

Edit (<https://github.com/foundeo/cfdocs/tree/master/guides/en/error-handling.md>)

Error handling

CAUSE AN ERROR

Let's expect a Form submitted by a user. You can have validated it on client side but there could be hacky input notwithstanding. The validation state can be even more unclear if development was separated into front end and back end developers who just talk about interfaces needed.

```
if (not isNumeric(Form.age)) {  
    abort "Your value for ""age"" have to be numeric!";  
}
```

TRY / CATCH / FINALLY

If you write error-prone code or just expect something you can't or do not want to handle manually, you can wrap (/wrap) it into a try-catch-block like this:

```
try {  
    throw(message="Dooooooooomsday!", detail="You destroyed everything!");  
}  
catch (any e) {  
    writeOutput("Error: " & e.message);  
    rethrow;  
}
```

HTTP REQUEST RESPONSE

In age of REST-API usage we often get specific requests for specific answers. Then we have to validate incoming params conscientious. The response is essential for the requester or the requesting script to continue to work.

```
errors = [];  
if (not isNumeric(Form.age)) {  
    ArrayAppend(errors,'Your value for "age" have to be numeric!');  
}  
if (Len(Form.name) lt 8) {  
    ArrayAppend(errors,'Your value for "name" is to short. Type 8 letters at least!');  
}  
  
if (ArrayLen(errors)) {  
    statuscode = 400;  
    message = errors;  
}  
else {  
    statuscode = 300;  
    message = 'OK';  
}  
  
cfheader(statuscode = statuscode);  
writeOutput(serializableJSON({'status': statusCode, 'msg': message}));
```