

ZINE W G A M

ISSUE 1/2019 Magento community journal www.magezine.co

**Read what
our columnists
have to say:**

Joshua Warren,
Ben Marks and Guido Jansen

**The most
experienced
Magento
developers and
architects:**

Igor Miniailo,
Riccardo Tempesta,
Mark Shust, James Zetlen

Hot topics!

Migration to Magento 2,
Docker, Service Isolated
Architecture, PWA Studio



Magento[®]
An Adobe Company

www.magento.com

MAGEZINE

We've been working with Magento for 10 years and over time we see a growing interest in the platform around the world. We're not surprised by this fact at all.

However, we've noticed the lack of a single medium that would gather knowledge useful to merchants, developers and all Magento's enthusiast. Everything else is there – international conferences, meetings all over the world, training, hackathons. Not to mention thousands of blogs, forums and social media profiles where experts share their experience. Although, there is no periodical magazine that could comprehensively report what's going on in Magento world, and which would be a platform for sharing news, tips and know-how with the entire community. So the idea of Magezine was born.

And finally, after long hours of work and a few moments of doubt (but who doesn't have them, right?) we're happy to give you the first issue.

Let us know what you like about it, what we still need to work on and what content is most attractive to you. We're counting on your suggestions – we truly want you to have an influence on Magezine. After all, by working together we're getting stronger as a community.

Sounds good, doesn't it?



Borys Skraba

Editor-in-Chief of Magezine

Magefresh _____ **4**

MERCHANTS ZONE

Time has come!

Last call for migration from Magento 1 to Magento 2 _____ **6**

How to effectively manage products in Magento? _____ **14**

Omnichannel adventures with Magento _____ **20**

CASE STUDY

How TOUS Boosted Sales with Multi-Source Inventory (MSI) _____ **25**

Stop Press! DC Thomson Launches on Magento Commerce 2 _____ **28**

Inside Paul Smith’s Magento Upgrade.
Commerce without Compromise? _____ **30**

DEV ZONE

The right tools for the right job in Magento 2 _____ **32**

A long way from Monolith to Service Isolated Architecture _____ **40**

Docker for Mac filesystem volume mount
approach for performance _____ **48**

INTERVIEW

Behind the scene. Interview with James Zetlen,
PWA Studio Magento Contributor _____ **54**

COLUMNS

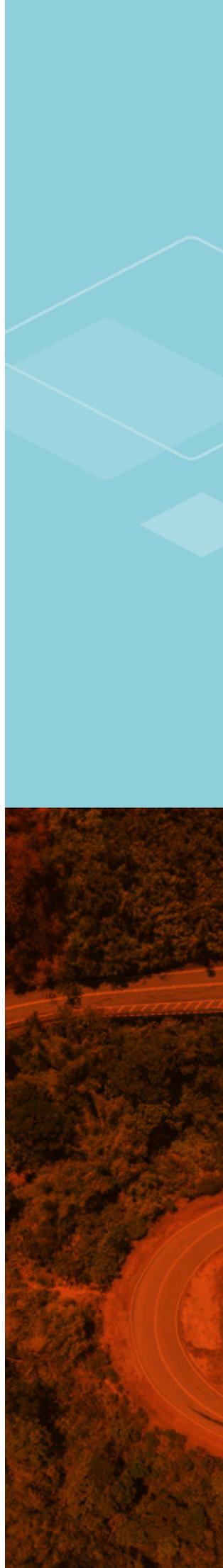
Guido Jansen:
Power of Magento community _____ **39**

Joshua Warren:
What We’re Up To in Magento Association _____ **53**

Ben Marks:
The Next Leader of Magento _____ **58**

Magento events _____ **60**

Solution & technology partners showcases _____ **61**





**Docker
for Mac
filesystem
volume mount
approach for
performance**

48



40

**A long way
from Monolith
to Service
Isolated
Architecture**



**Omnichannel
adventures
with Magento**

20

TOP 5 Magento Contribution Partners in the first quarter of 2019

On April 12, Magento announced the results of the Partners Contribution Rewards Q1 2019. So far, 2019 has seen incredible support from the Community and Partners in Magento core contributions and numerous special projects.

<p>First place:</p> <p>Atwix</p> <hr/> <p>Points: 3055</p> <p>PRs creates 101 and accepted 99</p> <p>Issues: 22</p>	<p>A continued leader, Atwix is an e-commerce agency focusing on B2C and B2B end-to-end implementations, assisting in strategic planning, design, website development, and post-launch maintenance. Atwix is a long-time Magento Solution Partner and one of the first Magento 2 Trained Partners in the world, having offices in Austria, Slovakia and Ukraine. Atwix's team of seasoned and committed professionals helps its clients achieve top results in every aspect of their online businesses.</p>	<ul style="list-style-type: none"> • The team reviewed 194 PRs by maintainers, which we greatly appreciate! Thank you, Atwix! • Additionally they covered 25 PRs with different type of tests. • Complex changes in the EAV module: https://github.com/magento/magento2/pull/20526 • Provided great test coverage for the GraphQL project. • Involvement to the coverage of shipping cart by GraphQL endpoints. • Top contributor on the new project for functional tests migration: https://github.com/magento/magento-functional-tests-migration/pulls
<p>Second place:</p> <p>Krish</p> <hr/> <p>Points: 2857</p> <p>PRs creates 248 and accepted 240</p> <p>Issues: 33</p>	<p>Krish is a full-service Digital Commerce Agency specializing in Magento consulting, design and development, implementation, and optimization to provide personalized business solutions. For over 14 years, they have provided solutions for more than 3,000 clients and 300 Magento projects. A passionate company committed to curated experiences and custom development, this team includes eCommerce Consultants, 74 Magento Certified Developers, UI/UX Experts, Technical Analysts, Marketing and Technology experts.</p>	<ul style="list-style-type: none"> • Top contributor to the DevDocsrepository (74 PRs), submitted 66 complex changes to the documentation with 2 major doc updates. • Focused on Magento 2 core bugfixes, porting solutions between release lines, making the Magento code base cleaner with fixes for typos, docblocks, etc.
<p>Third place:</p> <p>TwoJay</p> <hr/> <p>Points: 1177</p> <p>PRs creates 133 and accepted 133</p> <p>Issues: 30</p>	<p>Two Jay is a Magento Enterprise Partner with a strong reputation as one of the UK's leading Magento agencies. With over 15 years experience designing and building eCommerce websites, they specialize in helping businesses seeking to transition to the next stage of growth. They have launched over 20 Magento 2 websites, and one of the only 5 agencies in Europe with experience implementing Magento Commerce.</p>	<ul style="list-style-type: none"> • Focused on Magento 2 core bugfixes, porting solutions between release lines (63 PRs). • Involved with the new issue verification process on GitHub.
<p>Fourth place:</p> <p>ISM eCompany</p> <hr/> <p>Points: 1172</p> <p>PRs creates 85 and accepted 85</p> <p>Issues: 19</p>	<p>With over 25 years of experience and more than 200 strategic, online marketing, creative and technical e-Commerce professionals worldwide, ISM eCompany delivers end-to-end e-commerce solutions for (enterprise) retailers, brands, department stores, and wholesale businesses. Their in-house Magento product team and online marketing department create marketing driven Magento technology, combined with a strict quality assurance program for Magento coding.</p>	<ul style="list-style-type: none"> • Started contributing to the MSI project. • Focused on code base improvements, bugfixes, documentation updates.
<p>Fifth place:</p> <p>Interactiv4</p> <hr/> <p>Points: 1108</p> <p>PRs creates 2 and accepted 2</p>	<p>Interactiv4 is a full-service digital agency that has been developing eCommerce projects for over 10 years. Interactiv4 offers a suite of services, including Magento design and development, custom integrations, marketing and content strategy, CRM training, and technical maintenance. Interactiv4 brings an international approach and resources to the Magento space to help businesses meet and exceed their goals with their websites, while thinking outside the box about how eCommerce can be the most powerful tool.</p>	<ul style="list-style-type: none"> • The team completely focused on incoming Pull Requests, reviewing and processing 96 PRs on GitHub.

Source: <https://community.magento.com/t5/Magento-DevBlog/Partners-Contribution-Rewards-Q1-2019-Rankings-Announced/ba-p/128083>

DEVELOPERS

New Magento releases and security updates from March 26

At the end of March, Magento released Magento 2.3.1 Open Source and Magento Commerce, including a comprehensive tool for editing Page Builder content, increasing inventory and performance, as well as software and security updates. On the same day, updates for Magento 1, 2.1 and 2.2 were released. These releases include important security improvements and it is recommended that you prepare yourself for a quick update of Magento.

Source: <https://community.magento.com/t5/Magento-DevBlog/New-Magento-Releases-and-Security-Updates-Coming-March-26-2019/ba-p/125807>

DEVELOPERS

PWA Studio 2.1.0 has been released

The tools provided by Magento PWA Studio project allows to create websites that are fast, mobile-friendly, and reliable. The new v2.1.0 release is compatible with Magento Open Source and Commerce v2.3.1.

- Updates for GraphQL queries
- New query validation tool
- Increased unit tests coverage
- DevDocs updates

Source: <https://community.magento.com/t5/Magento-DevBlog/PWA-Studio-2-1-0-has-been-released/bc-p/127496#M490>

MERCHANTS

Magento introduced Adobe Commerce Cloud

This integration enables easier management, personalization and optimization of consumers' shopping experiences at every point of contact with the brand. Also Adobe Experience Cloud offers online store owners the ability to measure business results and optimize them with AI support.

More information: <https://www.adobe.com/commerce/magento.html>

MERCHANTS

Retailers can no longer ignore Progressive Web Apps

The implementation of PWA is essential. This solution allows for easy access to the mobile version of website directly from the search engine and does not require additional installation activities from the consumer. Since January 2019, Magento has released PWA Studio, which enables retailers and developers to create a reliable, fast and attractive mobile experience for their customers. For online stores that want to increase conversion rates and engagement – PWA is a must have for online stores that want to increase conversion rates and engagement.

Source: <https://magento.com/blog/magento-news/magento-announces-availability-pwa-studio>

MERCHANTS

Adobe announced next generation of the Adobe Exchange Application Marketplace

Magento customers now have access to more than 10,000 applications, extensions, data and services on the market to optimize every step of their journey (Adobe Creative Cloud, Adobe Document Cloud and Adobe Experience Cloud).

Source: <https://magento.com/blog/magento-news/adobe-announces-next-generation-adobe-exchange-application-marketplace>

"He who moves not forward, goes backward" – the author of this quote, so eagerly used by business coaches today, is a well-known writer Johann Wolfgang von Goethe. Of course, when he penned these words, the 18th century German author did not actually refer to selling on the Internet, but the idea has remained unchanged for years – whoever does not adapt to changes, has no chance to survive. E-commerce is no exception. Consumer behaviour is changing, the market is changing, and so do the technologies. Therefore, following trends and looking for new solutions to optimize work or increase sales conversion is the bread and butter of all merchants.



Time ha

Last call for migration
from Magento 1
to Magento 2



s come!

In every developing online shop there comes a moment when the current version of the platform, on which it is based, does not keep pace with the growing needs of customers or market requirements. So, it is the right time to think about changing the system or upgrading it to a new version. Retailers have been facing such a challenge (or in other words – opportunity)

after the end of the Magento 1 support was announced. Yet, apart from the difficulties that have to be dealt with when migrating the system, a wide range of new chances opens up for entrepreneurs. How to recognize when the best moment for migration is? What is the best strategy to adopt? And what are the advantages of the new version? Let's find out.

WHY SHOULD YOU MIGRATE NOW?

At the beginning of 2017, Magento announced that it would stop supporting the first version of the system in November 2018 (Magento 1). What does this mean in practice? No official support, no platform development, no security patches, no new features. The conclusion is simple – merchants who decide to stay on M1 take a lot of risk. Simply put, for big players, the transition from Magento 1 to 2 is not only a matter of choice, but also a necessity to stay on the competitive market.

Joe Ayyoub, Magento Operations Support Director,

issued a statement in September 2018 (e.g. two months before the originally planned deadline) and assured that software support will be provided for Magento Commerce 1 and security patches for Magento Open Source 1 by June 2020. This is the response to the needs of merchants, for whom the first date – November 2018 – was too short to prepare the store for migration.

WHAT IF YOU DECIDE TO STAY ON MAGENTO 1?

In theory, the transition to a newer version of the platform is not obligatory yet. In practice, however, this means that any problems that occur in the old version will have to be dealt with on your own. Therefore, it

HOW TO PLANNING THE MIGRATION?

When planning the migration, but even before its implementation, it is worth following the steps below:

- **Making a full backup of the store** – together with databases, files, system information.
- **Create a clone of the first version of Magento**, which will be used during the migration.
- **Perform an analysis of shop contents** and make the decision whether all of the information there is still needed.
- **Check whether the extensions used in Magento 1 are also available for Magento 2** (e.g. in the Marketplace).
- **Transfer data from the Magento 1 clone.**
- **Remove unnecessary or outdated data** (old passwords, logins, product browsing history).

After going through the entire preparatory stage, you can focus on the actual migration – and its process can be divided into four main parts:

- 1. Theme migration**
Keep in mind that it is not possible to migrate a theme from Magento 1 to 2. You need to create one that meets the needs of both system users and customers from scratch. You can do it yourself or buy one of the ready-made templates from Magento Marketplace (remembering that this one will never be fully customized).
- 2. Extensions migration**
Extensions are an important component of an efficient shop – they provide new functionalities or enhance those already available. Therefore, you should install the selected extensions, bearing in mind that the extensions from Magento 1 will not work on Magento 2. However, a great number of extension providers for Magento 1 have also created equivalent versions for Magento 2.
- 3. Customizations migration**
There are cases that a customized code used on M1 is compatible with M2 after a little help of Code Migration Toolkit. However, taking into account the differences in the structure of Magento 1 and 2, the migration of the code may involve additional work of developers.
- 4. Data migration**
We are at the end of the road – you still need to transfer your data and settings to Magento 2. For your convenience, use Magento 2 Data Migration Tool – with this tool you can migrate such important data as products, orders, categories, store settings and configurations, etc.

Bear in mind that some data can be inconsistent after migration, so it's crucial to verify data consistency at each stage of development.

is worth knowing the risks involved in the decision to stay on Magento 1 and make a pros and cons balance. Below, we present in just a few brief points the most important threats that may affect merchants who decide to stay on M1 or delay the decision to migrate.

Weaker security

In June 2020 Magento will stop the delivery of security patches for Magento 1 Community and Enterprise, which means no protection to recently found vulnerabilities. Even before this deadline, in December 2018, all the technologies below PHP 7.0 will not be updated, so there will be no new security firewalls for the software. In other words – by staying on Magento 1 you expose your shop and your customers to highly risk.

No module updates

Since Magento 1 is left without support, a large number of its modules will become out of date. This should not be surprising – most of Magento's developers will be focused on building and improving Magento 2. It is worth considering that many of the Magento 1 modules are crucial for the proper performance of the platform, and without their updates, the site may not work right.

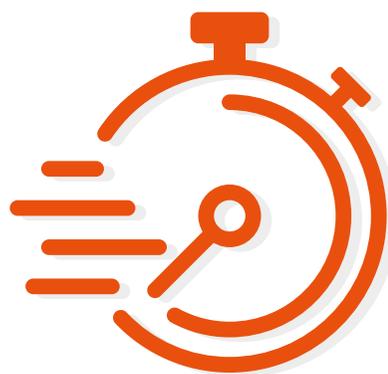
Limited number of developers

The number of certified Magento partners compared to the shops built on this platform is very limited. The longer merchants delay the decision to migrate, the less experienced implementation agencies will be willing to take up this challenge in such a short time. In addition, fewer and fewer certified developers will want to participate in Magento 1 projects. Companies prefer to spend their training budgets for employees on getting to know the new version better and obtaining certificates for Magento 2.

