

< Back to blog

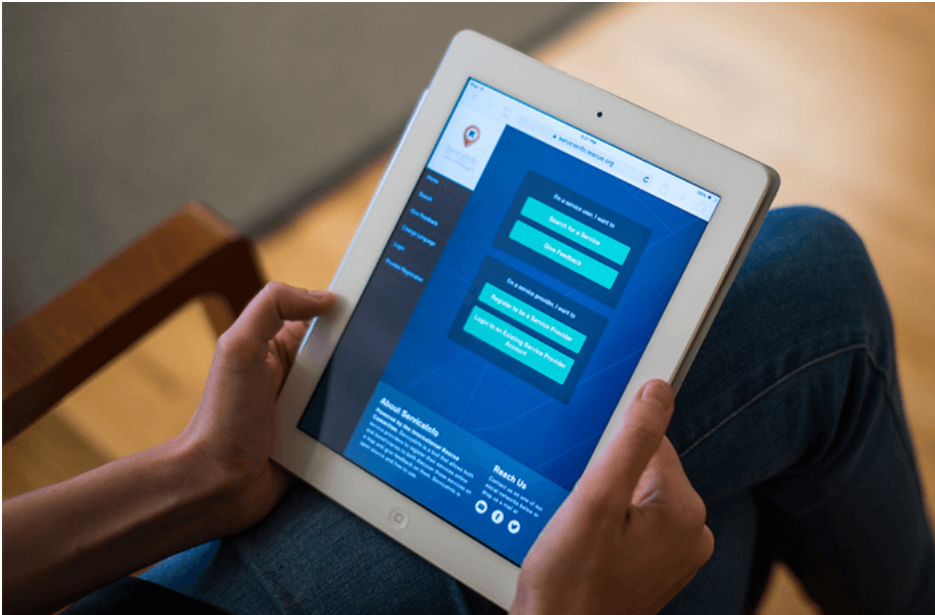
Testing & QA (/blog/categories/testing-qa/)

3 Common Form Testing Issues (Plus 1 Helpful Tool)

testing (/blog/tags/testing/), best practice (/blog/tags/best-practice/), forms (/blog/tags/forms/)



🐦 ([https://twitter.com/intent/tweet?text=3%20Common%20Form%20Testing%20Issues%20%28Plus%201%20Helpful%20Tool%29&tw_p=t28%2C%20common-form-testing-issues-plus-1-helpful-tool/](https://twitter.com/intent/tweet?text=3%20Common%20Form%20Testing%20Issues%20%28Plus%201%20Helpful%20Tool%29&tw_p=tweet=3%20Common%20Form%20Testing%20Issues%20%28Plus%201%20Helpful%20Tool%29&tw_p=t28%2C%20common-form-testing-issues-plus-1-helpful-tool/)) [in](https://www.linkedin.com/cws/share?url=http://www.cactusgroup.com/blog/2018/05/28/3-common-form-testing-issues-plus-1-helpful-tool/) (<https://www.linkedin.com/cws/share?url=http://www.cactusgroup.com/blog/2018/05/28/3-common-form-testing-issues-plus-1-helpful-tool/>) [f](https://www.facebook.com/sharer/sharer.php?u=http://www.cactusgroup.com/blog/2018/05/28/3-common-form-testing-issues-plus-1-helpful-tool/) (<https://www.facebook.com/sharer/sharer.php?u=http://www.cactusgroup.com/blog/2018/05/28/3-common-form-testing-issues-plus-1-helpful-tool/>)



Forms are something that I find myself testing frequently, whether it's an e-commerce checkout page or a new model (<https://docs.djangoproject.com/en/2.0/topics/db/models/>) in the Django admin. The challenge of forms is that users will often enter things that may not have been accounted for when the form was created.

For example, they might enter 罗比 for their name when you were only expecting users to enter names with letters of the Roman alphabet. If the unexpected characters cause errors, the user may not be able to continue using your site.

Subscribe to our newsletter

youremail@email.com

Issues with user-generated inputs can be discovered before they ever happen to a user by testing with better data.

Some people will use test data similar to Mr. Tester 123, Fake Street, City, NC 12345, test@example.com. This kind of fake data can be useful if you need to remember what you used or be able to pick it out of a list of real data. The problem with using this, or other similar data, is that it is unlikely to uncover any bugs in the system because it does not match real-world uses of the system.

The type of data you use will depend on the environment in which you are testing. You may be able to use real addresses if you are in a testing environment that is isolated from other systems. However, if the environment is not sufficiently isolated you may not want to use something like Mr. President, 1600 Pennsylvania Ave, NW Washington, DC 20006 (an address that I have seen used for testing). When using real addresses you risk real-world events. You wouldn't want to have to answer questions from law enforcement if a strange package is shipped to the President by accident.

One solution to ensure nothing is sent from a test environment is to use data that is easily identifiable as fake. When doing this, you need to be sure to still use data that is comparable to what a user may actually enter so that you do not miss important bugs. You also do not want to use John Doe, 123 Fake St, City, NC 12345 for every test. A few issues with using this or a similar address for *all* your tests jump out right away.

