BRIXWIRE Newsletter

Sommer 2020

Sehr geehrte Leserinnen und Leser,
Ein seltsames Jahr geht seinem Ende entgegen und wir waren fleißig am Abbauen technischer Schulden, haben das Team vergrößert und nach dem Großen Thema „Plattformunabhängigkeit“ sind wir wieder verstärkt in die Implementierung von sichtbaren Features eingestiegen. Ich hoffe sehr, dass die kommenden Wochen uns nicht wieder aus der Bahn werfen und formuliere freudig, aber dennoch vorsichtig ein ehrgeiziges Winterziel.

ABGESCHLOSSENE FUNKTIONALITÄT

Plattform Unabhängigkeit

Wir haben die Laufzeitumgebung von BrixWire ausgetauscht und sind jetzt in der Lage, auf den Plattformen Windows und Linux gleichermaßen zu laufen. Ein WebHost, wie IIS, ist nicht mehr notwendig, ist jedoch empfohlen und wird nach wie vor bei Installationen genutzt.

- BrixWire integriert sich besser in Ihre technische Infrastruktur und falls Sie uns cloudbasiert hosten, profitieren Sie von den geringeren Raten für Linux Systeme.

API

BrixWire verfügt jetzt über eine REST Level 2 API, sowohl für stories als auch für die administrativen Tasks. Die API basiert auf REST Level 2 und nutzt, als Standard für den Aufbau der Queries in den URLs, ODATA, Version 4. Hierüber kann der Client über Filterung, Paging und Sortierung entscheiden. Eine YAML basierte Doku kann zur Verfügung gestellt werden, wenn Sie gegen die BrixWire API integrieren möchten.

- BrixWire Funktionalität kann in andere Anwendungen integriert werden.

- Automatisierte Publish Workflows sind jetzt auch über externe Lösungen möglich.

- Für uns ist die API Grundlage der Angular UI.

Push Notifikationen für Browser Updates

Im bisherigen Ansatz wurden Aktualisierungen der Datenbasis über einen Polling Mechanismus der UI umgesetzt.
Dies wurde auf eine serverseitige Push Nachricht geändert.

- BrixWire kann jetzt auch problemlos auf akkubasierten Tablets oder Smartphones eingesetzt werden, da der serverseitige Push bedeutend weniger Ressourcen verbraucht al sein Polling Ansatz.

- Die Latenz zwischen dem Eintreffen einer Nachricht und deren Darstellung konnte nochmals reduziert werden.

Docker

Wir können auch in Dockerisierten Umgebungen installiert und betrieben werden

- cloud basierte Installationen sowie docker basiertes Monitoring und Failoverkonzepte können genutzt werden.

Überarbeitete UI

An Angular sind wir vorerst gescheitert aber wir haben die aktuelle UI überarbeitet, sämtliche Bibliotheken upgedated und die neue REST Api anstelle der alten Api eingezogen. Die Übersetzungsresourcen wurden reduziert, so dass die Anpassung und Übersetzung von BrixWire mit weniger Aufwand und Fehlern möglich ist.

Office Pack

BrixWire ist über das Office Pack in der Lage, Word und Exceldateien als Artikel in die Datenbank zu übernehmen. Das lässt sich mit einem Desknet Workflow kombinieren oder mit der automatisierten Verarbeitung von Attaches in einem Mail Newseingang.

Custom Metas
Das Datenmodell kann durch kundenspezifische Metadaten erweitert werden. Metadaten werden im Administrationsumfeld angelegt, sie stehen im NewsBrowser für die Attributierung und die Suche zu Verfügung:

- Attribute für Kampagnien, Sammlungen für Filterung der Daten.
- Nutzen Sie Custom Metas für Exporte.

Kommentarfunktion für Stories
Jede Story kann kommentiert und gerankt werden

- da die Kommentare für alle Teammitglieder sichtbar sind, kann eine dezentrale Abstimmung der zu veröffentlichenden Stories stattfinden.

Library updates für UI und server
Wir haben einen Prozess eingeführt, der releasebegleitend Updates sowohl für die UI als auch für die Serverbibliotheken garantiert.

- Sicherheitslücken, die aufgrund von 3rd Party Bibliotheken bestehen, werden schnell geschlossen.

Die UI kann customiziert werden, Branding ist mögich
Die UI kann via Konfiguration den Wünschen des Kunden angepasst werden.

Import von stories aus AWS S3 bucket
Wir sind in der Lage aus einem S3 Bucket abgelegte Stories zu importieren.

- Journalisten können Ihre Stories in einem Bucket ablegen und BrixWire importiert diese zur Weiterverarbeitung.

IN ARBEIT

Angular UI
Über die API haben wir die Voraussetzung für die Nutzung von UI Frameworks wie Angular geschaffen. Wir erhoffen uns von Angular

- eine höhere Konsistenz der UI
- vereinheitlichte Customizingmöglichkeiten

Workflow
BrixWire enthält eine Workflow API, die sowohl für den Import als auch für selbst erstellten Content genutzt werden kann und wird.

- Workflows, Gruppenverantwortung und Eskalation so dass keine Story liegenbleiben kann

JWT Absicherung der APIs
Die fertiggestellten APIs werden über JWT abgesichert und bereit für den Einsatz außerhalb der internen Netze gemacht.

WINTERPROGRAMM

Dieser Abschnitt ist geschrieben, um Ihre Bedürfnisse, Ihre Ideen und Anregungen zu sammeln, und wir benötigen Ihren Input, um die Implementierung von BrixWire in eine agile Richtung zu lenken.

Varianten orientierter Editor
Für die Bearbeitung vorhandener Stories oder die Erstellung neuer Stories liegt der Schwerpunkt in der kalten Jahreszeit auf einem Editor und der zugehörigen Metadaten. Der Editor ist projektiert als eigentständiges Feature und zielt auf das Multichannel Publishing einer Story

CMS / Print Channels
Wir wollen unsere CMS Channels (Standard CMS) fertigstellen, das sind:

- Drupal
- Joomla!
- Typo3
- Wordpress

und Plugins für die Print Systeme hinzufügen, die eine vollständige Synchronisation erlauben

Social Media Plugin
Wir werden, vorbereitend für das Sommerprogramm ein Social Media Plugin (Input und Output) zur Verfügung stellen. Welches genau das sein wird, ist derzeit noch in Entscheidung.

BrixWire AnyContent Server
Wir bauen einen Entwicklungs- und Demoserver in der Cloud auf, mit dem Ziel zu zeigen, dass es keine Rolle spielt, ob Content in einem CMS, einem Printsystem über BrixWire oder wo auch immer entsteht.
Entstandene Stories können interaktiv oder vollautomatisch auf allen zur Verfügung stehenden Channels publiziert werden.

ICH HABE EINE IDEE