Short time

A very short deadline is one of the main factors determining the upgrade to the new version of the platform. When the announced end-of-life for Magento 1 comes, it may turn out that many stores will not manage to migrate. Such process takes from 3 to 7 months, depending on the number of developers involved. It should also be taken into account the complexity of your store, the number of extensions, functionality, the number of SKUs and many other factors influencing the time needed to make a thorough upgrade.

MAGENTO 2 BENEFITS



50%
page loading speed
than Magento 1

Checkout
38%
faster than Magento 1

Capability to taking
117%
more orders per hour
than Magento 1

Can handle
2.1 mln
more page views
per hour

MAGENTO 2 BENEFITS



Design **adapted to mobile devices** with any screen resolution.

Magento 2 PWA Studio for **better mobile experience to boost conversion** rates and increase engagement.

Tools for customising content and adding local preferences.

Order summary on one card eliminates the risk of error by the customer, e.g. when choosing a shipping method, which significantly improves the customer experience.

Reduced number of steps in shopping cart check out (from six to two) – less risk of cart abandonment.



HOW DOES THE WHOLE PROCESS LOOK LIKE?

Migrating from Magento 1 to 2 will not be a simple, fast and cheap process – it is best to clarify this issue from the very beginning. It is also impossible to resolve it with a single click – it just doesn't work like that. Remember that Magento 2 is a completely different system than Magento 1.

The risk associated with not migrating to M2 is so high that for many merchants the main question about the migration should not be whether but when. It is therefore necessary to take into account not only the complexity of your system when setting the date of release of a new version, but also sales peaks in the industry – then it is best to focus on ongoing tasks, and leave such large projects as migrations for a “less busy” period.



Thanks to the applied technologies, Magento 2 better adapts to the customers' needs, improves their shopping experience, which directly indicates higher sales conversion. It has improved overall performance and security. It also gains with faster and more intuitive administration panel. Magento 2 is also adjusted to the omnichannel and mobile environments.

HOW CAN YOU BENEFIT FROM MIGRATION TO MAGENTO 2

Thanks to the applied technologies, Magento 2 better adapts to the customers' needs, improves their shopping experience, which directly indicates higher sales conversion. It improves overall performance and security. It also gains with faster and more intuitive administration panel. Magento 2 is also adjusted to the omnichannel and mobile environments. These arguments should be enough to convince all those who hesitate to migrate. Of course, there are more advantages to it – not only customers, but also merchants, e-commerce managers and developers can benefit from it. How? Below we present the main perks of the transition to Magento 2.

Increasing website's speed and performance

- Faster server response time for all activities on the website.
- Increased database flexibility and scalability to cope with peak sales workloads.
- Higher speed compared to Magento 1 thanks to full-page caching, which Magento offers with Community and Enterprise versions.
- Magento 2 uses Varnish cache and provides a good Full Page Cache as well as load balancer on many application servers.
- Faster loading of the store's websites such as CMS pages, listings (category, search results) and product's page.
- Faster check out of clients and guests.
- Magento 2 is able to process 117% more orders than Magento 1 using the same server resources.

- Magento 2 supports 2.1 million page views per hour.
- Improved indexer that increases search results relevance
- Minified JavaScript, CSS and HTML and compressed images to reduce file size for faster loading.

Simplified and shortened path to purchase

- Magento 2 allows to continue shopping without having to fill in a login or registration form.
- The content is tailored to customer needs, thanks to cross- and upselling tools or product recommendations.
- Easy integration with payment platforms and shipping companies.
- Order summary on one card eliminates the risk of error by the customer, e.g. when choosing a shipping method, which significantly improves the customer experience.
- Reduced number of steps in shopping cart check out (from six to two) – less risk of cart abandonment.

Mobile friendly & responsive design

- Design adapted to mobile devices with any screen resolution.
- Magento 2 PWA Studio for better mobile experience to boost conversion rates and increase engagement.
- Tools for customising content and adding local preferences.
- Building and managing all channels with one code base, one deployment and app.
- Innovative commerce and CMS theming.
- Purpose-built developer tools for fast prototyping, helpful debugging, rich feedback, and increased productivity.
- Administration panel adjusted to touch screens makes it easy to manage a shop from a phone or tablet.

Intuitive and user-friendly admin panel

- Magento 2 Admin Panel has been completely redesigned bearing in mind usability for the beginners as well as advanced users.
- It makes it easier to find information, navigate through all sections of the panel, and thus ensure more efficient site management.
- The extended M2 dashboard shows: sales volume from the beginning, recent orders, average order value, main search terms, best-selling products and key customers, deliveries and quantities – which allows you to monitor the status of your business in real time.
- The panel can be adjusted to individual needs, which increases the efficiency of managing products, orders and customer data.
- Uploading new products to the panel is simplified by four times faster import.

More efficient website management

- Ability to handle more daily orders and SKUs.
- Easier creation and management of content on the website.
- Improved search and SEO components.
- Alignment with omnichannel strategy – the new engine is adapted to combine on- and offline sales.
- Easier scalability – Magento 2 allows you to scale your database for optimization during busy periods, such as holidays or sales.
- Enable more than one administrator to create and edit information on products without data conflicts.

More advanced reporting

- Updating of the report page in real time.
- Advanced reporting functions available in Magento Admin allow to create twenty reports, including the number of orders, AOV (average order value), number of registered accounts, or orders for products (including bestsellers).

Tools for marketing automation

- With the built-in Dotdigital tool users can connect customer data, powerful insights, and automate intelligent messages across email, SMS, social, and more.

Wide range of B2B functionalities

- Efficient management of company accounts – possibility of categorizing and organizing customer's information as needed.
- Fast purchase by entering the SKUs or uploading a CSV file with the required data.
- Possibility of creating various lists of frequently purchased products and assigning them to specific clients. Items from a requisition list can be easily added to the shopping cart or moved or copied from one list to another.
- In the backend of Magento 2.2, there is a segment for quote requests. Retailer can filter selection, use the search function to find different quotes and customize them.
- Better management of purchasing conditions for B2B customers.
- Customized catalogues and price lists – makes it possible to control prices, products, and categories shown to customers; to create different price lists, and to assign them to B2B clients.
- Updated API enables ERP integrations for different features.

System architecture based on the latest technologies

- Magento Core architects and developers have improved the technological stack by adding several technologies, including Composer, AMQP, REST API, Elasticsearch.
- Magento 2 supports only the latest versions of PHP – 7.1 and 7.2 These versions contain performance improvements, which have a significant impact on increasing the speed of the store.
- System for better diagnosis of problems for Magento support.
- Possibility of split databases into e.g. orders, catalogue (database sharding).
- Reduced unnecessary search engine operations on the client's side thanks to the improved JavaScript.
- The Magento approach with PWA Studio through open source on Github will allow development teams to familiarize themselves with additional resources and technology required.
- Better caching for static content; native support for CDN.
- Ready-made tools for easy creation and implementation of a store in the Magento cloud infrastructure.
- Enhanced algorithms (SHA-256) for passwords, making them less vulnerable to dictionary types of attacks.

Improved SEO components

- Creating search engine friendly URLs.
- Snippets that improve search results in SERPs.
- Native support of XML sitemaps.

GDPR compliance

- Magento out of the box helps meet GDPR requirements.

Extensions

- Conflicting extensions. There are many extensions to Magento on the market offered by external companies. However, the problem arises when two extensions overwrite and try to change the same functionality. It is possible to solve this problem manually, but it results in a lot of work and resources. In Magento 2, instead of overwriting the code, plug-ins enable its interlocking. Magento Marketplace Quality Assurance team ensures that submitted modules are not vulnerable to conflicts with other modules both with Static Code analysis, as well as manual Code Review.
- Cheaper extensions. Installing and updating new extensions has become easier and cheaper. Composer reduces implementation costs as it is the package manager who is responsible for installing and updating the package.

Taking into account all of the above, it is clear that Magento 2 offers much better solutions than Magento 1 in all aspects. It is a more technologically advanced platform, which is perfectly suited for mobile and omnichannel environments. Despite the unquestionable benefits of moving a store to M2, the process itself will not be completely painless – and you should be prepared for that too.

PREPARING FOR THE BETTER SOLUTION

Migration from Magento 1 to Magento 2 is a great opportunity for merchants to move their store a few levels higher. In such a highly competitive environment as today's e-commerce industry, they should choose solutions that provide customers with a better shopping experience, and people on the other side of the screen with better store management system. By migrating to a newer version of the platform, the store is provided with greater security against hacking attacks (and e.g. leakage of customer data), and its efficiency and scalability are improved.

Each process, such as shop migration, is also an opportunity to diagnose the condition of the shop – to explore what worked and what did not in the previous version – and to develop good practices for the future. ●

By Magezine Team

Sources:

1. Is it worth it to switch from Magento 1 to Magento 2, <https://www.pronkoconsulting.com/ecommerce-blog/is-it-worth-it-to-switch-from-magento-1-to-magento-2>
2. Jillian Hufford, Why Migrate from Magento 1 to Magento 2, <https://www.nchannel.com/blog/migrate-magento-1-magento-2/>
3. Joe Ayyoub, Supporting Magento 1 through June 2020, <https://magento.com/blog/magento-news/supporting-magento-1-through-june-2020>
4. Magento 2.3.x technology stack requirements, <https://devdocs.magento.com/guides/v2.3/install-gde/system-requirements-tech.html>
5. Magento Data Migration Tool, <https://github.com/magento/data-migration-tool>
6. Migrating to Magento 2, https://magento.com/sites/default/files8/2018-10/8262_Community_Insights_Corra_R2.pdf
7. Migration overview, <https://devdocs.magento.com/guides/v2.3/migration/bk-migration-guide.html>
8. Peter Sheldon, Magento Announces Availability of PWA Studio, <https://magento.com/blog/magento-news/magento-announces-availability-pwa-studio>
9. Progressive Web App Primer for Early Adopting Merchants, https://magento.com/sites/default/files8/2018-11/Magento_PWA_Primer_r3v1.pdf
10. Upgrading to Magento, <https://magento.com/sites/default/files8/2018-10/8282-Community-Insights-Perficient-M2-2.pdf>
11. Upgrade to Magento Commerce 2, <https://magento.com/sites/default/files8/2019-04/upgrade-to-magento-commerce2-infosheet.pdf>



How to effectively manage products in **Magento?**

Five – we use that many senses in traditional shopping. All that remains on the Internet is sight and hearing – the first one being the leading sense. An online customer can check the product through the information that describes it, so it must be complete to convince them to shop. Without this, online shopping experience does not exist.

PRODUCT INFORMATION & CUSTOMER EXPERIENCE

Consumers' assessment of products is linked to their experience - how they perceive the brand and the physical attributes of individual products. When buying from a stationary store, besides emotional factors (brand loyalty), the customer assesses the value of the product through all the senses. For example, when buying perfume, these would be: touch (the quality of the packaging material), sight (the attractiveness of the packaging), smell (the attractiveness of the fragrance) and hearing (the sound of the diffuser). In the e-commerce channel, it is much more difficult. So how to manage products in an online store to build

a positive consumer experience? Magento system includes 6 types of products: simple, group, configurable, bundle, virtual, downloadable. Getting acquainted with their division and possibilities will not only improve the management of the store but also make things easier for the customers who come to it to buy.

PRODUCT INFORMATION IN THE MAGENTO SYSTEM

As you prepare to sell with Magento, it is worth considering how to present product information in the system. In each e-commerce system, the product represents a set of data about specific parameters. So, assortment management boils down to managing a set of information about products that define them. Product information is divided into 4 areas:

- **information defining the product** (SKU/EAN, size, weight, norms/standards) certificates),
- **information affecting the consumer's choice of product** (descriptions, photographs, animations, videos),
- **information on product "shipments"** (stock availability, orders, advanced shipping information),
- **information related to the measurement of product effectiveness** (forecasts, analytics, sales reports).

Magento platform uses similar systematics. The effectiveness of sales in the e-commerce channel depends on how we manage product information.

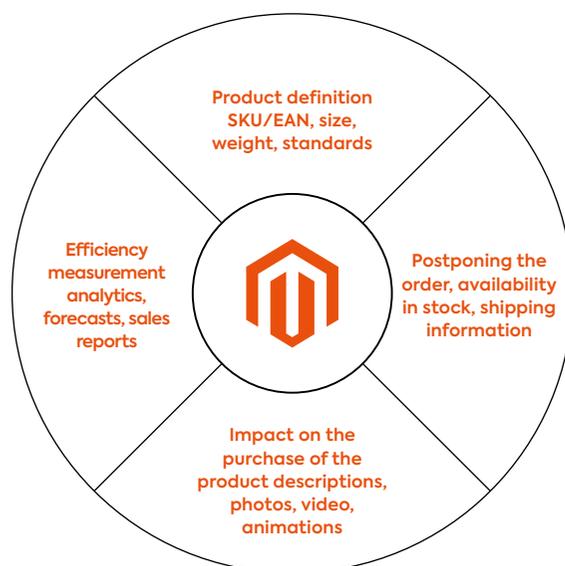


Fig. 1. Division of product information in Magento.

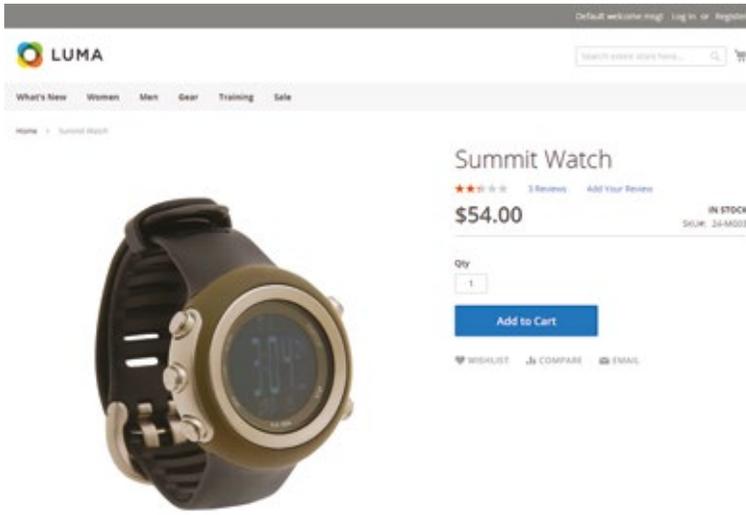


Fig. 2:An example of an on-line page with a simple product. source: docs.magento.com

From the user's perspective, Simple product is not complicated. From the store manager's side, it has many functions influencing sales. Simple products are the foundation of Magento system and are associated with the pricing, promotion and storage policy of other types of products.

PRODUCT TYPES IN THE MAGENTO SYSTEM

Magento platform makes it possible to manage the assortment through 6 types of products. In this article, the focus is on the sale of physical goods, not virtual. Therefore, the solutions for 4 product types (simple, group, configurable and bundle) are presented in more detail.

Simple product

A simple product is the basic and the most important product type in Magento system. It is a representation of a physical product in a warehouse that has a unique Stock Keeping Unit (SKU).

According to the above mentioned definition, a product has certain attributes that characterize it and are reflected in its physical characteristics. Let's take a look at an example: a pair of high heels in size 39, a heel height of 8 cm, made of red leather, which has a SKU ID of 12345.

In this case, the product page is the least complicated and has a clear call to action. The only action that separates the customer from the purchase is the "Add to Cart" button.

Practically every online shop can be built on the basis of an assortment consisting of simple products only. However, the management of such a system is not always optimal. This is the case for products with variations and different configurations, e.g. the same type of high heels can have different colours, sizes and materials, which means that the same type of shoes can have different colours, sizes and materials:

- **All variations will be available in the product list**, so for only 1 type of product in 3 sizes, 3 colours and 3 types of material the listing of products will contain 27 items of simple products.
- **The content can be duplicated**, which has a negative impact on SEO. The same pictures, names, product descriptions and ratings may appear multiple times in the list.
- **Visual product merchandising will be difficult and ineffective.**

Custom options in simple products

Custom options are an easy way to offer different product variations. It is worth knowing that they are not based on attributes; therefore, they cannot be used as filters on the product list, as well as none of the variations can have separate stock levels (which distinguishes them from configurable

products). Magento has introduced limitations to tying simple products using non-standard options when creating configurable, group and package products.

