Code Signing for the Web

Creating Digital Signatures for Backdrop CMS and Silkscreen CMS Modules

John Franklin
Sentai Digital
Silkscreen CMS Maintainer

In the next 50 minutes...

What are Backdrop & Silkscreen?

Encryption, PKI, and digital signatures 101

Why sign modules?

Code Sign API

Project module integration

Future development

Q&A

Backdrop & Silkscreen: History

Backdrop forked off Drupal 7

- Includes CMI
- Does not include Symfony or Twig
- Only supports MySQL
- Only stores config in JSON files
- Add new features: Layouts & Installer
- Most Drupal 7 modules port with only a few lines
- Targets smaller sites – small businesses, non-profits
- Intended to be easy to setup, easy to build, easy to use.

Backdrop & Silkscreen: History

Silkscreen forked off Backdrop around 1.8

- Drop-in replacement: tracks releases in Backdrop
- Includes “driver” modules for
  - Databases (PostgreSQL and SQLite)
  - Config storage (in-database, in-memory, in-session)
  - Caching (in development)
- Contributes as much as possible upstream
- Targets sites that need a little more

Encryption 101

Symmetric (AES, DES, RC4)

One key, encrypts and decrypts

Asymmetric (RSA, DSA, ElGamal)

Two keys, one public and one private

Data encrypted by one, decrypted by the other

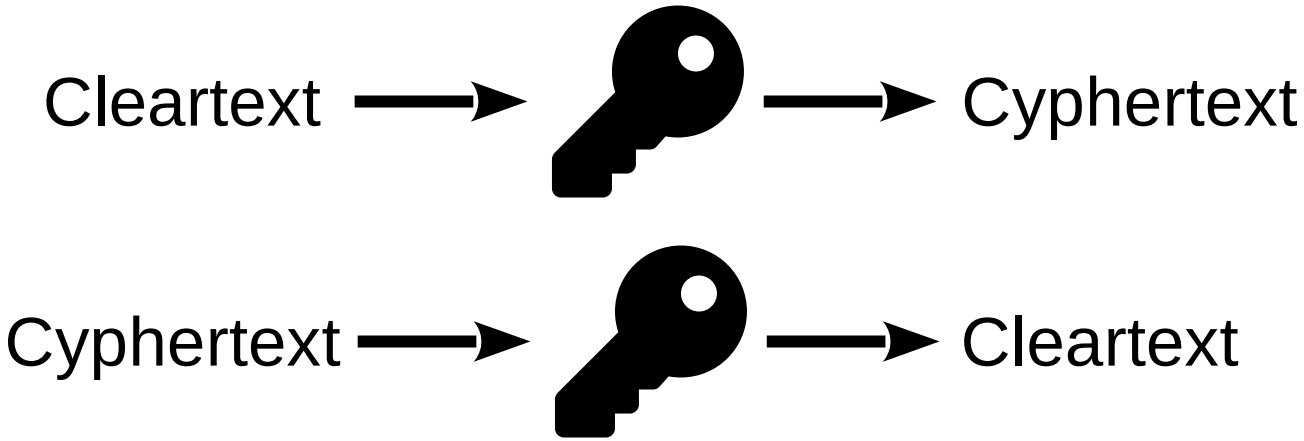
Hashing (MD5, SHA256)

