How to write an error message

# Introduction

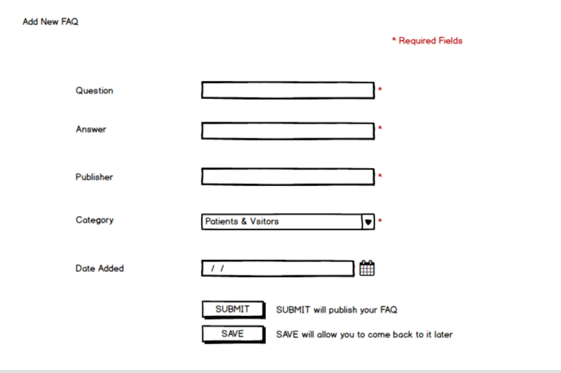
It is a common mistake among developers of anything to test their systems to only see that they work. Once it works – then they believe they are done. However, users are not so easy to predict. They will always do something unanticipated that will cause the system to stop working. When that happens, the system has to tell the user that an error has occurred and then offer a suggestion on how to fix it the error and encourage the user to try again. This is a key element in the user experience – recovering from errors. We also have to remember that users, and we developers do this too, often blame themselves for the errors and may become afraid to continue with the interaction. In that case the entire business case for the system we created is at risk since the user may abandon what they are doing and move on. It is worth the time and effort required to ensure that errors are easily fixed and the users encouraged to working on the interaction.

## Types of errors

We need to anticipate the types of errors possible in the systems we create. A lot of content management is about putting data into a database. We use html forms to do so. Those forms have extensive validation to ensure that the data is correct. When the validation fails then an error occurs, and the interaction stops. It is important then that we manage the error state and do two things: prevent the error and respond to the error appropriately.

We can prevent the error in two ways: one is to show that the content is required and the other is to show an example of the proper formatting for the data for that form field. The easiest means of showing that a form field is required is to use a red asterisk \* then on the same page and possibly nearby we would have the text, also known as a legend, \* = required. The red asterisk is a strong convention but not universal. Being redundant by having the legend and the red asterisks is appropriate. It also acknowledges that not everyone can see a color and this is a clear explanation for the asterisk.

Here is an example of the red asterisk in use:



Below is an appropriate way of showing the error in the form is to change the field border colour to red and then display the error text also in red.

Image from: Faddis, Ray "The Blog of Ray Faddis"  http://www.rayfaddis.com/blog/2013/05/27/jquery-validation-with-twitter-bootstrap
Accessed December 28, 2019.

This is a good example of a caught error – but you can also see that the red asterisk has not been used – which is an inconsistency that should be avoided. You need to warn the user about what fields are required before they enter their information. You will also note the inconsistency in the error messages – one is general – “This field is required” and the second specific “Please enter a valid email address” – the second error message is the style you want to use in all error messages.

Now that we have marked the error and stopped the interaction – what do we write? There are three components to an error message:

1. Identify the error
2. Suggest the correction
3. Encourage the user to continue

Developers are usually good at the first two and not always good on the third. All three components need to be in place for a successful error message. A successful error message keeps the user engaged so that they try and fix the errors.

If we look at the example above the first error message would now be:

“Your Name is required – please enter your Name”

“Required” identifies the problem as this information was not submitted.

“Enter” is the suggestion and tells them what to do.

“Please” is the encouragement.

Automating your error messages is would also be good in the case. For each required form field your error message would be programmed with a variable that takes the field label and adds it to the error message in two places.

Your “field-name” is required – please enter your “field-name”

It is important that you write these messages out before you start programming. Which means that you need to know what the interaction is that you are building so that you can see what the error messages are and how they should be written. This is another purpose for the draft features assignment.

Another way for preventing errors is to show an example of the content before the user enters it. A typical example for this would be a telephone number. You could accept any ten numbers, convert it to a string, and then enter it as text rather than numeric in the database. While this is more work for you – it is less work for the user. Your validation would still have to check for numbers to avoid non-numeric characters being entered. Perhaps the easiest way is to show your telephone field with a properly formatted example for the user to follow.