Zip code

Zip codes are one of the first problematic fields that come to mind when I think of test data. This is a field that people tend to make different assumptions about based on where they are from.

People from the US often assume all zip codes are only 5 digits (the standard in the US), and that they are always called "zip codes". What users actually enter will vary widely based on where the users of the site are located. Some potential valid zip/postcode combinations include:

- Numbers only
- Numbers starting with a 0
- 3 digits
- 10 digits
- Single character
- A mix of letters and numbers
- Spaces in the zip code

Subscribe to our newsletter 

youremail@email.com



Of course, some countries do not use zip codes at all. If the zip code field is limited to only accept five digits, a large portion of users may run into issues.

Names

The next piece of data in John Doe, 123 Fake St, City, NC 12345 that jumps out as problematic is the first and last name. An important thing to keep in mind with names is that they can vary in length, characters used, and the order in which first names and family names are used. They can also include characters from other alphabets. What happens if a user enters the name Drop Table or Null? Will Christopher Null (<https://www.wired.com/2015/11/null/>) never be able to submit the form because the site thinks the field is blank, or will a database table be dropped because of Drop Table (https://www.w3schools.com/sql/sql_drop_table.asp)?

Unexpected inputs

The Hamlet test (<http://qahatesyou.com/wordpress/2007/07/unleashing-the-hamlet/>) is one of my coworker's favorite tests for unexpected inputs in text fields. It consists of entering the entirety of Hamlet into a text field, all 135,013 characters, with the spaces removed. This is a good way of rooting out bugs related to text length and formatting. While you may not need to allow the user to enter so much text without spaces, you do need to be sure that any resulting errors are handled in a user-friendly manner.

While test cases may not always call for entering unexpected data, including them in testing can uncover or answer additional, previously unconsidered questions. For example:

- What languages should users be able to enter? Does this site have encoding for non-Latin alphabets or other languages?
- If a user enters characters from a language the site does not support, how is it handled? If a user enters Arabic, that is written from right to left, will that cause some unexpected behavior?
- What will happen when a user enters an emoticon in a text field?

There are many more to consider, but these are a few I have encountered.

Keeping track of test data

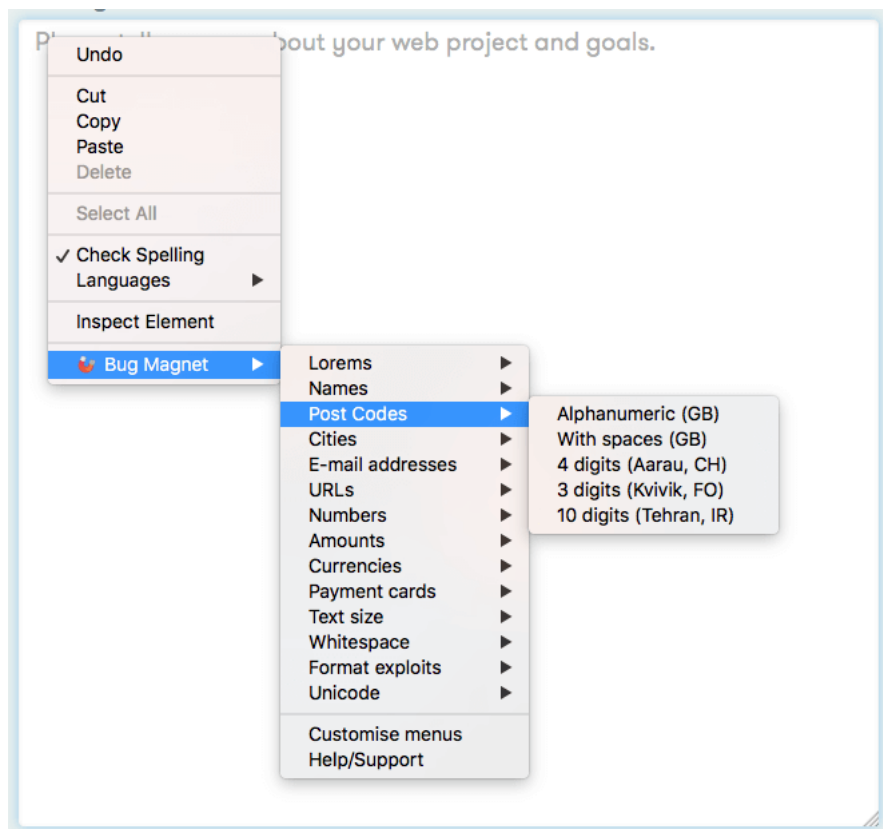
People often fall into the habit of using the same data for a field each time they test. To make keeping up with test data easier I have used text documents, spreadsheets, and sticky notes in the past. There are other tools to make this much easier.