Used for signatures

Encryption 101

Symmetric

One key, encrypts and decrypts

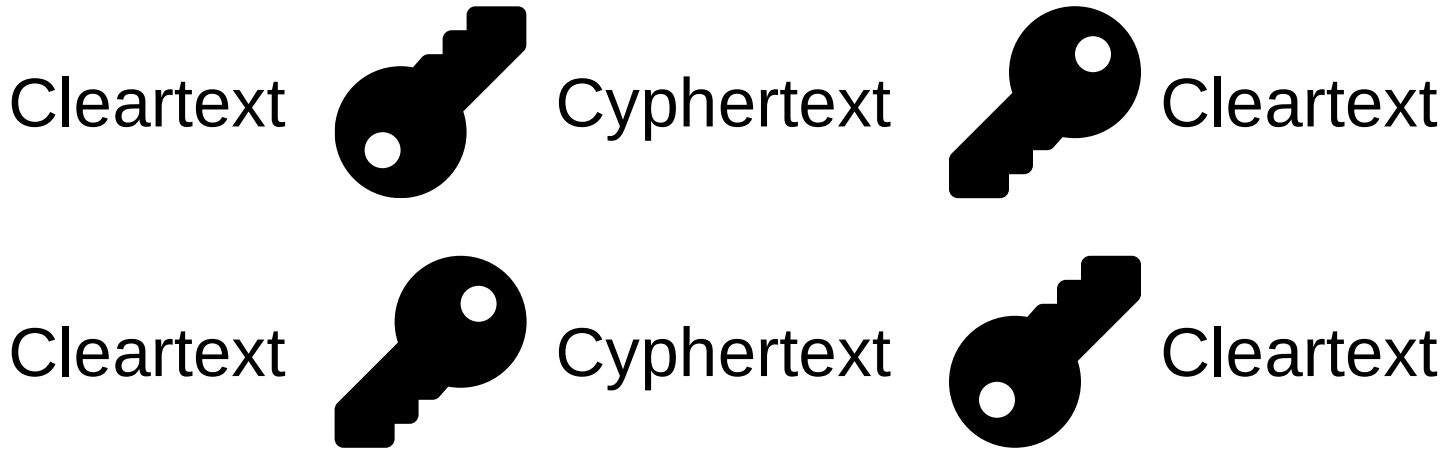


Encryption 101

Asymmetric

Two keys: public key and private key

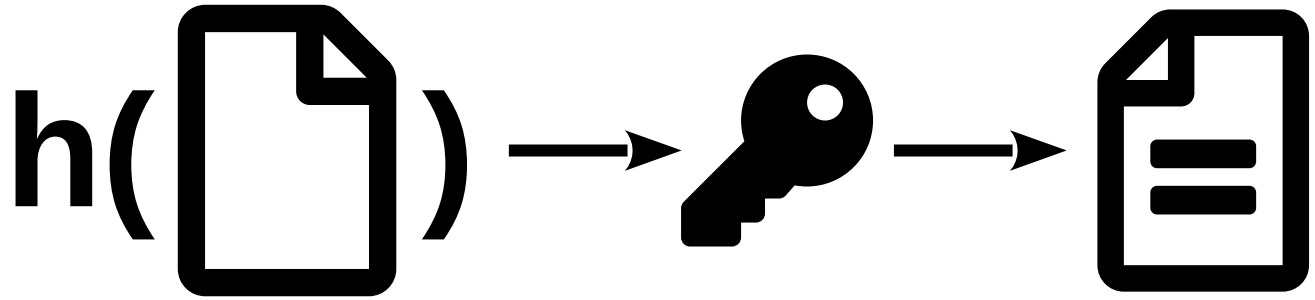
Encrypted with one can only be decrypted with the other



Encryption 101: Create a digital signature

Hash a message

Encrypt the hash with your *private* key

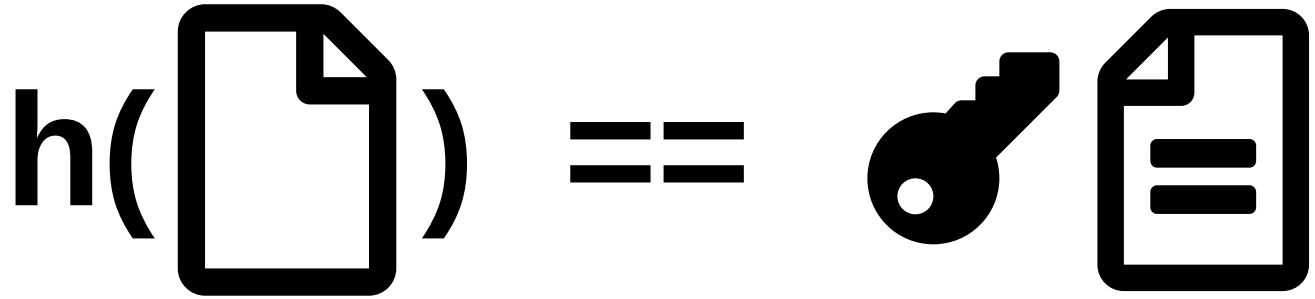


Encryption 101: Validate a digital signature

Hash the message

Receiver decrypts signature with *public* key

Hashes must match!



