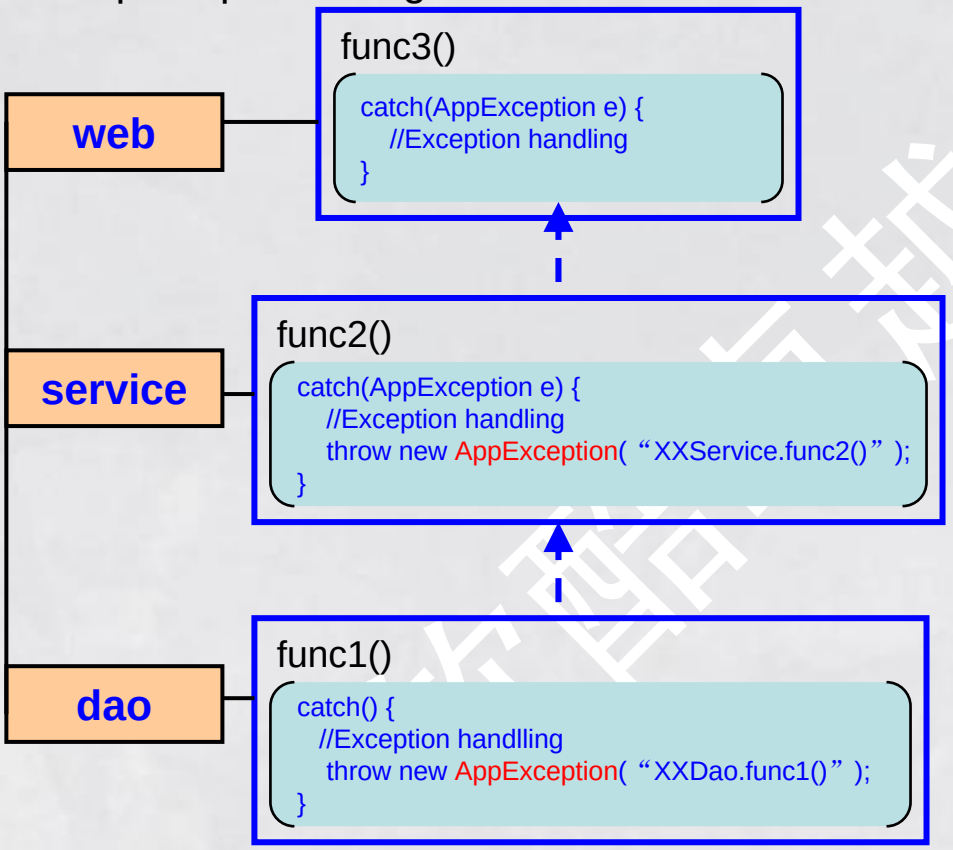
(2) User-defined exception class. After different types of exceptions are captured inside layers, user-defined exceptions are thrown upward instead with the exceptional messages being carried.

Throw the exception class that carries specific exception message upward from data access layer to view layer for processing.

1.Exception handling design

Try…catch is used to capture exceptions from view,business logic and data access layers, and then user-defined exceptions are thrown upward instead to view layer for further processing.

Define AppException class in com.ruanko.utils under project contract, as the tool class of exception processing.



Call fun2() of business logic layer in view layer or web, to catch user-defined exception thrown from business logic layer for processing. Export the request to the page where exceptional message is posted.

Call fun1() of data access layer in fun2() of business logic layer, to catch the user-defined exception thrown from data access layer and throw the user-defined exception upward. The exception message includes class name and method name.

If there is an exception in the function of data access layer, it throws user-defined exception. The exception message includes class name and method name.

2. AppException class design

Create AppException class in com.ruanko.utils, which inherits from Exception.

The exception class object is generated dynamically in the process of program running, so no Set method is used to assign value to the data member. When constructing user-defined exception object, the constructor is used directly to assign value to exception code and message.

(1) Data members

Declare two private data members indicating exception code and message each.

```
private int exceptionCode; //Exception number
private String message;    //Exception information
```

(2) Methods

Method 1, public AppException(String message)

Function: constructor, sets exception message.

Parameter: message, exceptional message.

```
public AppException(String message) {
    this.message = message;
}
```

Design

Method 2: public AppException(String message,int exceptionCode)

Function: constructor, sets exception message and exception code.

Parameter:

message: message of the exception.

exceptionCode: code of the exception.

Method 3: public int getExceptionCode()

Function: gets exception code.

Method 4: public String getMessage()

Function: gets detailed exceptional message. The detailed message can be either exception code and exception message, or user-defined message.

Implementation

Iterative development on the basis of Entity Class Design.

Firstly, write user-defined exception class AppException according to design idea and requirement.

Secondly, illustrate the operation of AppException.

Step 1, create AppException class.

Step 2, AppException Handling.

www.ruanko.com

Thanks

Exception Handling