You can select custom options:

- **text fields**
- **files to upload**
- **select fields (drop-down, radio buttons, checkbox, multi select)**
- **date and time entry field**

For each of the selected options, a rigid or percentage dependence of the price for a specific option in relation to the main price of the product can be applied. Although custom options have limitations, they work well when selling products combined with services. A product strategy based on such a solution can be implemented, for example, when selling T-shirts with the "personalization" option. Let's assume that we have white and specific SKU shirts in stock. We allow the user to choose the options of dyeing to specific colours (you can use select fields). We can also add the option of sending a file, e.g. logo for printing and a date entry field, thanks to which the customers will determine the approximate, expected date of product realization.

Group product

A group product is created by tying several simple products (without custom options/ custom options) in one set.

On the product listing, in a given category, this type of product appears with a unique photo, name and price presentation for each of the elements.

The most important thing when deciding on the use of such type of product is the way it can be purchased. Group products are ideal for selling products in larger quantities, e.g. B2B e-commerce sales, although they can be successfully used in B2C product strategies.

An interesting example are Internet delicatessen, which sell products grouped into specific recipes, e.g. a recipe for pizza. In this case, the group product thus created will be linked to simple products: yeast, flour, ham, etc. By choosing this type of products, the customer will be able to order ingredients that are missing for pizza preparation or all included in the set. Importantly, it will decide for itself whether to increase or decrease the number of individual ingredients.

Group products also have limitations. First of all, it is not possible to manage the price of the created set separately. The presented price is always in keeping with the cheapest simple product associated with the set. From here the information "prices start from:" appears. This means that promotional activities, e.g. seasonal discounts/special prices, etc. can mainly take place through price management of related products. On the other hand, it is possible to get promotions on the whole set only by configuring promotional rules of the basket.

Configurable product

A configurable product - often referred to as a variation product type - resembles a simple product with custom options for each variation. However, here each option is an attribute associated with separate simple products (without using custom options).

Therefore, by choosing any options in a configurable product, Magento system checks the availability of the stock of the related simple product.

Configurable products are suitable for sale in the lifestyle/fashion industry, where each of the variations, each of the colours and each of the sizes appears as a separate SKU with an individual stock.

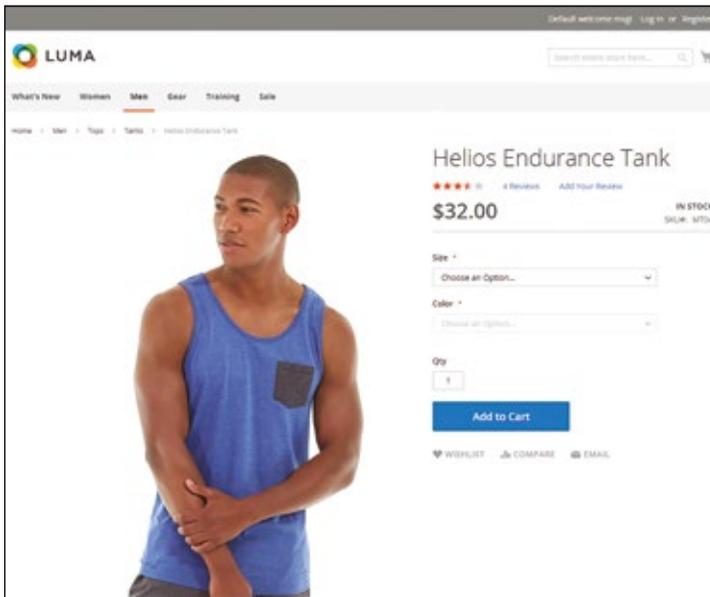


Fig. 4: Appearance of the page with configurable product
 source: docs.magento.com

When creating configurable products, remember that it is only possible to bind simple products that do not have custom options. Simple product visibility settings allow to create a strategy in which configurable products will be visible to the user, and each option that is a simple product will only be visible from the administration and warehouse side. Such a solution is widely used in the case of e.g. shoe or clothing sizes, where a configurable product has been assigned all merchandising elements (price, descriptions, photos, etc.), while related simple products by size function only in the administrative area of sales and warehouse management.

Price management in configurable products is the same as in the case of simple products. Multiple price parameters can be set. However, it is worth remembering that for this type of products the system takes into account the price parameters set for a configurable product. Native Magento does not take into account the prices of related simple products. The only option to make the price dependent on the selected variation is to set prices (in percentage or amount) in the related

products section, which is added to the main price of the configurable product.

Native Magento system does not change the URL of the page when selecting a variation. It is important as it is not possible to configure it (assuming no additional programming work) in such a way that, for example, for a selected colour variant, it would be possible to send a URL that would relate directly to the selected variant. Configuration strategies should therefore be considered carefully. They can affect promotional activities: inside the shop (lack of visibility of variants on the product list), outside in Google, or in price comparison engines (which need direct links to selected variants).

It is important that you can create products with complex configurations that will consist of simple products with more than one attribute. It is also possible to enable colour-swatches, e.g. graphic representation of selected attributes (it is possible to specify thumbnail sizes both on the product card and the list of products).

It is worth noting that it is possible to set the visibility of a selected attribute on the list of products, the click on which will change the thumbnail of the product image on the list, according to the one assigned to the variant (simple product).

Bundled products

A bundled product allows the customer to build their own product from among the various available assortment options (simple products without the use of non-standard options/custom options).

Imagine that you want to sell sports equipment for yoga, which consists of many elements such as: mats, balls, belts and bricks. Using a bundled product, we can offer customers different variants of each of these elements, e.g. different

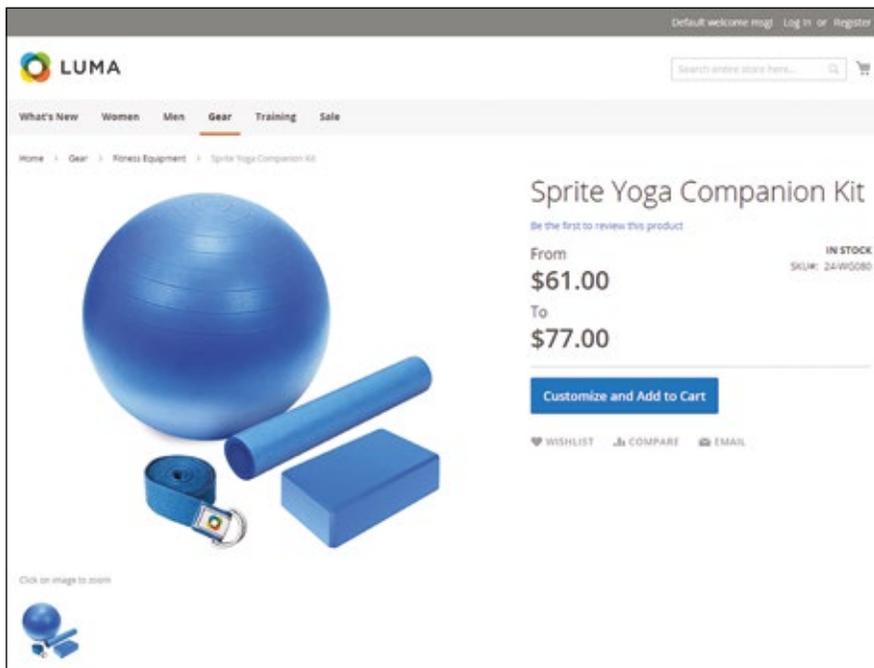


Fig. 6: Appearance of a page with a packaged product
 source: docs.magento.com

This type is primarily intended for the sale of products that consist of various components or sub-assemblies or to create sets.

They work well in the case of sales of specialist products, where within the configuration options of the "kit" we are able to support and lead the customer "by the hand" to the finalisation of the purchase. This ultimately leads to an increase in conversion and the quality of purchasing experience.

A wide range of pricing strategies for bundled products is important. As in the case of simple and configurable products, special prices can be applied, depending on the number of units purchased. Importantly, the basic price can be calculated dynamically (the price presentation can be set to "from" "to" or "not less than") or can be defined as fixed. Similarly, an SKU identifier can be created dynamically or can be set as fixed.

When choosing this type of product, one should be aware that after adding a bundled product to the basket, it is visible as a single product, and not - as in the case of a group product - individual simple products that are components of the created pack.

THERE IS NO GOLDEN RULE FOR PRODUCT MANAGEMENT

Each company has a unique way of building a competitive advantage. Differences can occur in many planes. Starting from the diversity of the assortment, unique features of the products themselves, through applied promotional strategies, specific sales models, and ending with the warehouse and logistics policy. All these factors influence the type of product used in the Magento system. Therefore, a thorough analysis of the assortment should be carried out before implementation. ●

By Magezine Team

HOW TO ANALYS ASORTMENT?

As a starting point, it is worth to apply the W3H rule and analyse each product by answering a simple set of questions, e.g.:

- **Who is the customer?** (WHO)
- **What do we sell?** (WHAT)
- **Why should the customer buy it?** (WHY)
- **How does the customer buy this product?** (HOW)

Such analysis will make it possible to identify those product features that determine the purchase and determine the different product variants/ configurations. The result will be the creation of an appropriate set of attributes, which will become the starting point for managing the assortment in Magento system.

The next key step is to analyse the most frequently used pricing strategies and promotion types for a given product/ category of products:

- **What price strategy do we adopt for a given assortment - low-price or premium price?**
- **Are there any discounts for the purchase of a specific number of products?**
- **Are there any B&G (Buy&Get) promotions? If so, it is worth specifying the most common ones.**
- **Are different prices applied to different customer groups and segments?**
- **How important is the market price suggested by the manufacturer? Does it have to be shown on the product card?**

In the next stage, it is worth analysing how the purchase process translates into logistic solutions:

- **Is it allowed to sell the products which are not available (backorders)?**
- **Are there any minimum quantities needed to make a purchase?**
- **Are there any restrictions on the maximum purchase quantities?**
- **Do we use any specific quantity increases?**
- **What conditions are related to the logistics of a given product (e.g. obligatory sale of a product+pallet package)**

For this purpose, it is necessary to impose on the areas analysed above a fundamental step or a sales model, as its specificity may preclude the application of specific product strategies.



Omnichannel adventures with Magento

Omnichannel is a necessity if sellers want to be competitive and do not want their business to die a natural death. Why? Omnichannel allows us to provide the same level of customer experience and satisfaction in both traditional and online stores, as well as in all new brand channels – social media, internet of things, marketplace, etc.

"Customer expectations go beyond a simple 'buy anywhere, anytime' mentality. Retailers must continually optimize their programs with added value and service levels to deliver a smooth customer experience for cross-channel ordering and fulfilment. For example, 54% of US online shoppers who use BOPIS service expect the retailer to notify them within an hour or less that the items are ready for pickup."¹

It's good to remember that effective implementation of omnichannel strategy concerns many areas of business operations: sales, merchandising, IT infrastructure, logistics and customer service. It's not just a matter of a well displayed web-shop on various devices, but above all of a well-functioning business process.

BE TRULY OMNICHANNEL WITH MAGENTO

What do today's consumers want? They want to buy both online and in stationary stores. They want to be sure that the shop will allow them to place an order from anywhere. They want the shop to deliver the order to the selected location. They want to get their order home, but they also want to be able to buy online and pick up the order in store. The list is really long.

CONSUMER EXPECTATIONS ARE RISING

At the end of the day, customers may not even remember which shopping channel they used. However, they will surely remember that there was no problem with buying and receiving products. And this is really important in the context of the implementation of the omnichannel strategy. How to meet the ever-increasing customer demand for trouble-free shopping? Retailers must invest in people, processes and technologies, which also requires of them to be agile and fast in action. All of this is needed in order

to create a smooth and consistent shopping experience at all customer touch points.

One of the main keys to success in implementing an omnichannel strategy in business is an accurate and complete overview of the inventory of products in all possible channels and locations.

BRIDGE THE ONLINE-OFFLINE GAP

When are we dealing with a gap in online-offline? For example, if a brand only has a stationary or an online shop. As a result, customers either cannot check their products in a traditional shop or cannot buy them online at all. It also happens that brands sell both online and offline, but these two channels work separately, e.g. customers can only buy a gift voucher from a stationary store or the product is only available online. In both cases, a gap is created between the shop's capabilities and the buyer's requirements.

Imagine such a situation – a customer enters a stationary shop and looks for specific trousers in size 40. A shop assistant verifies whether the product is in the store and it turns out that it is not available in this particular size. What does he do then? If the shop is not supported by appropriate technology, all that remains is to say "I'm sorry". However, when the shop has implemented Magento solutions, a few clicks are enough for the employee to find out about the availability of the product and then pass it on to the customer. For example, other brand stores can be recommended, where the customer will find the desired product; or the shop assistant may suggest buying it online with home delivery. In this way, the assistant armed with information does not lose the selling opportunity and the customer does not leave the shop disappointed.

91%

ETA for orders to store or home

89%

View local store inventory online

86%

Buy online and return to a store

73%

Ship to a local store

61%

Buy online and pick-up in store

Fig. 1 Web-Influenced Retail Sales Forecast 2012 to 2017 (US)
Image source: www.magento.com, source of data: Forrester Research,

Multi-Source Inventory (MSI) and Order Management System prevent similar situations from occurring and offer a number of improvements for traders who want to implement omnichannel in their business.



Fig. 2 Evolution of consumer's touch points with a brand
Source: <https://www.pwc.com/gx/en/advisory-services/assets/customers-are-calling-the-shots.pdf>

MULTI-SOURCE INVENTORY

Multi-Source Inventory (MSI) is a new Magento tool that allows merchants to manage their inventory in multiple locations so that it can properly reflect their physical inventories without the need for additional extensions. Moreover, thanks to MSI, merchants can shape sales based on many warehouses, e.g. when you order products that are in different warehouse locations.

The MSI project was carried out by contributors of the Magento Engineering Community. This makes the tool a native function of Magento 2 Open Source.

New to Magento Multi Source Inventory are Sources. The Source represents the place where the stocks are located. MSI creates a Source for every location where products can be physically located, e.g. warehouses, brick-and-mortar stores, distribution centres and drop shippers.

What does MSI give in short? Let's assume we're selling

10 dresses. So far, the exact statement that we have 5 dresses in stock in Toronto and 5 dresses in stock in Barcelona has been difficult to achieve. Now, MSI makes it possible because it determines the actual quantity that is in stock in many warehouses around the world. In this way, it increases merchants' control over their inventory and allows them to better utilize it in many sales channels.

MSI uses algorithms that track every available product in warehouses and stores. Algorithms run in the background, updating the number of products for sale, checkout and ordering shipping options.

Source Selection Algorithm (SSA) – It can be difficult to grasp multiple source locations, global customers and carriers with different pricing options, while knowing the actual inventory and finding the best shipping option. The SSA does this work for the merchants. It is used to calculate from which source an order can be delivered in the most efficient way. It uses all information that is available about customers, products, addresses, stock and sources. For example, by using the customer's delivery address, the order can be sent from the nearest department store or, if several suppliers offer the same product, the algorithm can select the option with the highest product margin.

Additionally, MSI also uses Reservation. The system tracks warehouse requests for the entire purchase process, e.g. when adding products to the basket, finalizing transactions (checkout) and managing returns. Once an order is placed, a Reservation is made to ensure that the stock is sufficient to fulfil the order. With Reservation, the checkout becomes more efficient even with large parallel sessions. In addition, it prevents over-selling of available resources.

In what situations is MSI useful? Let's imagine a theoretical scenario:

The UK online shop has delivery partners in France and Ukraine. It also has websites in 5 languages: English (UK), Spanish (ES), French (FR), German (DE). The UK, Spanish and French shops can sell products from stock available from a partner in France. The German store can sell both the stocks available from the order fulfilment partner in France and the stocks available in Ukraine. This is possible without complications thanks to MSI.