Encryption 101: Digital Signatures

A signature is an encrypted hash that *validates* the message.

Signatures are tied to a key pair.

Key pairs are owned by a person or entity.

Signatures tell *who* sent it.

Key management is *identity management!*

Encryption 101: Key Management

Monolithic

Follow the chain of signatures until you find a trusted “root”.

Example: TLS

Web of Trust

I trust people I’ve signed

I trust people they’ve signed (a little less)

Example: PGP / GnuPG

TOFU

Trust On First Use

Example: SSH host keys

Why sign modules?

Verify the module's integrity

Hashes of the tarball ensure it is valid before it is even unpacked.

Provide the identity of the developer

Who wrote this? Do we trust them?

Provide the identity of the module itself

A module's cert can be revoked without revoking the developer's cert

Signature Chain

Root cert *Silkscreen Module Signing CA*

- signs *Module Signing Intermediate CA*

- signs Silkscreen Module Signing Service

- signs the core modules

- signs contrib modules

- signs *John Franklin, Developer*

- signs *config_session*

- sign sandbox modules

- signs *Sentai Digital*

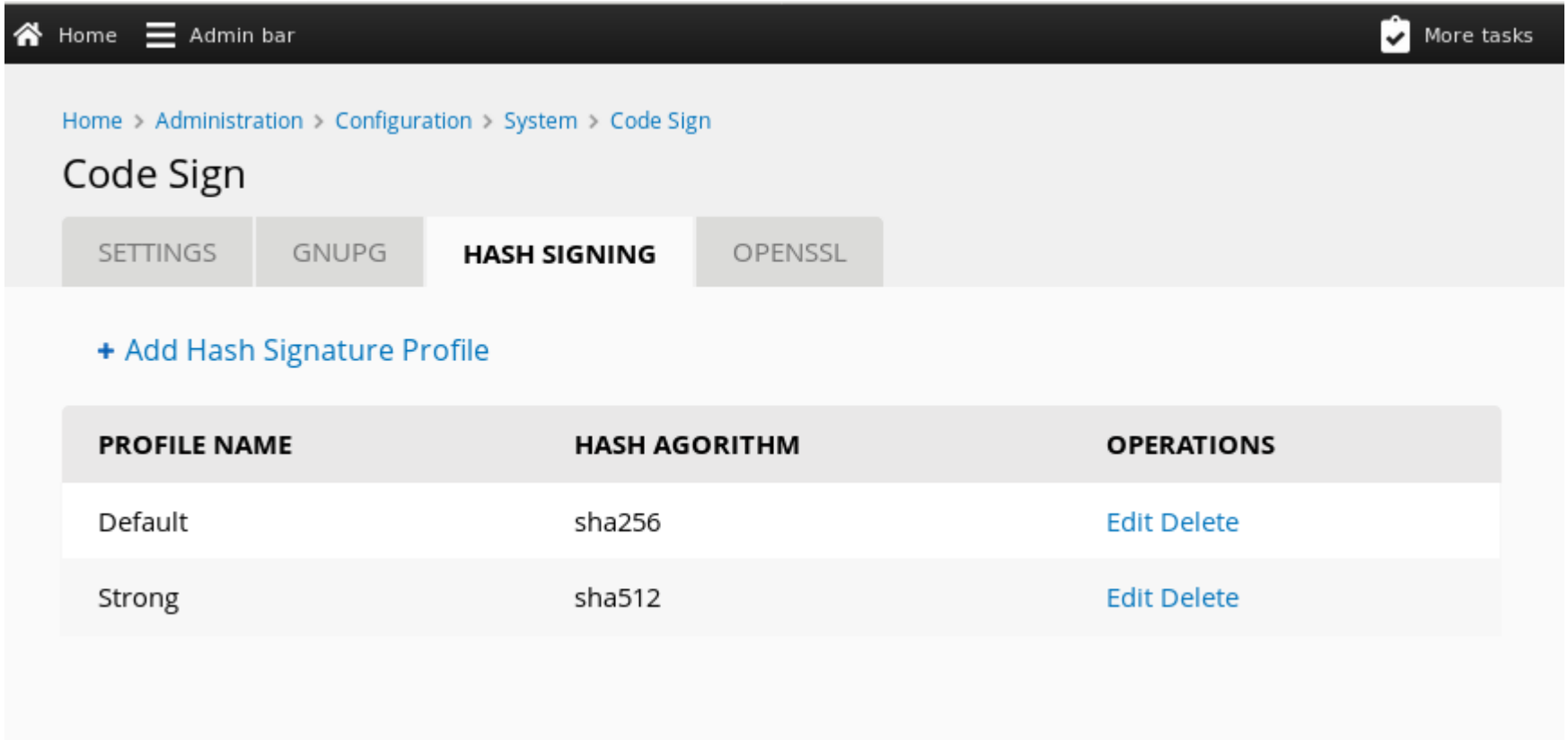
- signs custom modules for a site

Code Sign API

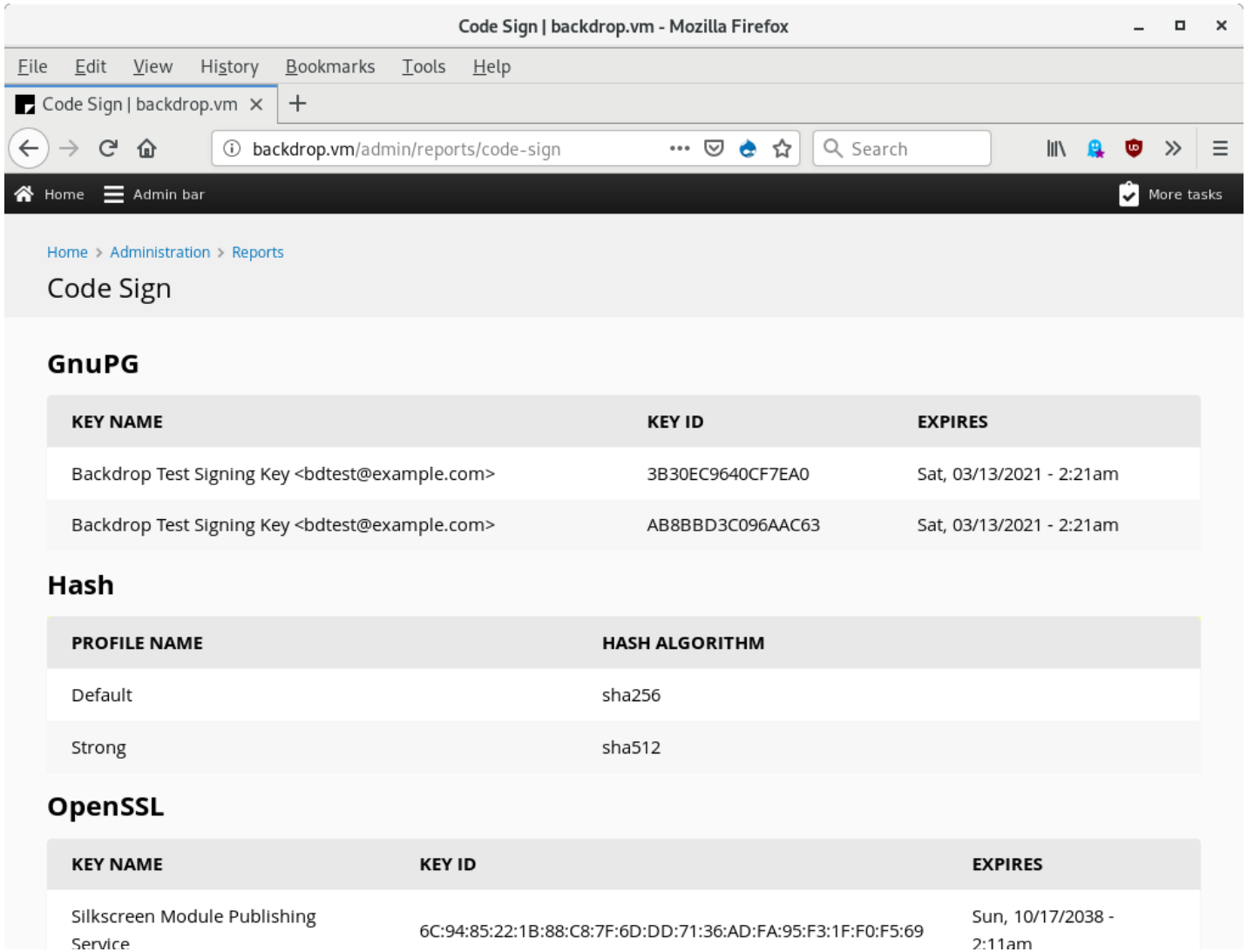
Code Sign module in core

- API that signs and verifies data
- Manages signing profiles
- Contrib modules handle specifics
  - In-core: basic hashing (not really signing, just for tests)
  - OpenSSL (code_sign_openssl)
  - GnuPG (code_sign_gnupg)
  - Sodium (future module)

Code Sign API



Code Sign API



Code Sign API

```
function code_sign_sign($signer, $profile, $data, $options = array());
function code_sign_verify($signer, $data, $signature = NULL, $options);
function code_sign_get_signer_status($signer);
function code_sign_get_private_keyring($signer);
function code_sign_get_public_keyring($signer);
```

Code Sign API

```
function code_sign_sign($signer, $profile, $data, $options = array());
function code_sign_verify($signer, $data, $signature = NULL, $options);
```

$signer – Signing engine (hash, gnupg, openssl)

$profile – Profile ID the $signer will understand

$data – The data to be signed / verified