<form action="/action\_page.php">

Telephone: <input type="tel" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}" required>

<input type="submit">

<span>Format: 123-45-678</span>

</form>

Similarly, you would use a date picker to choose a properly formatted date. Country and province lists are also better in order to choose locations.

## Success messages

Anytime a user interacts with the system or feature then there should be a response from the feature indicating progress: error, success, or status. Success messages are added to complete the interaction and tell the user what to expect next. They can be an overlay, or modal window, or a completely separate page. For example if a user submits information via the contact form once the form is submitted redirecting the user to a new page with the message: “Thank you for your submission. We will respond to your message within the next 72 hours.” In this way the user knows that the message has been sent and when they can expect to hear back. The interaction is finished and they can continue with their day. Sending a follow-up email is also common and is a further opportunity to remain in contact and even encourage them to do business with the company or organization.

The donation feature would be an interaction where we would do this as well. Once the donation is sent – then a similar message would be shown. You might make this page something the user could bookmark as a means of keeping track of their donation. You can also use the page to add a more refined and careful thank you that thanks them and encourages them further. A follow-up email, which may also act as a tax receipt, is so commonplace as to be mandatory.

In a similar fashion admin users also need to know whether they have been successful when they make submissions in the CMS. You have to be more careful in redirecting the admin user as you want to make their workflow easier – especially if they have a lot of content the work with. The simplest way is to return them to the page view listing with a message in green – “Your content has been saved to the database”. The message can be shown in a temporary heading area where they can see it. Moving them to a new page, like in the donation feature, would force them to go back to another page and over time require more work. If you don’t show them the new or edited page – then providing a link to the new page in the success message is another option to consider. You want to ensure that your interaction design and messaging make their work easier for them.

## Status messages/warnings

If there is some sort of delay in processing – for example someone has purchased something that will not arrive right away – then you will have to account for that delay in your message. Similar to the response to a contact form, the 72 hours to respond, you want to inform your users that they have started something and then give them a clear understanding of what happens next. If you order something from Amazon – you get all sorts of messages and emails showing you what is happening, when the product shipped, and when to expect your purchase.

An example would be a book an appointment tool for the hospital. Since booking the appointment would probably require the agreement of the doctor/service in question you message would be: “Thank you for your appointment request. It has been sent to the person in question and you will receive a confirmation within the next 72 hours”. There is no colour for this message (yellow is too hard to read on a computer screen) and so you want to make sure that you have managed the user’s expectations. If you set a time – make sure it is a sufficiently long enough time period to always meet that. People will wait for as long as you tell them and become annoyed if you make them wait longer.

Warnings as indications that the interaction has stopped until the user takes some action. The most common example is the warning that occurs when deleting a record in the CMS. Generally, the message will be: “Are you sure you want to delete this record?” Then with options to continue or cancel. There are other warnings depending on your validation and checking to make sure the content is correct.

## Finally

This is an introduction to writing error messages and other kinds of messages. These messages are a key part in the user experience of the systems we design and implement. They take time to do – but it is time well spent. Recollect you own experience with a system that does not take care of the user – would you do business with them?

# Error messages assignment

Last week your built the dataflow diagrams for an interaction your mapped out. Today you will take that interaction and add the appropriate messages to your document.

Feel free to trigger and write down as many messages you can find with the feature you worked with. If there are messages that should be there and are not – then add them to your documentation.

It is important to add these messages to your documentation in order that the developers can simply copy the messages and paste them into the code.

You will need error messages for:

Validation and data errors.

Empty form field errors.

Improper or incorrect login attempts.

You will need success messages for:

Successfully submitted forms – both public and admin.

Successfully deleted records.

Wherever a user has made a change, and it has gone through.

You will need status messages for:

The prompt to use on deleting records.

Whenever the user has to wait.

There may be more depending on the features you are building – these will give you a good start. Some messages may not show up until you after you have logged in – so feel free to use a service that you would normally use, like Gmail, and generate and copy the error messages from that application. When you add them to this document just tell me where you got them.

Add your messages to the document you sent to me and send it again. If I have suggested changes to the document – please make those changes now. It is due by the end of the week.