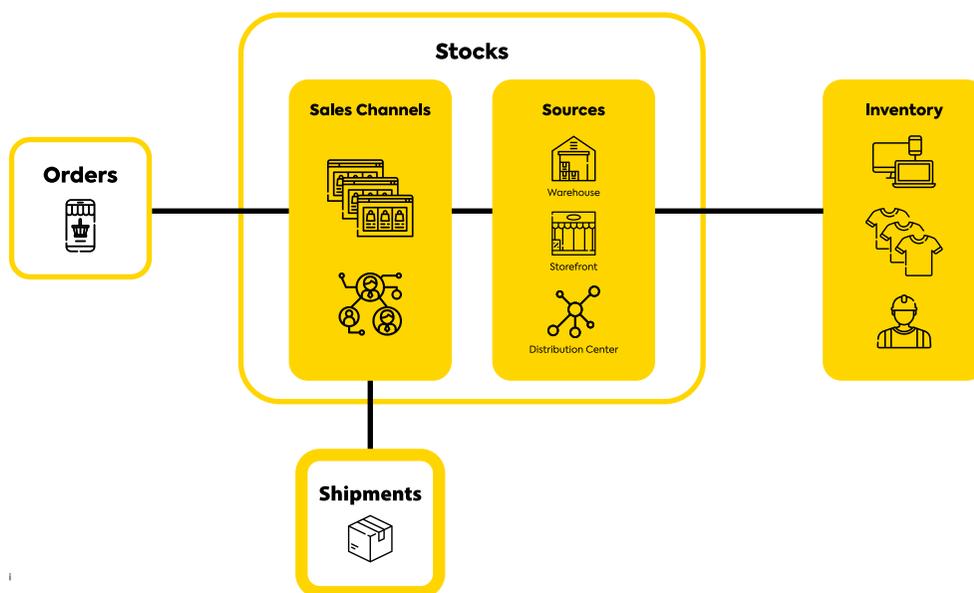


Fig. 3: Magento MSI operating diagram
 Source: <https://github.com/magento-engcom/msi/wiki/MSI-features-and-processes>

Building a scheme of warehouses in Magento requires little effort. If we want to open an additional warehouse in the Netherlands, it is enough to create a Source system and link it to the shops in Europe. Similarly, if you want to expand your sales to new markets and, for example, send orders in the USA from warehouses in different cities in the United States.

An example of a schema in Magento:

Stock:	Europe	USA
Sources:	Store NL, Warehouse DE	Warehouse NY, Warehouse LA

ADVANTAGES OF MSI

With MSI, merchants can:

- **Better manage all physical locations (Sources), and products stored there (SourceItems).** This allows them to control stock levels in each source.
- **Increase Inventory visibility for more efficient sales.** By improving the visibility of inventory, retailers know where the products are: whether in their own warehouse, at an external supplier, franchisee, retail partners, shippers, or in local and worldwide physical stores. This knowledge allows merchants to satisfy consumer expectations, affects sales growth and management of implementation costs.
- **Improve the efficiency of transaction execution (check out)** – allows traders to eliminate excessive blocking of

databases during the purchase process, adding products to the basket, transaction execution.

- **Easier to handle Drop Shipping** – set up an external source for Drop Shipping. Simply add your shipping partner to your list of available sources.
- **Control your inventory and receive low-level, up-to-date reports** – Magento supports inventory control at all physical locations (Sources) and notifies you when products in some locations end⁴.
- **Send orders in parts** – e.g. when a customer wants a specific products from the order delivered in advance. A merchant can create a separate shipment and choose a different warehouse that offers faster delivery time to set delivery in advance. The remaining items will be sent using the specified algorithm.
- **"Buy online and pick up in store" option to increase consumer convenience and shop traffic.** It is worth knowing that omnichannel customers spend on average 4% more each time they are in a landline store and 10% more when shopping online².

“73% of digital business professionals surveyed tell us that their firms are investing in ‘buy online, pick-up in-store’.”³

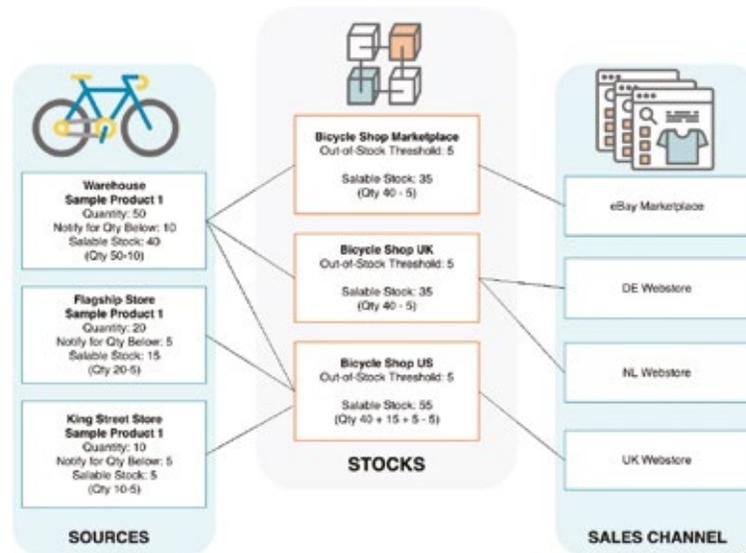


Fig. 4: Schematic diagram of order processing with MSI Magento
 Source: https://docs.magento.com/m2/ce/user_guide/catalog/inventory-about-sources-stocks.html

MAGENTO ORDER MANAGEMENT

Magento Commerce offers additional, advanced features that affect more effective sales in multiple channels. These are the following:

- **"Ship from store" capability for more efficient use of inventory.** Customers who buy online will be able to receive the product from a local store instead of a central warehouse, which significantly speeds up delivery.
- **The possibility of "ship to store!"** The ship to store (STS) function allows to optimize inventory in different sources for in-store picking orders. It also makes it possible to reduce costs through local shipping and reduce margins on unsold goods.
- **The ability to "save to sell" with access for co-workers to all inventory.** Consumers have more options to purchase items from a global warehouse. Retailers of individual stores can "record sales" that they have lost due to lack of inventory in the store, and the customer never leaves empty-handed.
- **Expansion on a global scale with multiple brands, using different devices.** Manage your global business with a central procurement system across all channels, including multiple brands or websites, markets, mobile devices, kiosks, stores and call centres.
- **Build a single source for orders, inventory and order fulfilment.**
- **Support more advanced order scenarios such as pre-sales, return orders, etc.** Orders can be realized for a group or an individual customer. Orders can be retrieved in advance of product release and "no inventory" with an estimated shipping date.

- **Enable customer service to have a better overview of your orders.** Order information from all sales channels is centrally available and inventory information is updated in real time. This allows customer service to easily help customers process orders, exchange or return products.

TECHNOLOGY HELP RETAILERS MEET CONSUMERS' EXPECTATIONS

With Magento there is a possibility to fully integrate functions that enable your customers to search for products online and take advantage of various pick-up and delivery options. However, Magento is much more than just providing customers with the best experience in the online store. The system gives an insight into inventory details across all channels, providing full control and access to inventory anywhere. By giving customers opportunity to omnichannel purchases, sellers can have a huge influence on conversion rates in their stores. ●

By Magezine Team

Sources:

1. B. Witcher with F. Swerdlow, C. Tajima, S. Wolken, Mastering The Art Of Omnichannel Retailing, <https://www.forrester.com/report/Mastering+The+Art+Of+Omnichannel+Retailing/-/E-RES129320>
2. Here's why omnichannel consumers are more valuable, <https://www.businessinsider.com/heres-why-omnichannel-consumers-are-more-valuable-2017-1?IR=T>
3. H. Williams, Defining Your Omnichannel Approach – Q&A with Magento Commerce, <https://www.dtcdaily.com/2018/05/defining-your-omnichannel-approach/>
4. I. Minialio, Magento Multi-Source Inventory (MSI), <https://www.slideshare.net/maghamed/magento-multisource-inventory-msi>

HOW TOUS BOOSTED SALES WITH MULTI-SOURCE INVENTORY (MSI)

With 54 stores across Europe, the Polish fashion retailer needed a robust order management solution.

+13.60% Black Friday Conversion

+7.98% Black Friday AOV

+29.74% Overall Black Friday Revenue

A SEAMLESS CUSTOMER EXPERIENCE

Founded in 1920, the luxury lifestyle brand TOUS is famous around the world for high-quality jewelry, handbags, and accessories. The company's Polish operation boasts 54 brick-and-mortar stores across Poland, Czech Republic, and Slovakia. The brand's eCommerce journey in Poland had begun on Magento Open Source in 2015, but as TOUS approached their 100th anniversary,

a major upgrade was needed. They offered over 6,500 SKUs but were struggling with product availability. If a shopper purchased a pair of gold earrings online, from where would they be dispatched? TOUS Poland had a dedicated warehouse for eCommerce but preferred to keep most of their stock on display in stores. Sometimes, during crazy holiday sales, they had to transfer stock from their shops to the eCommerce warehouse, just to keep up with demand.



With 30 percent of their business now conducted on mobile, TOUS Poland wanted to create a seamless offline and online customer experience. They needed to increase their online conversion rate, improve speed, scalability, and performance of their platform. But what they really needed was to deliver true multi-source order management, with real-time updated local inventories. In other words, they needed to know how many gold earrings they have for sale, exactly where they are, and how best to get them to the customer. In 2018, TOUS decided to upgrade to Magento Commerce 2.3, to take advantage of the platform's Multi-Source Inventory (MSI) capabilities. But could they pull off an upgrade with just three months before Black Friday?

IMPROVED OPERATIONAL EFFICIENCY

TOUS hired Strix, the Magento Enterprise Solution Partner, as they were one of the Magento community's core contributors to the Multi-Source Inventory module. The project was urgent, they explained. "The fourth quarter is a very important time in the jewelry business because people choose gifts for their families," said Magda Staszczuk, Head of eCommerce for TOUS Poland. "For TOUS to be on time with a scalable and stable new platform was crucial."

For some brands who start their eCommerce journey on Magento Open Source, it can be hard to justify the upgrade to the enterprise-level platform, Magento Commerce. But TOUS understood that the next 100 years of their business would be dominated by eCommerce. "Once they reached a certain level of complexity and orders, they were ready to migrate to Magento Commerce 2.3," said Bartłomiej Szymański, Head of Software Development at Strix. "This is an important consideration for all merchants on Open Source, who are probably facing many challenges with performance and stability." As one of the first merchants to use 2.3, TOUS reaped the benefits of MSI.

The use of MSI in the project turned out to be a very good decision.

Bartłomiej Szymański

Head of Software Development,
core-MSI contributor at Strix



MSI integrated easily with existing TOUS systems (e.g. ERP) and helped to improve operational efficiency by managing inventories across multiple physical locations, all within the Magento admin. TOUS can now manage stock in various locations, from their warehouses to shop windows, while handling stock creation and aggregation. Meanwhile, low stock notifications reduce out-of-stock situations, and most importantly, TOUS can offer more products for sale online, whether the actual items are in a warehouse, or in a store.

MSI SOLVED ALL THE BRAND'S PROBLEMS WITH STOCK

The new TOUS platform was launched with time to spare, just a few weeks before Black Friday. The new platform and MSI solved all the brand's problems with stock. "Once the product is sold, MSI can ship the product from all places," said Szymański. "It's quite advanced, we have three different countries

with three different shipping methods and locations, and 56 sources of info for the warehouse. It's proof that MSI can solve complex problems." As soon as TOUS was able to show more stock to online customers, sales naturally increased.

Black Friday was the platform's first true challenge. It was a very exciting evening for everyone involved in the upgrade. The night before the TOUS sale started, at 11:45 pm, nearly 20,000 shoppers filling their shopping carts and waiting for prices to drop before they checked out. "The site really performed," said Szymański. As the Black Friday madness began, the conversion rate rose to 13.60 percent higher than in 2017. Transactions grew by 20.15 percent, and Average Order Value (AOV) increased by 7.98 percent. In total the solution tripled revenue to 29.74 percent higher than the previous year's Black Friday, proving that omnichannel is the key to growth. ●

By Magento (Adobe)

PRODUCTS

Magento Commerce

SOLUTIONS

Omnichannel Solution

PARTNERS

Strix

STOP PRESS!

DC Thomson Launches on Magento Commerce 2

When DC Thomson, one of the oldest media creators in the UK, upgraded to Magento, they reduced 84 extensions to just 18, introduced personalization, and increased their Average Order Value by 23%.



THE 113-YEAR-OLD PUBLISHER NEEDED A 21ST CENTURY PLATFORM.

DC Thomson Media is one of the leading media creators in the UK, and part of the DC Thomson Group. The company is headquartered in Dundee, Scotland, with a London base in Fleet Street. DC Thomson Media publishes newspapers, magazines, and comics, and boasts an extensive eCommerce and events portfolio, alongside interests in gaming and radio.

SELL MAGAZINES ONLINE

To sell their magazine subscriptions, gifts, collectables, and puzzles direct-to-consumer online, DC Thomson Media launched the DC Thomson Shop, using the popular Magento Commerce 1 Open Source platform. But in 2017 the

company realized they needed a more up-to-date and robust solution to accelerate their growth. DC Thomson Media had heavily customised their online store and it was suffering from sluggish load times and a dated user experience. There was also little opportunity for cross-promotional marketing or personalization. For DC Thomson Media to achieve their ambition of doubling their revenue in the next financial year, they needed a fast, stable, and secure platform.

The Digital Team at DC Thomson Media performed a careful analysis of five top eCommerce platforms, including Magento Commerce 2, comparing both on Cloud and on-premise solutions. Their KPIs were to improve workflows for adding new products, enjoy better analytics, server-side performance analysis, and a well-supported range of extensions. Alan

Melrose, Digital Commercial Operations Manager for DC Thomson Media, looked at the results: "The Cloud edition won by a clear margin," he recalled.

This project has been hugely successful and delivered immediate and tangible results. We're looking forward to more development and greater sales as we improve our skills and knowledge in Magento Commerce.

Kirsten Morrison

Head of Digital, DC Thomson Media

A CUSTOMIZED, PERSONALIZED USER EXPERIENCE.

"It was critical that the platform we selected could provide the core functionality without the need for heavy customisation," said Melrose. "This had been a constant drain on resources over the previous four years."

The main attraction of Magento Commerce 2, from a UX perspective, was its out-of-the-box capabilities: "There was very little front end customization added and additional functionality was largely done using extensions," he said.

The next steps were to add personalization to the store, and cross-sell/up-sell capabilities at checkout. Now customers who drop "My Weekly Gardening Calendar 2019" into their shopping basket, might be offered "Pass It On Cooking Tips From The 1950s" before they checkout. Customers can even upload a photograph of a loved one, and comic book artists will add them to a personalized print or canvas, next to DC Thomson Media's most beloved cartoon characters.

"We've added around 18 extensions to improve administrative functions – for comparison we had 84 on our Magento Commerce 1 instance," said Melrose. "On the customer side

we added Gigya and Sailthru extensions to power our single customer view and deliver marketing automation. This allows the store to share and consolidate customer profiles and data points across a collection of 26 websites."

This solution now powers personalized marketing based on propensity modelling and cross-platform browsing, turning the store into an intelligent marketing machine. Finally, DC Thomson Media built an application to collect direct debit orders from customers' bank accounts, and configured the REST API to provide an endpoint for their internal data reporting warehouse, and fulfilment partners.

THE STORE IS NOW AN INTELLIGENT MARKETING MACHINE.

The end-to-end project took nine months in total, but the design/build/deploy phase took just six months. DC Thomson Media are thrilled with the results: "The impact in the short term has been immediate, with all site metrics improving, including server response time, page load speed, page views, conversion rate and pages per session," said Melrose. "Our increase in revenue has already outstripped our initial investment." In the new build, DC Thomson Media's improved user experience allows customers to breeze through to checkout, and use different payment methods for multi basket items. But most importantly, the move to Magento Commerce 2 improved their workflow, leaving the team to concentrate not on processing products, but building brands to last another century. ●

Magento Commerce 2 is an incredibly powerful platform so there has been little customization required. We witnessed an immediate improvement.