$signature – The signature block

$options – Passed along to $signer, currently unused.

Note – Verify has no $profile! Must be in $signature!

Code Sign API

API for getting list of providers

```
function code_sign_get_signature_backends();
function code_sign_get_signing_profiles();
```

Theme defined:

code_sign_result – Used in Module list to show if the signature is valid.

Signing Engines may also add a link to more info. (e.g., cert chain)

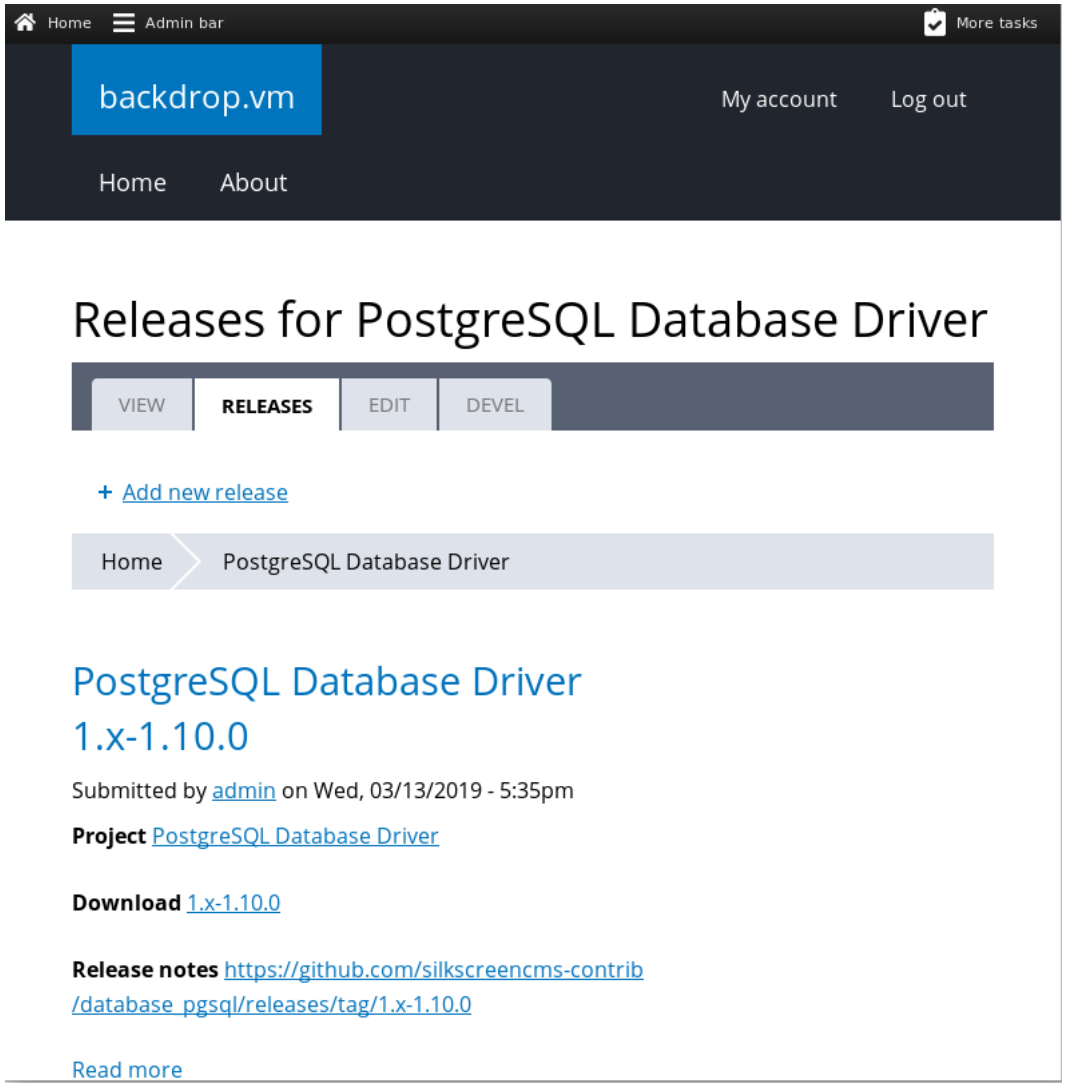
Code Sign API: Define an Engine

```php
/**
 * Implements hook_code_sign_info().
 */
function code_sign_openssl_code_sign_info() {
  $signers = array();

  $signers['openssl'] = array(
    'title' => t('OpenSSL'),
    'sign callback' => 'code_sign_openssl_sign',
    'verify callback' => 'code_sign_openssl_verify',
    'status callback' => 'code_sign_openssl_status',
    'public keychain callback' => 'code_sign_openssl_get_ca_list',
    'private keychain callback' => 'code_sign_openssl_get_signing_certs',
    'file' => backdrop_get_path('module', 'code_sign_openssl') . '/openssl.codesign.inc',
  );

  return $signers;
}
```

