



Newsletter

Summer 2020



Dear Readers,

A strange year is coming to an end and we have been busy reducing technical debts, have enlarged the team and after the big topic "platform independence" we have started again to implement visible features. I very much hope that the coming weeks will not throw us off track again and formulate an ambitious winter goal joyfully, but nevertheless carefully.

COMPLETED FUNCTIONALITY

Platform Independence

We have replaced the BrixWire runtime environment and are now able to run on both Windows and Linux platforms. A WebHost, like IIS, is no longer necessary, but is recommended and is still used for installations.

- ▶ BrixWire integrates better into your technical infrastructure and if you host us cloud based, you benefit from lower rates for Linux systems.

API

BrixWire now has a REST Level 2 API, both for stories and administrative tasks. The API is based on REST Level 2 and uses ODATA version 4 as the standard for building queries in the URLs, allowing the client to decide on filtering, paging and sorting. A YAML based documentation can be provided if you want to integrate against the BrixWire API.

- ▶ BrixWire functionality can be integrated into other applications
- ▶ Automated publish workflows are now also possible via external solutions
- ▶ For us, the API is the basis of the Angular UI

Push notifications for browser updates

In the previous approach, updates of the database were implemented via a polling mechanism of the UI. This was changed to a server-side push message.

- ▶ BrixWire can now also be used on battery-based

tablets or smartphones, as the server-side push consumes significantly fewer resources than its polling approach.

- ▶ The latency between the arrival of a message and its display has been further reduced.

Docker

We can also be installed and operated in dockerized environments

- ▶ cloud based installations, as well as docker based monitoring and failover concepts can be used.

Updated UI

We had a setback on Angular, but we have revised the current UI, updated all libraries and moved in the new REST Api instead of the old one. The translation resources have been reduced so that the adaptation and translation of BrixWire are possible with less effort and errors.

Office Pack

BrixWire is able to import Word and Excel files as articles into the database via the Office Pack. This can be combined with a Desknet workflow or with the automated processing of attachments in a mail news inbox.

Custom Metas

The data model can be extended with customer-specific metadata. Metadata is created in the administration environment;

it is available in the news browser for attribution and search:

- ▶ attributes for campaigns, collections for data filtering.
- ▶ use Custom Metas for export

Comment function for stories

Each story can be commented and ranked

- ▶ since the comments are visible to all team members, decentralized coordination of the stories to be published can take place.

IN PROGRESS

Angular UI

Via the API we have created the prerequisites for the use of UI frameworks such as Angular. We expect from Angular

- ▶ a higher consistency of the UI
- ▶ unified customizing options

WINTER PROGRAM

This section is written to collect Your needs, Your ideas and we need Your input to drive the implementation of BrixWire into an agile direction.

Variant oriented editor

For editing existing stories or creating new stories, the focus during the cold season is on an editor and the associated metadata. The editor is designed as a stand-alone feature and aims at multichannel publishing of a story.

CMS / Print Channels

We want to finish our CMS Channels (Standard CMS), these are:

- Drupal
- Joomla!
- Typo3
- Wordpress

Library updates for UI and server

We have implemented a process that guarantees updates for both the UI and the server libraries during the release.

- ▶ Security holes, which exist due to 3rd party libraries, are quickly closed.

The UI can be customized, branding is possible

The UI can be customized via configuration.

Import of stories from AWS S3 bucket

We are able to import stories from an S3 Bucket.

- ▶ Journalists can store their stories in a bucket and BrixWire imports them for further processing.

Workflow

BrixWire contains a workflow API that can and will be used for both import and self-created content.

- ▶ Workflows, group responsibility and escalation to ensure that no story can be left behind

JWT protection of the APIs

The completed APIs are secured via JWT and made ready for use outside the internal networks.

and add plugins for the print systems that allow full synchronization.

Social Media Plugin

We will provide a social media plugin (input and output) in preparation for the summer program. Which one exactly will be, is currently still being decided.

BrixWire AnyContent Server

We are building a development and demo server in the cloud to show that it does not matter whether content is created in a CMS, a print system via BrixWire or wherever. Stories that are created can be published interactively or fully automatically on all available channels.

I HAVE AN IDEA!

Check us on:   

www.brixware.com



info@brixware.com

www.brixwire.com