Kirsten Morrison

Head of Digital, DC Thomson Media

By Magento (Adobe)

INSIDE PAUL SMITH'S MAGENTO UPGRADE

Commerce without Compromise

Paul Smith is Britain's foremost designer. He is renowned for his creative aesthetic, which combines tradition and modernity. Reaffirming the values that Paul set down in 1970, "classic with a twist" remains the company's guiding principle. Happily positioned between high fashion and formalwear, Paul Smith takes reference from both, and is proud to stand apart. Everything Paul Smith creates is underpinned by a dry, British sense of humor — but his greatest attribute, according to Vogue, is his simplicity. In pursuit of a simple, streamlined eCommerce experience, in October of 2017 Paul Smith decided to upgrade to Magento Commerce 2.



Paul Smith's high-end shoppers were demanding an omnichannel experience — and customizable products.

Paul Smith began their eCommerce journey in 2004, by developing a bespoke web store with the help of an outside agency. But since 2012, Paul Smith's team had managed eCommerce development in-house on Magento Commerce. However the business was changing fast: 50 percent of their traffic was coming from users on mobile devices. The Paul Smith brand was also expanding fast into Asia, and their high-end shoppers were demanding an omnichannel experience — and customizable products.

A DIGITAL TRANSFORMATION

With digital now one of their largest sales channels, optimizing the Paul Smith online experience would have a direct impact on the company's success. The goal for their upgrade was to future-proof the site, deliver a more sophisticated mobile experience, add global capabilities, and omnichannel shopping and fulfillment, while simplifying their eCommerce operations — the Paul Smith way. The clean code and streamlined performance of Magento Commerce 2 was just what they needed.

ON-THE-GO COMMERCE

Magento Commerce 2 helped Paul Smith focus on improving their mobile experience, with clearer listing pages, accessible links, and streamlined menus. The team implemented the default Magento checkout to ensure existing third-party extensions worked seamlessly, and used PayPal Express Checkout to increase conversions. They also enjoyed minimized cart abandonment on mobile and desktop alike.

A CUSTOMIZED EXPERIENCE

Paul Smith used the new site to launch their customizable products. Shoppers can now select a wallet, scarf, or purse, and add their initials, or one of Sir Paul's famous "doodles". A monogram preview feature allows users to visualize how the design will look in real life. The Paul Smith team also developed their own content editor (using React JS) to easily build responsive pages, and to support new products and brand collaborations. Meanwhile, in the "Stories" section, Paul Smith content comes to life with widescreen video, vivid imagery, and copy that's as eye-catching as their fashion.

FULFILMENT MADE EASY

Fulfilment is now simpler than ever: If stores can't fulfil an order directly, Paul Smith can fulfill using stock inventory that flows through Magento. Then, customers can pick-up in-store or have the product shipped directly to their home. They can even buy their monogrammed purse from a sales assistant and pick it up later, or have it delivered.

This omnichannel process can be extended to any global store in the Paul Smith network, and set-up is easy too. It's all done on the Magento Admin Panel

with zero help from the web development team. All orders are pushed to PRIMA ERP using Api, and are processed by the dispatch team at a central warehouse. By integrating with multiple couriers, Paul Smith offers the choice of Free/Express/Scheduled, and Click-and-Collect delivery options.

*Six countries,
two languages,
and multiple
currencies..."*

The upgrade to Magento Commerce 2 achieved each of Paul Smith's unique goals, allowing the brand to respond faster to changes, deliver a better customer experience in-store and online, and support their global expansion. Performance significantly improved with faster global page-load times, thanks in part to the platform's built-in Varnish page-caching. With the launch of shoppable Instagram, Paul Smith also saw a five percent improvement in mobile conversion.

The new Paul Smith site supports six countries, two languages, and multiple currencies. Key to the site's efficiency is the default site structure which reduces the volume of content, while allowing each country to manage content locally—a truly global operation. Using Adyen payments worldwide, Paul Smith can offer more options, managing both online and in-store purchases seamlessly.

In the six months after their upgrade, Paul Smith has seen revenue increase by 15.5 percent. Their focus on the mobile customer experience paid off with an increase of 8.6 percent in mobile users, and an increase of five percent in mobile conversions. For Paul Smith, simplicity really was the answer. And with Magento Commerce 2, they're ready for whatever their customers demand next. ●

By Magento (Adobe)



The right tools for the right job in Magento 2





The difference in complexity between Magento 2 and Magento 1 was quite clear from the very beginning: Magento 2 introduces a huge set of new tools, patterns and techniques allowing an incredible software flexibility, but it comes with a price in the scope of required skills, developing and debugging time. This new complexity should be faced using the right tools for the right job, tools that can save us a considerable amount of time and protect ourselves from time consuming errors and unforeseen difficulties.

DEVELOP VS DEBUG

Before diving into a list of software solutions, we should first understand how we spend our time while doing our job.

A very superficial – but effective – analysis of a developer's daily job highlights that we spend our time in three main activities:

- Developing a new feature.
- Debugging an existing feature.
- Going through zillions of cute kitten pics in the main social networks.

Despite the fact that the last activity is universally recognized as the most important one, we will only focus on the first two and try to select the best tool for each of them.

DEVELOPING A NEW FEATURE

While developing a new feature in Magento 2, we will face several challenges, but most of them are related to boilerplate code, XML file syntax and coding standards. All of these aspects are not complex at all, but they require a mnemonic approach to the code and we all know that most of the time a developer cannot even remember what he/she ate for breakfast.

We can be quite sure that most of the developers – or at least the author of this article – cannot even write a simple di.xml file from scratch without copying and pasting from a previously implemented module and none of us is probably able to write a whole UI component represented by hundreds of lines of XML code.

This is why we should facilitate our job by using a correctly configured IDE with some code generator and code facilitators, but first of all we need to setup a performing developing environment.

BOX 1. THE DEVELOPING ENVIRONMENT:

Docker vs Vagrant

At the time of Magento 1, most of the developers were using a virtual machine approach to setup a developing environment. Vagrant was the most obvious option at that time.

Vagrant could be a working solution for Magento 2 as well, but since this latter is more demanding than its predecessor in terms of CPU and RAM, a virtual machine is not exactly the best option you may adopt. A more effective solution is to use docker as virtual environment. As you may know,

docker does not rely on virtual machines, it relies on an isolated running context. In other words: while Vagrant was using a virtual hardware emulation with an approximate computational power loss of 20-30%, docker directly uses your machine resources with – virtually – no computational power loss.

Setting up a docker environment can be tricky at the beginning, especially because you are supposed to setup a container for each single service, but

you can find some good hints from community contributions and tutorials. So, if you do not know exactly where to start from, a nice and very extensive video tutorial from eCommerceAholiC can be found here: <https://youtu.be/2VLSaceaDnM>.



IDE

Since it is not cool anymore to say that your website was developed using notepad (see websites of the late '90s), using a modern and professional IDE is mandatory for every project.

We have many different options while dealing with Magento 2, such as: PHPStorm, Visual Studio Code, Atom and others. They are all valid IDE with very good dedicated Magento 2 plugins, but in the personal

opinion of the author of this article, PHPStorm is probably the best IDE you may adopt with Magento 2, especially because of the wide set of tools you may want to integrate it with.

DEBUGGING AN EXISTING FEATURE

As anticipated, the second aspect of our daily job is debugging existing features. This is probably the most complex aspect of our job. It requires a full understanding of the code before changing it and the code we are working on can actually be written by someone else.

The other complex aspect of debugging a Magento module or feature is represented by all of the customization structures that Magento 2 provides. With the old Magento 1, it was quite complex to find a piece of code responsible for a certain feature; now, with Magento 2 this is nearly similar to a coding nightmare.

We have several structures like plugins, preferences and observers that allow customization over the standard Magento behaviour. Despite the fact that these are very powerful tools, they may lead to a code dispersion with the side effect of not allowing the developer to quickly identify the source of a problem to fix.

When trying to identify the source of a problem we have to focus at least on one of the following three questions:

- **What is the problem? Can I reproduce it?**
- **When the problem was introduced?**
- **Where is the code responsible for the problem?**

Most of the time, having a quick answer is almost impossible, but if we use the correct approach and the right tools, we can simplify our job dramatically.

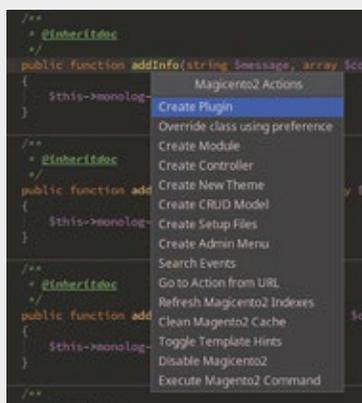
ANSWERING TO: WHAT IS THE PROBLEM?

This question is typically undervalued, especially by customers, as most of the bug reports come with a simple "It is not working anymore."

Understanding and reproducing the issue is the first step to fix a problem. This is why every bug report should be detailed, qualified and verified as much as possible. Every developer should spend some time trying to reproduce the issue on a local developing environment.

BOX 2. A POSSIBLE PHPSTORM SETUP

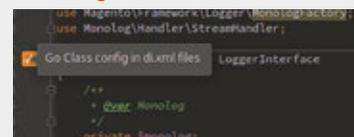
Magento



Magento is a commercial plugin from Enrique Piatti including a bunch of code generators and facilitators. Just as an example, Magento allows you to quickly generate a new module or a Magento plugin in a few steps.

Other than generating code, Magento also provides some tools for code assistance like a di.xml parser allowing the developer to quickly see if a method or class was involved in a preference or a plugin. Similar facilitators

are even available for the most common developing aspects of Magento 2 (e.g.: layout files, observers, etc.) www.magiciento.com



Official Magento2 PHPStorm plugin

This is a less powerful and complete, but free alternative to Magiciento. This tool appears to be officially supported and created by Magento and mostly provides code hinting and facilitators.

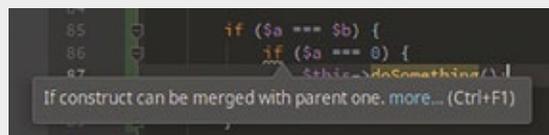
<https://github.com/magento/magento2-phpstorm-plugin>



PHP Inspections (EA Extended)

PHP Extended Inspections is a must have plugin for every good PHP developer. It allows deeper code style inspections, including some aspects that usually are considered secondary, like simplifiable logical conditions, redundant code, opcode optimizations and so on.

<https://github.com/kalessil/phpinspeceptionsea>



Configure code sniffers

<https://www.jetbrains.com/help/phpstorm/php-code-quality-tools.html>

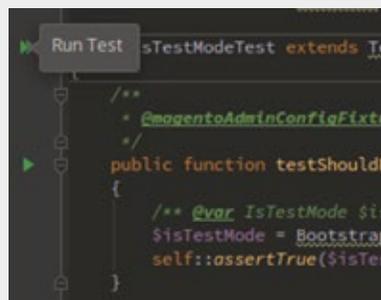
A good code style quality is the starting point for a correctly working software. PhpStorm can be integrated both with PHPCS and PHPMD. PHPCS is a code style checking tool based on coding standards (like PSR-2), while PHPMD is a "mess detector" capable of detecting overcomplicated software structures or code smells. For both of them Magento 2 provides a set of rules that can be found in "dev/tests/static/testsuite/Magento/Test/Php/_files" and can be referenced by PhpStorm. Similar testing tools are also available for JavaScript or CSS code styling.



PHPUnit configuration

https://devdocs.magento.com/guides/v2.3/test/unit/unit_test_execution_phpstorm.html

PHPUnit can be integrated with PhpStorm to run tests directly from the IDE interface. This allows us to quickly run a test without switching to another environment or running complex command line tools.



BOX 3. CODE GENERATORS:

Magento 2 is a complex software and it requires some boilerplate code and complex XML configuration files. Writing them is very time consuming and can expose any developer to potential syntax errors. Here is where code generators can help us.

Magento 2 code generator tool by Juan Alonso

<https://github.com/staempfli/magento2-code-generator>

Magento 2 code generator tool by Juan Alonso is a powerful set of command line tools capable of quickly creating the most time-consuming coding structures in Magento 2.

It can be installed through composer in a minute and provides a set of pre-configured operations that can even be easily extended by anyone.

Just as an example, a developer can create a new Magento 2 module by simply typing: "bin/magento t: g module"



Magento 2 PHPStorm Live Templates by Juan Alonso

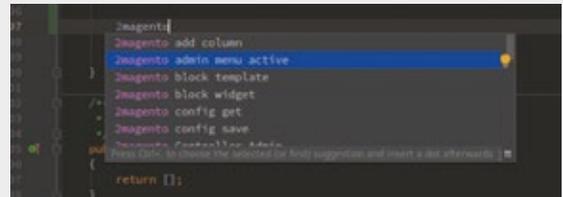
<https://github.com/staempfli/magento2-phpstorm-templates>

Magento 2 Live templates from Juan Alonso is a set of PHPStorm templates for the most common classes and software coding structures. It works as a set of shortcuts a developer can recall directly while typing the code. Just as an example, directly typing "2magento config xml", creates a default config.xml file.

A whole list of commands can be found here:

<https://github.com/staempfli/magento2-phpstorm-templates/blob/master/docs/liveTemplates.md>

As every tool written by Juan Alonso, the most amazing feature, is how the auto generated code fits at its best the required coding standards.



Templates



List of commands

Mage2Gen

<https://mage2gen.com/>

With Mage2Gen, none of developers will go nuclear. This software is capable of creating a whole module structure by simply typing its main aspects in an online tool.

At the time of this article writing, Mage2Gen still requires some minor adjustments to be fully compliant with all the new Magento 2 coding standards (like strict typing), but it still represents one of the most complete and powerful code generating software for Magento 2.



But it is known that it is virtually impossible to actually find time to do it, and a problem may only happen in production. In cases as these we have several tools that can help us.

Other than the most famous NewRelic (<https://newrelic.com/>) and BlackFire (<https://blackfire.io/>), mostly oriented to performance analysis, another valuable tool for a Magento developer is "Sentry.IO".

Sentry.IO

<https://sentry.io/welcome/>

Sentry.IO is a commercial service – also provided with a free developer license – capable of tracking PHP errors without waiting for a report from one of your customers or users.

Several integration modules are already provided by the great Magento community:

- <https://github.com/justbetter/magento2-sentry>
- <https://github.com/SebWite/Magento2-Sentry-Logger>

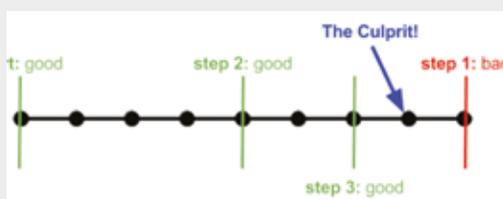


BOX 5. ANSWERING TO: WHEN THE PROBLEM WAS INTRODUCED?

This is not a secondary question at all, as it can be faced using one of the most amazing features provided by git. Very few developers know about the "git bisect" feature.

Working with "git bisect":

<https://git-scm.com/docs/git-bisect>



Git bisect allows a developer to identify the exact commit where a problem was introduced. We only need to know the last known working commit and the first known breaking commit.

We can start a git bisect session by simply typing "git bisect start". At this point, git will allow us to time travel the commit timeline specifying if the software is still working or not.

We can, just as an example type "git bisect good" or "git bisect bad" to point out a working or not working commit.

Every time we will qualify the commit as "good" or "bad", git will change our position in the timeline pointing to another commit and asking the same question.

After a certain amount of time travels, git will identify the breaking commit reducing our amount of code to analyse.

Using git bisect is subject to some restrictions and best practices as it relies on a good usage of git. If we have commits made by hundreds of files and miscellaneous modifications, git bisect will be almost useless because we will not be able to restrict our code base in a decent way. Another git bisect limitation is represented by the Magento database management as it relies on filesystem to determine the database structure. Time traveling the commit timeline can bring us to a previous version of our software configuration not more compatible with our database structure.



