uMunge introduction

Introducing data munging

Using live data for software development increases the risk of data being lost or stolen. New data protection laws (GDPR) apply from May 2018 to anyone who handles personal data, with hefty fines for those who are careless or misuse data.

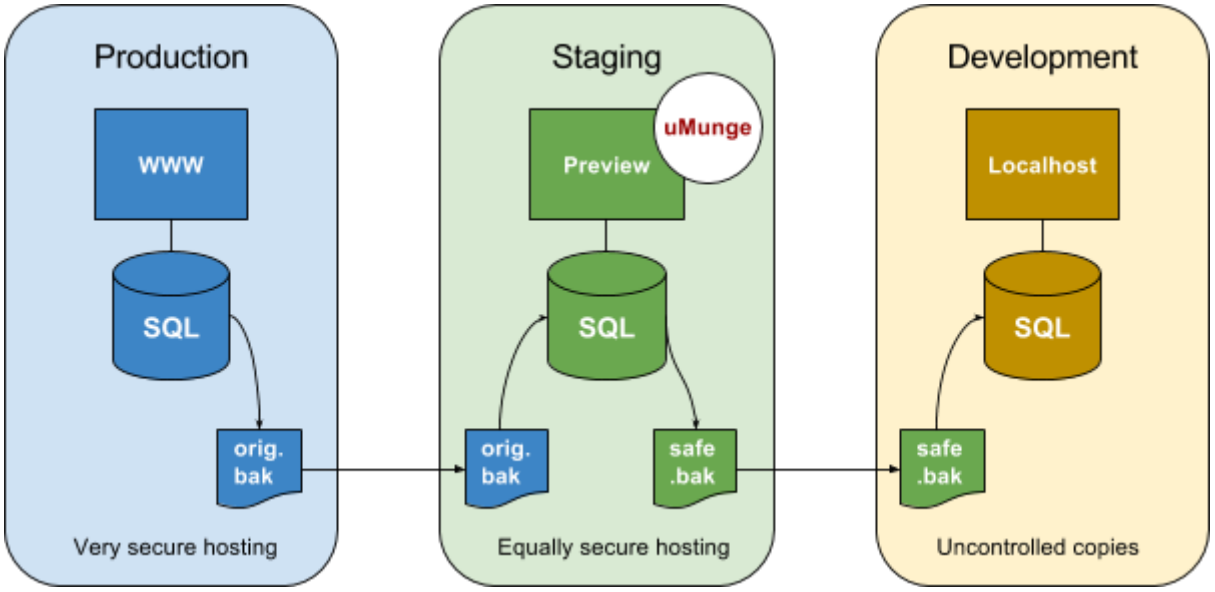
Reduce that risk by anonymising or munging your database before use in development or staging environments. Munged data retains the structure, size and readability of the original source, but removes the liability.

Where do I use it?

uMunge is an Umbraco package that is designed for use with production-scale websites. It should play nicely on Umbraco 7.4+ sites you have already built and launched.

We built it to suit the workflow of our agency, keeping the managers happy without getting in the way of developers.

Recommended architecture



Ideally, there should be at least 3 environments for your project:

1. Production, with live data, powerful servers and very secure and controlled access to data.

2. A staging environment (also known as QA, UAT, Preview or Integration) with equally good servers and security, where customers are often able to see your shiny code for the first time. Live data would be safe here, but your customers may feel happier if they see munged content.
3. Development, where things get broken in order to get fixed. Where you are free to create Umbraco magic. Where you should not have the liability of personal information.

Your code is created in development, and deployed up to staging and then production - by using Courier, uSync or your favourite source control and build tools.