Subscribe to our newsletter 



I recently came across Bug Magnet (<https://bugmagnet.org/>), a helpful tool for testing user-generated inputs. Bug Magnet is a plugin for Chrome and Firefox that organizes valid and invalid data in an easy to use fashion.

Now, instead of referencing my list of possible zip codes from a doc or sticky note, I use this plugin. Right-clicking on the field that you are testing opens a submenu with many options which can be problematic if not properly addressed.



Bug Magnet includes options for a variety of fields. If you do not see something you need, one of the other great features is the ability to add your own configurations. I recently added date formats to mine using the directions found here (<https://bugmagnet.org/customising.html>).

Ready to test?

When using test data, make sure to use what is most likely to uncover bugs, not just what is easy and memorable. Also, make sure that the data you use fits the testing you are doing (you do not want to make news for accidentally shipping something to the White House).

Keep in mind users will enter unexpected things if given the opportunity. Testing fields with a wide variety of data, and making it easier for yourself by installing tools like Bug Magnet, can go a long way toward improving user experience if they enter an input that your system doesn't support.

Subscribe to our newsletter

youremail@email.com



[Editor's Note: Neither the author nor Cactus have any connection with the Bug Magnet developers, other than as users. No payment or other compensation was received for this review. This post reflects the personal opinion and experience of the author and should not be considered an endorsement by Cactus Group.]



Show Comments

Posts:

Technical (/blog/2019/02/15/old-projects/) Feb. 15, 2019 (/blog/2019/02/15/old-projects/)

News & Events (/blog/2019/01/26/7-conferences/) Jan. 26, 2019
(/blog/2019/01/26/7-conferences/)

x

My New Year's Resolution: Work Less to Code Better

[{/blog/2018/12/27/new-years-resolution-work-less-code-better/}](/blog/2018/12/27/new-years-resolution-work-less-code-better/)

Technical [{/blog/2018/12/27/new-years-resolution-work-less-code-better/}](/blog/2018/12/27/new-years-resolution-work-less-code-better/) Dec. 27,

2018 [{/blog/2018/12/27/new-years-resolution-work-less-code-better/}](/blog/2018/12/27/new-years-resolution-work-less-code-better/)

Resources:

Teach Your Sites to Call for Help: Automated Problem Reporting for Online Services [{/talks/automated-problem-reporting/}](/talks/automated-problem-reporting/)

Talk [{/talks/automated-problem-reporting/}](/talks/automated-problem-reporting/)

Teach Your Sites to Call for Help: Automated Problem Reporting for Online Services [{/talks/automated-problem-reporting/}](/talks/automated-problem-reporting/)

Talk [{/talks/automated-problem-reporting/}](/talks/automated-problem-reporting/)

Django 1.1 Testing and Debugging [{/talks/#books-section}](/talks/#books-section)

Book [{/talks/#books-section}](/talks/#books-section)

Contact us

solutions@caktusgroup.com <mailto:solutions@caktusgroup.com>

T 919-951-0052

F 919-928-5516

108 Morris St, Suite 2
Durham, NC 27701

Quick links

[Services \[{/services/}\]\(/services/\)](/services/)

[Our work \[{/casestudies/}\]\(/casestudies/\)](/casestudies/)

[About us \[{/about/}\]\(/about/\)](/about/)

[Blogs <https://learn.caktusgroup.com/caktus-blogs>](https://learn.caktusgroup.com/caktus-blogs)

[Careers \[{/careers/}\]\(/careers/\)](/careers/)

[Events \[{/events/}\]\(/events/\)](/events/)

[Talks \[{/talks/}\]\(/talks/\)](/talks/)

[Ebooks \[{/ebooks/}\]\(/ebooks/\)](/ebooks/)

[Press \[{/press/}\]\(/press/\)](/press/)

[Contact \[{/contact/}\]\(/contact/\)](/contact/)

Newsletter

youremail@email.com

Subscribe to our newsletter 

youremail@email.com



Get tips, see case studies, and stay up to date on Cactus news.

Learn more. (<https://learn.cactusgroup.com/newsletter>)

Privacy Policy (<https://www.iubenda.com/privacy-policy/91739120>)

Cookie Policy (<https://www.iubenda.com/privacy-policy/91739120/cookie-policy>)

Follow us



(<https://twitter.com/cactusgroup>)



(<https://github.com/cactus>)



(<https://www.facebook.com/CactusGroup>)



(<https://www.linkedin.com/company/cactus-consulting-group-llc>)



Cactus Group
@CactusGroup



We're [#hiring](#) for a contract [#WebDeveloper](#) based in the US with [#Python](#), [#Django](#), [#SQL](#) database & [#Windows](#) Python Packaging and deployment experience: cakt.us/30gXPgu

Spread the word [@ThePSF](#) [@PythonWeekly](#) [@pydatageek](#) [@remotepython](#) [@djangogigs](#)
[@djangoproject](#) [@djangojobsnet](#)



WE'RE HIRING!

CONTRACT WEB DEVELOPER

Come Grow With Us

Subscribe to our newsletter 