Riccardo Tempesta

Riccardo grew up on bread and computer science (his PC was a gift at the age of 6). Over time he became a developer with an exceptionally broad knowledge of the most disparate areas of the field. He has been with the Skeeller srl (former Idealia Group srl) since 2001 as a founder and technical director. The mainstay of the technical department for MageSpecialist, he is a senior analyst and software architect, Magento Specialist, senior PHP developer, and technical manager.

Riccardo started working with Magento solutions since early 2008 with the old 1.x version. After falling in love with this new technology he became Magento Certified developer and Certified developer focusing on developing and finding new Magento solutions. In 2015 he started working with Magento 2 and contributing with Magento Community. In 2018 became one of the most active contributors in key projects like MSI, a community maintainer and a Magento Master Mover in 2019.

BOX 6. ANSWERING TO: WHERE IS THE CODE RESPONSIBLE FOR THE PROBLEM?

As anticipated, finding the code responsible for a problem can be extremely tricky in Magento 2, but – luckily – we have several valid tools to face this aspect.

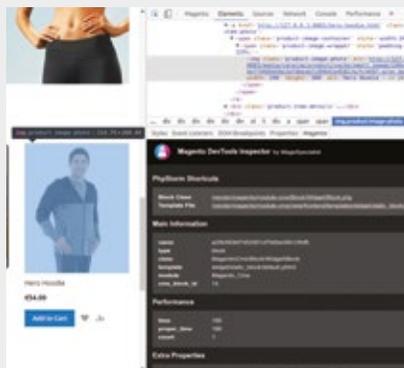
MageSpecialist DevTools:

<https://github.com/magespecialist/mage-chrome-toolbar>

This is one of the must have software solutions for any Magento developer as it provides a chrome plugin to deeply analyse all the hidden aspects running under the hood of Magento. It provides information both for a frontend developer and a backend developer and it is also capable of running basic performance analysis.



MageSpecialist DevTools – Inspector:



MageSpecialist DevTools is integrated with chrome inspector and can provide template and block information related to a specific page element.

This feature is extremely useful for frontend developers to quickly understand what the template file or block related to a specific part of the page they are working on is.

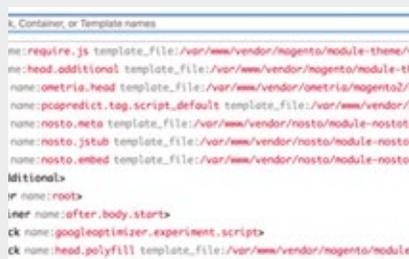
A PHPStore integration is provided to directly open the PHTML file responsible for the piece of page we are analysing.

MageSpecialist DevTools – General Tab:



MageSpecialist DevTools can also provide general information about the running session highlighting involved blocks, design modifications, plugins, observers and even MySQL queries involved in the current page.

JH M2 DevTools by Shane Osbourne:



This is another priceless tool mostly oriented for frontend developers capable of quickly analysing the layout structure of the page displayed in our current browser session.

It also provides a convenient search system to identify a layout element and its information.

THE POWER OF MAGENTO COMMUNITY

When I first encountered Magento it was 2008. For many merchants the new open source platform provided a shopping experience and backend tools that was years ahead of the competition. Many businesses – both agencies and merchants – wouldn't even have survived or started in the first place if it wasn't for the competitive edge Magento provided them.

Not only did many businesses start using the product, but many also contributed back to the ecosystem (in many different ways, definitely not only in terms of code), resulting in an ever-expanding, amazingly responsive community bringing the ecosystem to what it is today. As I said before: I fell in love with the product and stayed for the people. As did many others.

To single out one of these recurring community contributions, let's have a look at the broad range of events our community has developed. The first Meet Magento event ever was born on January 27th 2009 in Utrecht. This year we hosted our 11th annual Meet Magento in The Netherlands and in the meantime the events series took off worldwide: from a handful of events in Europe it expanded to over 30 yearly events worldwide. And it didn't stop at Meet Magento events: we have unconferences, developer paradises, MageTitans and loads and loads of local meetups. If you want to learn and/or want to meet people, there's an event for you. And the most amazing thing is: all of these events are organized by local community partners without the need for Magento (the

company) to be involved in the organization. An impressive achievement unlike any other tech community in the world.

(If you want to start a Magento event: definitely reach out to the Magento Association and connect with other organizers in the Meetup organizer subforum on community. magento.com. Lots of resources and good advice can be found there!)

We (the Magento community) have a rich history in building new collaborative programs and successful events across geographies. But – as always – consumers change the way they shop, technologies evolve/emerge, and the market is changing rapidly. Even more so after the Adobe acquisition.

With the new Magento Association, we have a new body in place that is dedicated to fostering and supporting innovative technology products, coordinating existing and exploring new community events, training & education and expanding collaboration between all parts of our ecosystem. A central body that, with your support, can bring more stability and growth to the overall ecosystem.

The vision of the association (which I think reflects the wishes of many in the community) is to have an open, healthy and powerful Magento ecosystem for the continued success of future generations of our businesses. That's what we'll be striving for. We all have a very exciting time ahead of us! ●



Guido Jansen

Guido is a psychologists and usability expert focused on E-commerce. He helps companies to build international customer journey optimization teams and to build a culture around experimentation and validation. Since 2008 his work included many Magento projects, he is a Magento Master and is part of the Board of Directors for the international Magento Association. He is often invited to speak at international events about persuasion, e-commerce, conversion optimization and experimentation cultures.

A long way

from Monolith to Service Isolated Architecture

The goal of this article is to share some ideas and thoughts regarding how Magento is going to look like in several years and how it is reacting to the market's changes and its demands. It's going to be focused on Service Isolation – probably one of today's biggest buzz words in Magento – maybe not as big as Progressive Web Apps (PWA) but still. It is worth to remember that actually PWA is just one of the bricks in the wall which is called Service Isolation.

Service Isolation isn't just an abstract product of a few wise man's imagination. This is the project which Magento has already committed to do – there are approved and confirmed plans that lead to start the work. It's not going to be about the API deprecation. All the released APIs are not going to be affected and supposed to be forward compatible. This is why it's so important for 3rd party developers while extending Magento nowadays to rely on Service Layer APIs. This will prevent potential problems in future while upgrading to new Magento releases.

EVOLUTION, NOT REVOLUTION

Magento is not going to bring any revolution changes to the platform from 3rd party developers standpoint, despite the fact that many changes to Magento architecture going to be applied. But along with that existing extensions and customizations should not be affected with the changes much. It's all about evolutionary way in creating smarter solutions. Moreover a single box Magento installation mode would be preserved for those merchants who no need distributed deployment.

However, there are some challenges that Magento has to deal with when developing Service Isolation:

- **The application grows in size.** Magento 2 is more sophisticated platform than Magento 1. That comes with more complexity than it was before. It is not because of Magento's architects and developers, but market's requirements that appear on day-to-day basis. Magento is now trying to fit for bigger merchants. Those who remember Magento from ten years ago remember that it was considered the system for medium-size merchants with couple of hundreds SKUs. Right now, Magento is targeting new market – and especially after acquisition by Adobe – the platform is aiming to fit bigger requirements, many thousands of SKU's and many thousand transactions per day. So the system is supposed to be scalable for both medium-size and high-level merchants.
- **Merchant customization grows in complexity.** To make the particular customization for Magento, the development complexity of agencies dealing with it is also raising. This also leads to the issue where Magento releases are not very frequent – around once per year and few yearly patch release. Magento developers spend approximately three months for each release preparation.
- **Changes on the market.** Frequent releases are pretty much obligatory because the e-commerce market requirements constantly change and the merchants are supposed to quickly react to them. And if for some

reasons, like a lot of complexity in the customization of Magento or particular extension, merchants can't afford to proceed with those upgrades, their business would be impacted. Our business goal is to make as many releases as possible and drop all new changes to the market as soon as we can.

THE ANSWER FOR THESE PROBLEMS IS SERVICE ISOLATION

There are some differences between Service Isolation and microservices. When reading literature and textbooks about microservices, they used to be pretty granular, atomic and their responsibility is just for one particular job. In Magento way the isolated service is going to provide quite a lot of functionalities – it's not going to be service only for one purpose e.g. converting images and that's all. In Magento we will have services for inventory, isolated service for catalog, service for checkout, order management, shipping, payments and so on, covering all major components of e-Commerce domain – big services each one providing own set of APIs considered as independently deployable and scalable applications.

All the problems started with Magento 1 and leaked to Magento 2 (as it wasn't written from the scratch but inherited all the code). At some point developers just branched from Magento 1 and started to refactor the code over many iterations.

One of the problems was the dependencies between modules, classes and components. In Magento 2 there are a lot of those dependencies – not so evident in Magento 1 because of lack of dependency injection – classes were created just by request which resulted in lazy loading.

Dependency injection in Magento 2 really unveils the problem with too many interlinked dependencies.

There was no static check in Magento 1 which prevented developers from introducing incorrect dependency both on the class level and on a module level, which spoiled quite a lot of modulatory concept.

First idea and the first main problem that needed to be fixed when starting working with Magento 2 were introducing Single Responsibility Principle on the level of classes, models, modules and particular components. Each specific thing should be responsible for one and only one business operation.

It started with using it on the class level, but at some point the major refactoring of the level of process was made resulting in switching from this more abstract level to the level of component.

This is how the idea of modularity appeared and this was the first step to the microservice architecture and isolated services architecture.

For those who are familiar with a Domain Driven Design (DDD), the idea of Bounded Context is well-known. It fits to very large business domains as finance and e-Commerce and helps to write software for pretty sophisticated scenarios which have to be programmed by developers who are not really familiar with all the nuances of business. Thus, the main idea of the Domain Driven Design is to build a bridge and fill the gap between the business and development team by building up a common, rigorous language between developers and business representatives – Ubiquitous Language. Along with that dividing the system on highly cohesive and loosely coupled components which supposed to represent particular areas in the scope of the business domain, not just program modules which make sense only for developer. These components are called bounded contexts in terms of DDD.

The way how bounded contexts could be represented in Magento – combining already existing modules which could be distributed as independent applications, isolated services. So there are a catalog, customer, checkout, order management, shipping, inventory, payments, and one can consider isolated services with a bunch of modules pretty highly coupled and having a good cohesion inside each of those component, so the actual cohesion representing the main idea of this module is going to be bundled. For example, the Catalog isolated service consists not only of Catalog module, but along with that from CatalogSearch, all modules representing different types of products, Swatches, ProductVideo, Catalog Import/Export functionality, and even more some of the module which are distributed as a part of Magento Commerce edition and B2B – CatalogStaging, GiftCard and SharedCatalog.

The main idea behind microservices and isolated services is to have an ability to distribute the services on different environments and to provide the capability to scale them independently.

(slajd 15) When talking about scalability and the distributed computing there's an idea of the CAP theorem – which states that is impossible for a distributed data store to simultaneously provide more than two out of three guarantees:

1. **Consistency** – every read receives the most recent write or an error.
3. **Availability** – every request receives (non-error) response – without the guarantee that it contains the most recent write.
5. **Partition tolerance** – the ability of a data processing system to continue processing data even if a network partition causes communication errors between subsystems.

That's why there were typically two kinds of systems – the ACID and the BASE and most of the relational database systems such as MySQL are ACID compatible.

ACID stands for atomicity, consistency, isolation and the durability. MySQL is a perfect fit to the system which complies with both consistency and availability requirements applying MySQL transactional logic which symbolizes a unit of work performed within MySQL engine.

But MySQL isn't the best solution for partition tolerance, and tough goal is to make the scalability for the system, especially when talking about the write scalability (because read scalability is more or less achievable with replication). Partition tolerance can't be neglected because it is more or less a standard of the market nowadays, especially in such domain as eCommerce where merchant's business could start growing drastically quite unexpectedly or expectedly after the advertising campaign or seasonally like Christmas sales period, and there is no time for re-platforming in this case. It's important for a merchant to have a possibility to react quickly on market requirements and expand the system or particular sub-systems under load by demand. That's why this CAP theorem cannot be considered as up-to-date anymore – because if the partition tolerance must be supported, it's time to choose between the consistency and availability.

However, the availability cannot be neglected neither. Nowadays websites should be available 24/7 and this is what merchants required from agencies and developers.

There is a bunch of systems supposed to provide support to this metric. This system is called BASE, and stands for basic availability, soft state and the eventual consistency.

For example, with DB transactions involved a business transaction can affect several different DB servers relying on the two-phase commit (2PC) protocol, with one server node orchestrating all the local transaction commits and after all changes have been properly applied it confirms the successful operation as well as providing information which server was responsible for the transaction. But in reality the application of 2PC is very fragile and error prone because there is too many communication over the network involved in each such transaction. Implementation at the source code level may be problematic either, because as soon as an application becomes distributed application, some of the problems that have arisen at the network level need to be solved – such as atomicity, high latency on the network, network security, low bandwidth and the problem that some particular operation and commands would be delivered twice.

That's why much more efforts need to be introduced developing an application which supposed to be run on distributed basis. There are particular patterns which should be applied on infrastructure level for developing such system, some of those are Circuit Breaker where the main idea is to wrap a call to the remote function in a circuit breaker object, which monitors for failures. Once the failures reach a certain threshold, the circuit breaker trips, and all further calls to the circuit breaker return with an error, without the protected call being made at all; and Bulkhead (a term borrowed from cargo ships, where the bulkhead is a wall built between different cargo sections, which makes sure that a fire or flood in one section is restricted to that section and other sections are not impacted) is the pattern which guarantees failure in one service or a group of services not to bring down the whole application.

For example, when placing an order on Amazon, it takes a very short time for order placement process. There are no exclusive locking operations which block some crucial sub-systems and make customers wait, just a specific number associated with order id is assigned to newly created order object and all things like payment, shipping and all further notifications happen individually and asynchronously, so the client is simply getting some notifications regarding the status of the order that will be charged after the initial order is placed.

Even the inventory deduction can happen way after the order had been placed – for example Amazon can determine that they have not enough products in stock to complete the order so that they will send the client notification: sorry we don't

have an all products to fulfill your order so we're cancelling it and making a refund.

This shows how the big complex sophisticated systems solve that problem and they totally switch from the strict consistency in favor of eventual consistency with numerous of compensational transactions taking into place if something goes wrong on a way to successful order fulfillment.

One of the patterns that helps with resolving this issue is Saga – the approach when there are many short actions, and the execution of one of them usually leads to the start of another one – as in the case of Amazon – after placing an order the next action (e.g. payment) begins, and after it is finished the next step, like shipment is initiated.

For example, at the inventory deduction step, the system informs that there are not enough products in stock to complete the order – so a compensation transaction for both payments may be proceed, and return the money to the customer from whom system have already charged, and proceed to return all other business transactions and activities already done.

For implementing sagas the Event Sourcing pattern could be applied, which is keeping all the business events happened in an event log which result in possibility of building the data projection as well as the state of the system at any needed point of time (the state in the past).

Credit card could be a good example. Let's imagine this: a man expects to see one thousand dollars on his credit card but for some reason (which he cannot see at his bank statement) he sees one hundred dollars instead. He makes a call to bank and asks why he has less money than expected. And if somebody from the bank's side will just tell him that I'm looking at the MySQL table and this is what's database table says he probably wouldn't be really satisfied with the answer. Money is really sensitive issue and he expects that the bank employee will tell him something like: I see your transactions and you probably forgot that you gave your credit card to your wife, she went shopping and just spent nine hundred dollars – and here is the transaction she made.

Other term to take into account is idempotency – the mathematical term and the meaning that any message sent in the distributed system could be retried. Anyone who is familiar with the RabbitMQ knows that it supports at least once method of delivery; the published message would be delivered at

least once to the handler – meaning it can deliver the same message more than once.

Because of some network problems the broker of the message queue cannot get the notification that handler already received and processed that message.

That's why broker will still consider that the message has not been processed yet, so it will repeat the message delivery to the handler.

This means that the message should represent an operation in such a way that operation can be applied several times and that the result must be the same.