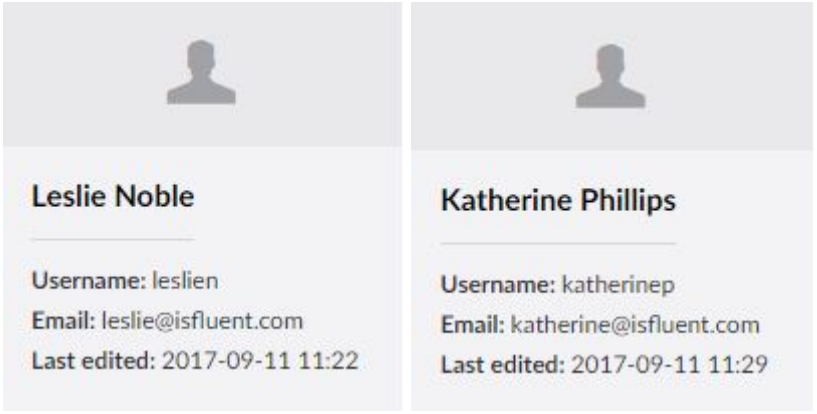
Every so often, you need a fresh copy of the latest live data in your staging or development environments. A database backup is copied to the staging server and restored. Munging takes place in the safety of the staging environment, removing sensitive personal information. A backup of this safe database is then created on the staging server and copied to development machines.

The munging configuration is stored in a file, not in the database, so it can be kept for when an up-to-date copy is needed again.

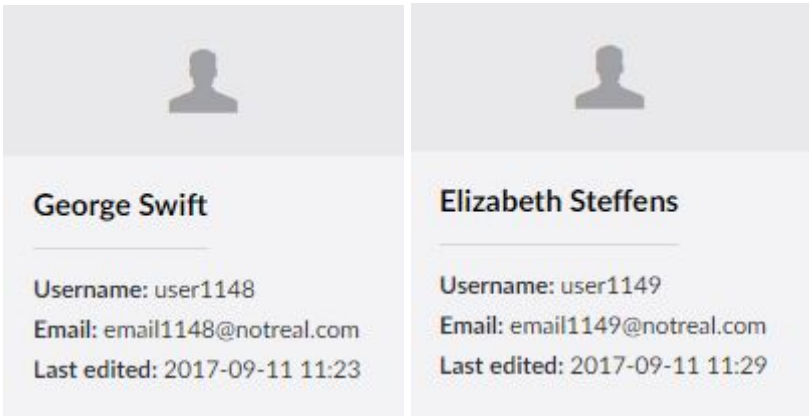
What does uMunge do?

uMunge modifies personal information stored in Content, Media, User and Member records. Node names and properties can be anonymised, choosing from a selection of pre-defined “mungers” to match the type of data.

For example, you might have a couple of Member records:



A Member’s name, username and email address all contain personal information that you’d like to remove from development copies of the database. After munging, these Member records might look like this:



Good for testers

In the example above, usernames and email addresses are simply generated from Umbraco's member ID. It is important to remove the personally-identifiable information from these fields, but it doesn't really matter what they become, so long as they are distinct from each other.

Member names, however, are more likely to be noticed during testing of back office or public sites. We could have changed Leslie's name to Member1148, but it's easier to work with a development site when people's names *look* realistic. uMunge includes a "Real Name" munger that picks from a list of forenames and surnames.

Moreover, we have found that it helps UI testing if there is a similar distribution of long and short names between the live and development sites. uMunge's Real Name munger achieves this by picking names that preserve length, as shown in the two examples.

Repeatable, not just random

Having got used to testing with George Swift, it would be annoying if munging a newer backup of live data suddenly turned Leslie Noble into Thomas Kirby instead. uMunge uses hashes of the old data rather than using random numbers, which ensures that Leslie Noble on Member ID 1148 *always* becomes George Swift. Munging can therefore happen as often as you need fresh data from live, without upsetting anyone.

Deleting history

Content nodes can be munged in similar ways to the example Member records above. Umbraco keeps a history of previous content versions, so personal information would still be there in the database if we didn't do more.

uMunge deletes all old content versions so that only the munged one remains, and no-one can use Rollback to get at the sensitive data. It can also delete the audit trail (database log) in case anything sensitive is stored in there.

For a mature database these deletions can make the backup size significantly smaller too, which is a bonus for copying it between development machines!

Logging

uMunge creates tables in the database to log munging actions. The munging history can be viewed on any machine that takes the munged backup, to verify both that it was munged, and also which fields were munged. This may be useful in the event of an audit.

Actions are also logged as text to the Umbraco trace log. In both cases, great care is taken not to log any of the original personal information!

Small print

uMunge is in beta (version 1.0). We would like to improve it in all directions, based on feedback from the community.

It is limited to smaller databases (500 nodes) and each version will expire three months after it was built - please contact us via the forums or support@datamunge.io if you need these restrictions lifting.