

Xử lý ngoại lệ trong Python

Nội dung

1. Ngoại lệ là gì?
2. Tạo ra ngoại lệ như thế nào?
3. Tại sao phải xử lý ngoại lệ?
4. Cách xử lý ngoại lệ
5. Ngoại lệ trong Odoo

Ngoại lệ (exception) là gì?

"An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions." (Java documentation)

Ví dụ về ngoại lệ

```
>>> print(1/0)
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
ZeroDivisionError: division by zero
```

Danh sách ngoại lệ trong Python

BaseException

- +-- Exception

 - +-- ArithmeticError

 - | +-- ZeroDivisionError

 - +-- TypeError

 - +-- ValueError

 - ...

<https://docs.python.org/3/library/exceptions.html#exception-hierarchy>

Tạo ngoại lệ

```
# raise a built-in exception type  
raise Exception('error')
```

```
# create a new exception type  
class CustomException(Exception):  
    pass  
  
raise CustomException('error')
```

Best practice: tạo ngoại lệ với thông điệp rõ ràng

```
if self.company_id.sinvoice_start and self.date_invoice and self.date_invoice < self.company_id.sinvoice_start:
    raise UserError(_("You cannot issue S-Invoice for the invoices %s that has invoice date earlier than the company's"
        " S-Invoice Start Date (%s). If you still want to do it, you could either:\n"
        "* Change the date of the invoice to a date that is later than or equal to %s;\n"
        "* Or, go to Accounting > Configuration > Settings and change S-Invoice Start Date to"
        " a date that is earlier than or equal to %s.")
        % (
            self.number,
            format_date(self.env, self.company_id.sinvoice_start),
            format_date(self.env, self.company_id.sinvoice_start),
            format_date(self.env, self.date_invoice)
        )
    )
```

Tại sao phải xử lý (handle) ngoại lệ?

- Bỏ qua ngoại lệ để chương trình tiếp tục hoạt động
- Log hoặc bắn ra ngoại lệ khác với thông điệp rõ ràng, phù hợp hơn
- Thực hiện các dọn dẹp cần thiết khi ngoại lệ xảy ra
- Thử lại lần nữa (retry)

Xử lý ngoại lệ

try:

do_something()

except:

do_something_if_exception_occurred()

else:

do_something_if_exception_not_occurred()

finally:

always_do_something()

Bắt một kiểu ngoại lệ

example 1

try:

print(1 / 0)

except ZeroDivisionError:

print('zero division error')

example 2

try:

print(1 + "0")

except ZeroDivisionError:

print('zero division error')

Bắt nhiều kiểu ngoại lệ

example 1

```
try:  
    do_something()  
except (ZeroDivisionError, TypeError):  
    print('zero division or type error')
```

example 2

```
try:  
    do_something()  
except ZeroDivisionError:  
    print('zero division error')  
except TypeError:  
    print('type error')
```

Thứ tự bắt ngoại lệ

```
# example 1: catch the most specific
# exception first
try:
    connect()
except socket.timeout:
    retry()
except Exception:
    print('error')
```

```
# example 2: catch the least specific
# exception first
try:
    connect()
except Exception:
    retry()
except socket.timeout:
    print('timeout')
```

Xử lý ngoại lệ với try...finally

example 1

acquire a database connection

cr = self.pool.cursor()

cr.execute(...)

release the connection

cr.close()

example 2

cr = self.pool.cursor()

try:

cr.execute(...)

finally:

cr.close()

Lấy thông tin ngoại lệ

```
# normal exception message
```

```
try:  
    do_something()  
except Exception as e:  
    print(e)
```

```
# detailed exception with traceback
```

```
import traceback  
try:  
    do_something()  
except:  
    print(traceback.format_exc())
```

Bỏ qua mọi ngoại lệ

example 1: bad practice

try:

do_something()

except:

pass

example 2: better

try:

do_something()

except Exception as e:

logger.error(e)

Ngoại lệ trong Odoo (odoo.exceptions.*)

- **UserError**

Example: when the user tries to do something that has no sense given the current state of a record.

- **ValidationError**

Example: violation of python constraints.

- **AccessError**

Example: when you try to read a record that you are not allowed to.

Ngoại lệ trong Odoo (odoo.exceptions.*)

- **AccessDenied**

Example: when you try to login with a wrong password.

- **MissingError**

Example: when you try to write on a deleted record.

Thanks!