It also means that putting such operation as increment which is not idempotent one in a message queue might be a bad idea (for example operation that adds +1 on the handler side), since the handler could accept that operation several times instead of just applying it once resulting in the incorrect final state on handler side.

What would be an idempotent operation in this case – is set a variable to five, because it can be applied any number of times and the result is always going to be the same.

DESIGN PRINCIPLES TO FOLLOW

- **All operations must be idempotent.** This means that all operations can be retried; it is possible to execute this operation several times.
- **Sagas should be used for consistency of distributed operations.** Let's consider an order for example – it is complicated mechanism so it should be split on several independent operations and each of them should be represented as a Saga. It should proceed from one operation to another and guarantee the atomicity and the consistency of each particular operation. When facing with some inconsistency or maybe some problem with operation and saga, it should proceed with compensational transactions and the compensational operation to compensate all of that operation which are already applied and persisted.
- **All new service contracts should expose asynchronous APIs.** The set of contracts we provide in the scope of Magento 2. We segregated those services of the implementation and we commit to preserve the backwards compatibility in all further releases of the system;

however we rely on developers to depend on those sets of APIs and not on the private code resided in the scope of Magento modules. Existing state of Magento Service Layer represented by synchronous services, but a new generation of service contracts must expose the asynchronous nature of APIs to avoid runtime coupling between services.

When dealing with a lot of data recovered from the product repository – if somebody wants to recover thousands of products via API, it really means to process every single product in its entirety, transform those specific products to the PHP object representation and return the link to the product collection.

All of those operations are really time-consuming and impact can cause performance degradation – that's why it's important to have all APIs to work in the asynchronous way. This means that if someone wants to proceed the command, import 1000 products, box of data can be send to Magento and there's no need to wait until Magento will handle this product.

Magento will send a link with 202 response saying „I accepted your request and here's the link where you can track the results“ and as soon as the results would be ready the link will provide the necessary product data.

- **All new state modifying operations should expose bulk APIs.** On the service layer Mageto worked with just one entity (like, Repository::save operation for entity persistence) but not with the bulk endpoints. Developers started to fix this problem and Magento 2.2 is more focused on the bulk APIs and there is possibility to proceed with many entities handled at the level of one API code.
- **All service operations must be stateless.** State between two subsequent service calls shouldn't be kept so having the same input data, service should always guarantee that it will respond with the same result.
- **There musn't be data dependencies between services.** All isolated services must have own independent data storage, and that storage should be considered as a private one so that no other isolated services could access it bypassing the service layer of isolated service. It means that two separate applications (for example inventory and catalog) shouldn't use the same database, so when writing the application from the point of inventory it is not allowed to touch and make joins on the catalog table.
- **CAS (compare-and-swap) technique to avoid race**



condition. This means that some entities are going to be signed to avoid race conditions – like on a product page in Magento, which has a bunch of product attributes, along with that the product quantity and stock availability. Let's consider a race condition example between two Magento administrators affecting the product data in different way. One of them may visit a product page to change some product attributes or upload a photo, and with this process, when one administrator changes the product attributes, other Magento administrator processes an order for this product – resulting in product quantity change in the process of order fulfilment.

First administrator is not aware of it and continues to make changes in the product attributes (sending the whole form with all attributes reside on the page to server). This will cause all the data sent by administrator one to be saved both for product attributes as well as for product quantity, despite the fact that administrator one did not change the product quantity anyhow. But persisting the previous version of quantity (because administrator one opened the product

page before other administrator shipped the order), which does not include changes from the transaction processed by administrator two will lead to losing changes applied in order fulfillment transaction made by administrative two. How the CaS technique is going to be used here: Magento can keep the version of the product and along with the attribute value system can always send the previous version of that value and can check whether there are applied changes to the product and if the product is being updated from the very last state to the new one or if there were any transaction in the middle. When applying compare-and-swap, first administrator would be told by the system that the product he is trying to persist already has a new version of some particular attribute.

THE MODULARITY PART

There are independent layer of abstraction interfaces and they supposed to be used to make sure that Magento will apply these changes to its architecture introducing the possibilities of scalability and distribution in different web servers. But as long as somebody is using the layer of services, the

backwards compatibility will be guaranteed for upcoming upgrades to the new Magento versions.

PWA is really big buzz word not only in Magento world but in e-commerce in general. This is the way how the admin is separated out of the storefront and for the storefront there is totally independent set of tools and APIs. So there are PWA + GraphQL and for the admin panel – the rest of APIs and UI component. (slajd 24)

Along with that distribution on different set of stacks: front-end stack and the admin stack, there will be module split. One big module would be splitted on many different, independent and small modules; each of them will be responsible for a particular type of operation e.g. storefront module, GraphQL module, PWA catalog.

In Multi-Source Inventory (MSI) project which has been already released so that its source code is available and developers could see how that idea is applied there there are two kind of UI modules – with the prefix front-end UI and with the prefix admin UI, so that in admin UI prefix there's a stack of presentations for admin like UI components and for the front-end PWA Studio and all of that kind of tools along with the API dedicated only for the front-end.

In MSI, for each particular functionality there is not one module but a bunch of modules like a front-end UI, admin UI modules, CLI commands, independent modules for API and independent modules for implementation – all of this for easy switch ability between different implementations using the interfaces provided in a scope of the out-of-the-box Magento. That means that 3rd party developer could easily substitute out of the box implementation provided by Magento with own one, relying just on interfaces which reside in a dedicated API module, and dismantle out of the box implementation from the codebase at all.

In another example it is worth to use headless Magento – it's pretty important for inventory because many merchants use inventory just for synchronization with some external ERP system, and they don't use Magento mechanism as a source of truth to process the inventory, with the order management happening in the external system – meaning there's no need to have Admin UI to deal with Inventory management on the Magento level, and the whole Admin UI part responsible Inventory management could be dismantle.

DATABASE SPLIT

One of the ground base rules is that isolated services are not allowed to share the data between the applications, and Magento already support the database split for the order management, checkout and main database (catalog) but currently it happens just with the Commerce Edition. The data relation decoupling should be performed in Open Source edition of Magento to simplify extension development. This means that Magento is going away from MySQL joins which currently combine many tables provided by different modules. Joins are allowed just in the scope of Isolated Service.

Today scaling Magento is about replicating the whole Magento application on web servers many times. There might be some kind of round-robin and the service which just sent the clients request a particular service for position but now this routine gonna be made on the level of application by itself but not on the level of external application, and if merchant has a particular bottleneck, like checkout which prevents the system to handle more orders per time, a merchant could scale the checkout service not affecting other parts of the application.

Each particular request it going to be handled by many different servers and services in an asynchronous way.

Magento is switching to the heterogeneous environment which is represented by a network, but there are some fallacies of distributed computing, and all of problems like latency, network reliability, security, and network should be included and should be handled by the application.

It is really hard to manage that kind of application, so Magento is going to introduce a service mesh – technique which provides an ability to wrap our services, responsible for mitigating all of those faults. So that, 3rd party Magento developers will continue to program just a business logic for their extension while the infrastructure part, like resilience during the network communication is the responsibility of Magento framework itself. All that additional complexity is the responsibility of Magento framework, but not the code of 3rd party Magento extensions.

The final goal of Magento is to preserve single box deployment for the medium-sized merchants and to

provide an ability of horizontal scalability for the big merchants.

This is going to be done on the level of service mesh, so the service mesh based on the configuration made by merchants will decide whether the particular code to the service should be pointed to the external service and send via the network or just a dropped as a message to the message queue or that particular service should be executive in a synchronous way on the same physical server as it happens now.

To make development of distributed instances easier, a new developer environment must be created. The environment should allow to easily manage multiple services of Magento.

Two developer environments are evaluated:

- **Environment based on Minikube VM with Kubernetes cluster**
- **Environment based on Docker Compose**

Isolated services are everywhere and the manual testing is not an option anymore – that's why Magento is promoting the Magento Functional Testing Framework (MFTF) framework which provides an ability to write acceptance tests on Magento. MFTF framework will validate all the extension currently available on the Magento Marketplace. Magento is going to run it on cloud-based solution to validate any changes before deployment of cloud.

This is the best way to effectively test extension and customization, investing as little time as possible, because if an extension is already existing, it's probably too late to rise the unit test covering that extension, maybe it's too late to write integration test for it, but it's not too late

to write an acceptance test. And even writing a few acceptance tests one can be sure that at least the P0 and P1 of the most critical scenarios of extensions are preserved and nothing critical is broken.

WHAT NEXT?

Magento's main goal is to release more often – as soon as it's ready and if some functionality is ready not to wait for an appropriate release of the whole system to make feature live – just drop a feature on the market and provide an ability for early adopters of getting the benefits of using it first, and then to introduce improvements based on their feedbacks. Some of those features could be released in a beta version, to preserve an ability to modify them, based on feedbacks.

This is what Magento is already doing with Multi-Source Inventory project – releases in MSI are independent from Magento releases, even so that MSI is considered as Magento core.

Next stage is to create new lightweight application framework, because when processing differently front-end and the back-end request, it's going to move further to the asynchronous logic. There's no need of layout processing or URL rewrite logic, if someone suppose just to create a command like order placement. This one of the change which will get Magento a step closer to the Command Query Responsibility Segregation (CQRS) implementation on a level of framework, when Query and Command logic handles differently.

And then Magento will be establishing and modularizing the isolated services like pricing, checkout, customer, tax, order management, CMS, reports, payments and so on, so the system will be migrating towards an isolated services representation. ●



Igor Minialo

Igor has been working for Magento for the last 7 years. He designed such functionality as Magento 2 Search and Async operations. Also, he has been involved in many general stuff routines as Magento 2 Service Contracts and performance optimizations on different levels.

For the last two years, Igor led Multi-Source Inventory (MSI) project, which introduced Multi Stock functionality into the Magento 2 platform. This big and very the important core functionality is being developed and designed in a community-driven way. Which is a unique experience for all the participants and Magento itself. Because being an Open Source, it is necessary to keep "the door open" for Community's contributions and ideas.

Currently, Igor works in Magento Architectural Team on the position of Lead Architect.

I've written articles in the past on how to improve Docker filesystem mounting performance, including using the overlay2 filesystem and performance tuning Docker for Mac. While they have drastically increased filesystem throughput and performance, none of them have been a silver bullet in truly solving filesystem performance issues with Docker for Mac.

DOCKER FOR MAC FILESYSTEM VOLUME MOUNT APPROACH FOR PERFORMANCE



The single biggest performance improvement Docker for Mac has made in the last couple years is adding the delegated flag for mounting filesystems. The “weak set of guarantees” noted by Docker is really quite sufficient, as there are really no noticeable changes between the default and delegated volume mount types. Working with open source software with over one hundred thousand lines of code (~Magento), I’ve noticed no issues by choosing to run this type of volume mount.

Speaking of Magento, working with a project with this many lines of code has exacerbated Docker for Mac’s volume performance issues. While a typical small PHP or small JavaScript application works just fine with Docker for Mac (with or without delegated volumes), you must

make one more additional concession to get 90%+ truly native speed: do not mount volumes treated as a “cache”. In fact, don’t mount any volumes that are part of “core” open source software code. Typically, this is code you do not have to edit anyways, as modifications to core OSS files are generally lost during upgrades anyway.

I’m going to talk into more detail about the Magento filesystem, as this is the system I am most familiar with and it presents a few different issues. That said, this approach should really apply to just about anyone having filesystem performance issues with Docker for Mac volume mounts, due to the way D4M operates as a sort of virtual machine to run on the Docker server daemon.

Here is the normal Magento filesystem, with explanations of each file/folder:

CHANGELOG.md, **LICENSE.txt**, **COPYING.txt**, **LICENSE_AFL.txt** readme/license files
Gruntfile.js.sample, **grunt-config.json.sample**, **auth.json.sample**, **nginx.conf.sample**, **package.json.sample**, **php.ini.sample** sample configuration files for respective packages
app/: folder for local code changes/updates
bin/: folder for helper scripts
composer.json, **composer.lock**: Composer configuration files
dev/: folder for built-in dev tools and unit testing
generated/: cache directory for code generation
index.php: main bootstrap file that handles internal requests
lib/: directory that contains frontend library scripts
phpserver/: folder for server configuration using built-in PHP web server
pub/: public directory for web server access containing files for routing/requesting static assets
setup/, **update/**: directories for web installers/updaters
var/: directory to contain temp files (sessions, logs, cache, etc)
vendor/: folder for artifacts from composer/library install command

Performance issues arose when mounting this entire filesystem. The source for for Magento was placed within a src directory, and mounted like so within a Docker Compose file:

```
volumes:
  -./src:/var/www/html: delegated
```

While this appears to be the most straight-forward way to work with a project on Docker for Mac, it's not performant at all, due to the vast amount of files and folders that are written to and read from. Using the above, page load times were roughly 14 seconds from a stale cache — just about twice as long as requests would be, versus running the application on the host machine without Docker at all. While acceptable for development, this really isn't as efficient as it should be.

Let's talk a bit how development works with Magento. Typically, all code updates and changes are handled in one place: the **app** folder. This is where local changes go, and really the one true place we need bi-directional sync of folders and files from and to Docker. So, why not just use native Docker volumes, and mount just that **app** directory?

```
volumes:
  - appdata:/var/www/html
  -./src/app:/var/www/html/app: delegated
```

We may also want a few other things synced, such as composer caches (which are infrequently used, just during times of installing composer dependencies), composer configuration files, and perhaps an Nginx configuration file. We may also want to specify the exact subdirectories within the **app** directory, to really fine tune performance:

```
volumes:
  - appdata:/var/www/html
  - ~/.composer:/var/www/.composer: delegated
  - ./src/app/code:/var/www/html/app/code: delegated
  - ./src/app/design:/var/www/html/app/design: delegated
  - ./src/app/etc:/var/www/html/app/etc: delegated
  - ./src/composer.json:/var/www/html/composer.json: delegated
  - ./src/composer.lock:/var/www/html/composer.lock: delegated
  - ./src/nginx.conf.sample:/var/www/html/nginx.conf: delegated
```

What do we do with our other directories and files that aren't mounted? Nothing. Upon setting up the initial project, we copy over all the files from our project to the native Docker volume. A simple bin script may help (fig. 1).

Then by running **bin/copytocontainer --all**, we can copy over our entire Magento filesystem to our PHP container. Once those files are in the container, we really never need them back out, while we also have a copy of the folders and files locally for debugging. We can even make an

```
#!/bin/bash
# Filename: bin/copytocontainer
[-z "$1"] && echo „Please specify a directory or file to copy to container (ex. vendor, --all) ” && exit
if [ "$1" == "--all" ]; then
docker cp src/./$(docker-compose ps|grep phpfpmp|awk '{print $1}'):var/www/html/
echo „Completed copying all files from host to container”
else
docker cp src/$1 $(docker-compose ps|grep phpfpmp|awk '{print $1}'):var/www/html/
echo „Completed copying $1 from host to container”
fi
```

Fig. 1 Bin script

```
#!/bin/bash
# Filename: bin/copyfromcontainer
[-z "$1"] && echo „Please specify a directory or file to copy from container (ex. vendor, --all) ” && exit
if [ "$1" == "--all" ]; then
docker cp $(docker-compose ps|grep phpfpmp|awk '{print $1}'):var/www/html/./src/
echo „Completed copying all files from container to host”
else
docker cp $(docker-compose ps|grep phpfpmp|awk '{print $1}'):var/www/html/$1 src/
echo „Completed copying $1 from container to host”
fi
```

Fig. 2 Inverse script

inverse script to copy things from the container, back to our host, in the event items change at certain times depending on our app (fig. 2).

The main performance improvements come from not mounting things considered “caching” directories that are heavily read-intensive during web server or app requests. Our biggest wins are these four directories:

generated/: This folder contains files and folders that are automatically-generated on demand, that use Magento’s “hook/plugin” system. Since this folder can contain hundreds or even thousands of files, and need to be written to and read from on every request, this is an obvious folder to keep within the native Docker filesystem. We also do not need bi-directional sync. The only time we’ll need to be able to view these files is during certain times of de-

Hopefully one day, we will not even need to do this! But until then, this is a pretty solid solution that works well for the edge-cases for OSS with very large filesystems.



