

# Throw Apex built-in exceptions

Did you know that you can throw Apex's built-in exceptions?

There's no need to create your own exceptions.

I suppose you've used **AuraHandledException** many times, but you can throw also other exceptions.

```
try {  
    CalloutException exception = new CalloutException();  
    exception.setMessage('My built-in callout exception!');  
    throw exception;  
} catch (Exception e) {  
    System.debug(e.getMessage());  
}
```

Here is a list of standard exceptions you can throw:



- AsyncException
- CalloutException
- DmlException
- EmailException
- ExternalObjectException
- InvalidParameterValueException
- LimitException (though it still can't be caught)
- JSONException
- ListException
- MathException
- NoAccessException
- NoDataFoundException
- NoSuchElementException
- NullPointerException
- QueryException
- RequiredFeatureMissingException
- SearchException
- SecurityException
- SerializationException
- SObjectException
- StringException
- TypeException
- VisualforceException



# Should you throw built-in exceptions instead of creating your own?

As we read in the documentation:

*An exception denotes an error that disrupts the normal flow of code execution. **You can use Apex built-in exceptions or create custom exceptions.** All exceptions have common methods.*

The perfect use case for built-in exceptions can be logic that wraps standard functionality:

- Unit of Work with DmlException
- Selector Layer with QueryException
- Callout Service with CalloutException

```
public SObject toObject() {
    List<SObject> records = toList();

    if (records.size() > 1) {
        QueryException e = new QueryException();
        e.setMessage('List has more than 1 row for assignment to SObject');
        throw e;
    }

    if (records.size() == 0) {
        return null; // handle: List has no rows for assignment to SObject
    }

    return records[0];
}
```