Project Module

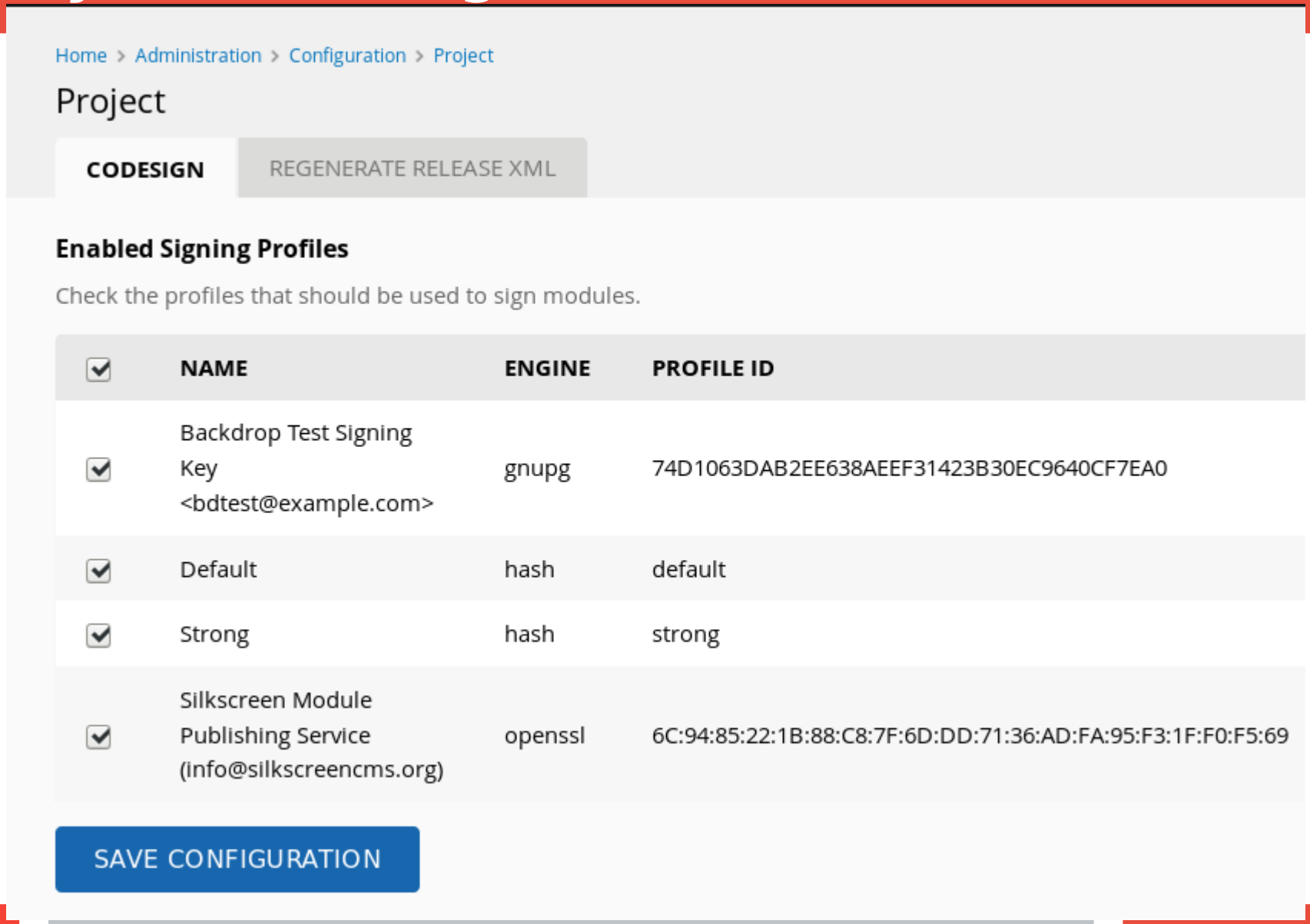


Project Code Sign module

Generate signatures on release

- Project Module calls “new release” hooks
- Project Code Sign implements hooks
- Project Code Sign calls Code Sign API with selected profile ID
- Code Sign calls engine callbacks passing in the profile ID
- Engine signs data with profile settings, returns signature
- Project Code Sign adds signature(s) for tarball to XML catalog

Project Code Sign



Project Code Sign: XML Sample

```
<signatures>
  <signature>
    <crypto_engine>gnupg</crypto_engine>
    <profile_id>74D1063DAB2EE638AEEF31423B30EC9640CF7EA0</profile_id>
    <signature_block>-----BEGIN PGP SIGNATURE-----
-----END PGP SIGNATURE-----
    </signature_block>
  </signature>
  <signature>
    <crypto_engine>hash</crypto_engine>
    <profile_id>default</profile_id>
<signature_block>sha256:84c282e7e1921c3bacd7618e83a539de296364aaa2d7b160908d995f62702
eb6</signature_block>
  </signature>
  <signature>
    <crypto_engine>openssl</crypto_engine>
    <profile_id>6C:94:85:22:1B:88:C8:7F:6D:DD:71:36:AD:FA:95:F3:1F:F0:F5:69</
profile_id>
    <signature_block>-----BEGIN CODESIGN-----
-----END CODESIGN-----
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
    </signature_block>
  </signature>
</signatures>
```

Project Installer

Home › Administration › Functionality

Install modules

LIST MODULES | **INSTALL NEW MODULES** | UPDATE MODULES | UNINSTALL MODULES

Search

SEARCH

Sort by: Relevance ▲ Most installed Title Latest release

Showing 1 to 20 of 375.

MODULES

Backup and Migrate

Add to Installation queue

This module makes the task of backing up your Backdrop database and migrating data from one Backdrop install to another easier. It provides a function to backup the entire database to file or download, and to restore from a previous backup. You can also schedule the backup opera... details

229 Installations

Webform

Remove from Installation queue

Webform is the module for making forms and surveys in Backdrop. After a submission customizable e-mails can

Installation queue

✕ Webform

INSTALL Clear queue

Manual installation

Future

More signing engines

- Sodium

Code signing core downloads

Module code signatures

- code_sign_sign(Manifest + hash of each file)
- Detect corrupted or hacked modules
- How to handle patches?
- Support revoking signatures

Code signing core

When will it be released?

Track now in 1.x-dev

https://github.com/backdrop/backdrop-issues/issues/1992

Backdrop 1.14 release

September 15th, 2019

Silkscreen 1.14 release

September 15th, 2019

Let's build some