Mark Shust

Is a Certified Magento Developer & Architect with extended knowledge of PHP, JavaScript, Laravel, React, Docker, and user interface design. He has nearly 20 years of web development experience in the eCommerce, real estate, business-to-business, and information technology industries. Mark is a Zend Certified Engineer, a 3x Magento Certified Developer, and has a wide knowledge range of other open source software and tech stacks.

Mark is married to his wonderful wife Juliann, and they are the proud parents of twin girls Lily Ann and Brielle. Mark lives in Northeast Ohio, and is a devout Cleveland Browns fan, loves eating Chipotle burritos, playing scrabble and solving Rubix cubes. Mark loves geek-inspired technology, and enjoys living a simplistic lifestyle.

Mark is currently working on a complete set of courses for the Magento ecosystem. You can see which courses are currently available and signup for updates by visiting

<https://markshust.com/courses>

bugging, in which case we can just use the **bin/copyfromcontainer generated** command to copy files back to our host for debugging.

pub/: Every single one of our web requests and requests for public access makes a request to this folder. This is most likely our heaviest-requested access point in our entire app. Since it's read heavy, and we aren't ever really editing files within this directory, we just keep this in the native Docker volume.

var/: Here is a very heavily-utilized folder in Magento, as it contains all cache files, session files, log files, etc. This is the true definition of a caching/temp directory. We rarely need access to anything in this file. If we needed access, we can always connect to the container to inspect the files needed, use the **bin/copyfrom-container var** command, or mount a specific subfolder we need access to (example: mount **var/log** to the host for bidirectional sync will only mount the var/log folder, which isn't a heavily written-to or requested-from folder).

vendor/ This is considered an "artifact" directory. Third-party Magento modules get installed here from Composer. It almost certainly contains the highest amount of read from files, and requests hit this directory many, many times on each and every code and app request.

Also, we really never edit files here. So, let's keep this just in a native Docker volume. If you did handle development within a subfolder of here instead of the **app** directory, just mount the namespace of your specific directory (ex: **vendor/foo**).

What happens after we apply these updates? Our initial requests of 14 seconds drop to around 7 seconds, and we achieve app performance that roughly 90% or better of native speed/performance. All of the main filesystem reads now happen within the native Docker container/volume, and don't get need to loop through our host machine unnecessarily. Doing this is how we can achieve a significant improvement in I/O within our Docker development environment.

I'm hoping this article helps put an end to filesystem performance issues on Docker for Mac. Hopefully one day, we will not even need to do this! But until then, this is a pretty solid solution that works well for the edge-cases for OSS with very large filesystems.

If you wish to check out my Magento Docker development environment in it's entirety and be able to use many other bin bash helper scripts I've created to deal with common issues, check out my GitHub project at [markshust/docker-magento](https://github.com/markshust/docker-magento). ●

JOIN US IN THE MAGENTO ASSOCIATION

Many of you first heard about the Magento Association a year ago, at Magento Imagine 2018. In that time, quite a bit has happened. Thanks to the efforts of Magento and the Founding Task Force, the Magento Association is a non-profit association incorporated in the state of Illinois. We have a mission, vision and culture statement and we have corporate bylaws. The Task Force even built out a series of guiding principles and a proposed strategic plan for meeting those principles.

Earlier this year, a volunteer Board of Directors was established to guide the Association. I have the honor of being elected as the Chair of that Board. Over the past few months, we've been hard at work building on the efforts of the Founding Task Force, creating and approving our Strategic Plan for the Magento Association. Additionally, we've been doing the work necessary for one of our most exciting announcements at Magento Imagine 2019 - opening up the membership process so that members of the Magento community that want to get involved can now officially join the Magento Association.

Going forward, we will be launching several committees that we will ask our members to join. These committees will drive the efforts of the Association and

will focus on areas such as diversity, membership, events, communications and more. These committees will help us reach the vision for the Magento Association - "an open, healthy and powerful Magento ecosystem for continued success of future generations."

For me, that success goes beyond the current Meet Magento events. I am excited to support the current Meet Magento events, but I know as a community, there is so much potential for additional initiatives to promote knowledge sharing and growth within our community. Magezine is an excellent example of that - a group of energetic volunteers saw a need in the community and set out to meet it. It's my pleasure to be able to encourage this spirit in our community and personally promote initiatives such as Magezine.

Over time, the Magento Association will expand to support, guide and encourage many projects like Magezine, but we can only do it with the support and input of the entire Magento ecosystem. Please visit us at MagentoAssociation.org to join the Association and volunteer for one of our committees. I look forward to working aside each of you that are reading this column to ensure the Magento ecosystem remains healthy and thriving for many, many years. ●



Joshua Warren

Joshua has worked in eCommerce since 1999. He serves as the Chair of the Magento Association as well as CEO of Creatuity. As a frequent conference speaker and author, he is constantly exploring the future of commerce and how technology will provide richer, more engaging shopping experiences both online and offline.

BEHIND THE SCENE

Interview with James Zetlen,
PWA Studio Magento Contributor

A year has passed since the announcement of PWA Studio by Magento. Despite this, it seems that in 2019 is still a very hot topic. That is why we asked James Zetlen – Front-end Architect and one of the lead architects of the Magento PWA Studio for details related to progressive web applications. In an interesting conversation, he told us about his predictions about PWA, the advantages of this solution and the changes that will soon be introduced.

Magezine: Last year, one of the key trends in mobile technology was the huge increase in popularity of progressive web applications. There are some of the big names who have already started benefiting through PWAs, such as Twitter, Forbes, Starbucks, Washington Post, Pinterest, etc. Is it true that now the whole developer world is switching to PWA? What are your predictions and expectations from the world of progressive web apps technologies?

James Zetlen: PWA sounds like a completely new platform, but it's not; it's a convenient term for a handful of new browser features and design patterns. So it's an evolution of Web development, much like "Responsive" has been.

So the whole developer world IS moving towards PWA, just like we moved towards Responsive before. We're not abandoning some existing platform and switching to a new one; we're following the new capabilities of the Web platform.

I think everyone agrees that these things will become normal for new Web projects. The core technologies of PWAs—ServiceWorkers, the Cache API, dynamic imports—are Web standards and not proprietary features (even though you mostly hear about them from Google), so they'll definitely become de facto standards as well.

I know there's some doubt about that, and it's understandable. Many sites have added those features without seeing significant return on their investment. As more developers learn the new best practices, the next wave of projects won't just add the features; they'll redesign their sites around the features, thus producing the real performance boosts and native feel that PWAs promise. The advantage, and business need, to adopt PWA will become clearer to everyone over time.



MGZ: When it comes to choosing the right PWA tool, Magento 2 developers have access to many options. Next to Magento PWA Studio there is Deity Falcon, Vue Storefront, Front-Commerce. Of course, all these tools are still in the development phase, but can we already say that they are a real competition for each other? Do you see significant advantages and disadvantages of PWA Studio in opposition to these solutions?

JZ: PWA Studio is not the same type of thing. The three products you mentioned are all great. Deity is building a service-oriented architecture and has done a great re-platforming with Falcon. Front-Commerce is a brilliant team doing amazing things with GraphQL. Vue-Storefront features trending technologies, great momentum, and an energetic promotion campaign. They certainly act as alternatives to working directly with Adobe/Magento using PWA Studio.

But I don't consider them competitors with our work. They're competing with each other, but we have worked with them to talk about standards and extensibility concepts. PWA Studio is a collection of tools, not just a storefront. The Venia storefront is the culmination of PWA Studio today, the whole orchestra playing together. But you can use each of our technologies alone; the Peregrine library for hooking up to data and implementing PWA best practices, the Buildpack toolkit for optimal bundling, compression and optimization, and UPWARD as an emerging Web app delivery spec that reaches beyond eCommerce alone in its usefulness.

We anticipate that other PWA projects will appear, and many will use our tools to help them integrate with Magento, its APIs and its marketplace. I wouldn't be surprised if Deity, Vue-Storefront, and Front Commerce do the same. We aren't just building a product for sale; we are building the next Magento frontend platform.

MGZ: In November 2018 Magento PWA Studio received a new feature update. At what stage of development is PWA Studio now? What has been done and what are you working on now? Can you divide the tool development process into stages?

JZ: We've released v2.0.0 and v2.1.0 of PWA Studio in 2019 already! The November update was a "prerelease", but now that we're in full release status, we are following semantic versioning. The big news in 2.0.0 was UPWARD and the new portable middle tier. In 2.1.0 we updated our GraphQL queries for the new Magento 2.3.1 release, in addition to several other bugfixes and smaller enhancements. All this information is available at <https://github.com/magento-research/pwa-studio/releases> by the way! All our processes are out in the open, on GitHub and ZenHub.

We are always working on additions and enhancements to the Venia storefront, with the generous help of community contributions. But the core team is also working on major additions to the architecture. I'm always excited about seven things at once, so when I write lists of upcoming features, I want to revise them only a day or two later. So bear that in mind, and please don't get me in trouble with product management.

We are about to move a lot of configuration and build tooling out of the Venia storefront and into the Buildpack library, so that storefront projects have less boilerplate and fewer direct dependencies. That will lead to a real "scaffolding" solution, so you don't have to clone our core repository to get started.

We are hard at work abstracting our data connectors and dynamic loading utilities into a new version of the Peregrine library. Peregrine will use React Hooks, a new ultra-simple API for mixing behavior and side effects into simple functional components.

I've consulted with a few really smart people about the future of UPWARD. We'll come out with a new version of the specification which makes config files clearer to read and analyze. We also want to make a ready-made Docker container with a generic UPWARD server which can serve any PWA. This container might include full server-side rendering!

As more Magento concepts move to GraphQL, more of our core data and state management in the PWA will start to look like GraphQL. We are moving slowly on this, because Redux is very mature and user-friendly, and we won't migrate

Redux actions and reducers to local GraphQL queries and resolvers until we know it's as easy to understand as Redux.

More, and more, and more...did you know that major browsers are about to get native lazy-loading? Did you know that we're ready for touch and swipe gestures, but we're working on a library for them that is small and fast enough to meet our standards?

PWA Studio was a totally different beast, with an architecture completely uncoupled from Magento core, a workflow and vision very different from Core Engineering, and an ambition to eventually persuade everyone to get on board with our systems voluntarily, because they will help everyone so much.

MGZ: Is PWA Studio ready to start working with it? How it looks from the point of view of the merchant, frontend and backend developers and other service providers?

JZ: The PWA Studio repository is absolutely ready to clone and start working with. The docs at pwastudio.io are excellent (and where they might have flaws, please let us know!) and it should be easy to get a local instance up and running.

Right now, we support one basic scenario: an end-to-end JavaScript developer working on a frontend app, who is comfortable working with the NodeJS toolchain as well or at least is happy to leave it alone with no customizations. But we're working on more tools to make it clearer how to get started:

- **A project scaffolding generator**, so you don't have to "start" with Venia, though you can import as much of it as you like
- **Individual components of Venia to be used as versioned dependencies, instead of the whole app**
- **A containerized application for running a PWA on an UPWARD server**
- **Extension mechanisms and bundling so that extension creators and vendors can support PWA in their storefront UI changes.** (This will be driven by a binding

relationship between GraphQL concrete types and components that "know how to display them".

MGZ: What do they often ask you about, what are they afraid of, or what is an obstacle for them?

JZ:

- PWA marketing often makes it sound like a brand new platform, and it isn't. It's the Web, with newer and better features and some more strict best practices.
- Working mobile-first is tough, because the workflows we've all learned over the last years are not very mobile-friendly. Sure, you can click the icon in Chrome Developer Tools to simulate a phone's shape, but most developers are not running simultaneous integration tests on local iOS and Android emulators. That will have to change. But on the bright side: I think everyone knows that the computing devices we use the most are our phones. It should feel very natural to start targeting phones instead of desktop computers.
- The tools to learn (ServiceWorkers, Cache APIs, asynchronous loading) are complicated at first. But they make sense, and our tools automate your use of a lot of them.
- Magento Core Engineering has been very frank and transparent about mistakes we have made in previous APIs. Some concepts have been overly complicated. Some have been unfinished. Some have been difficult to "reason about", in the sense that a developer has a hard time in her editor keeping track of causes and effects. Our community is very vigilant about introducing new complexities. So, every time the PWA Studio team introduces a new abstraction or concept, we know we have to explain, clearly and repeatedly, why we think it's necessary. That's as it should be.

MGZ: What are you most proud of when it comes to progress and new functionalities?

JZ: This is easy. I'm most proud of the core team and generous community which has blessed this project. We have come together around some small and vague ideas of mine, and turned them into robust, powerful systems. I sit around and invent acronyms, mostly. The team produces new beautiful Venia features, improves security and deployment code, works tirelessly on test coverage, and continually invents and debates new UX principles.

I began with an intent to make a good developer experience, and I remain focused on that intent. But meanwhile, folks on our development, QA, and UX teams have concentrated on a good shopper experience and starter app. Soumya Ashok, Kayden Althen, James Calcaben, Devagouda Patil, Stephen Rugh, Jimmy Sanford, Andy Terranova, and Tommy Wiebell and many brilliant people in our community, too many to name have taken ownership of this project in a way I could never have dared wish for.

MGZ: What is the greatest strength of the team working on the development of PWA Studio?

JZ: Can I say everything? I'm gonna go ahead and say everything.

Magento has shown itself to be a very adaptable organization and corporate culture, despite the long legacy of parts of its codebase. But PWA Studio was a totally different beast, with an architecture completely uncoupled from Magento core, a workflow and vision very different from Core Engineering, and an ambition to eventually persuade everyone to get on board with our systems voluntarily, because they will help everyone so much. Both our team, and the teams surrounding us and supporting us, have risen to that challenge. It makes me very happy.

James, thank you for the interview!

By Magezine Team



James Zetlen

James is the architect of the new PWA frontend of Magento: Magento PWA Studio. He has built eCommerce storefronts for years, and how he's helping Magento build a more flexible toolbox so that developers don't need to build PWAs from scratch.

Change is driven by opportunity, and opportunity is driven by change. It is the inevitable outcome and driver of commerce, a force borne of the confluence of markets, consumers, brands, and customers. The endless variability of commerce is a fundamental assumption of the Magento platform and its extension and integration ecosystem. The Magento community is no stranger to change. Rather, we expect it. We engage it. We embrace it.

THE NEXT LEADER OF MAGENTO

As a company, Magento itself is also no stranger to change. We are part of Adobe now, totally and completely. This is underscored by the departure of our former CEO Mark Lavelle who led Magento through three distinct transitions since 2014: eBay to eBay Enterprise; eBay Enterprise to private equity; private equity to an Adobe company. His leadership has been a study in expertise, agility, and faithfulness to the soul of Magento's vibrant, vocal, and passionate ecosystem. I have been proud on his behalf to see his dedication, recognized by employees and community members alike. His departure is a bittersweet celebration, but a well-earned reward for ensuring the long-term success of Magento, a multi-billion-dollar force in all of our lives. Thank you, Mark.

Lavelle's departure has the ecosystem asking, „Who will lead us now?“ It's an important question because until this time, there has always been a leader at the top of the Magento organization. There has always been one person to put our faith in, to challenge when we experience dissonance between messaging and reality, and to steer us through to success. This is the most atypical change we've seen since the eBay acquisition in 2011. And yet, the ecosystem seems calm, faithful, and most importantly, as engaged as ever.

I feel that this patient calm is due in part to our remaining leadership: Jason Woosley, Gary Specter, and Mark Lenhard. They have each had massive positive impact on our business, and in my work with each of them I have enjoyed the pleasure of engaged disagreement, earnest support, and above

all, a commitment to our community. I expect this commitment to continue now that we are structurally integrated within Adobe, just as I expect the opportunities for all of us in the ecosystem to deepen and flourish given the unparalleled resources, access, and exposure that being a part of Adobe affords us. With all that said, what does the next leader of Magento look like?

*It looks like you,
dear reader. You
are the leadership,
you are the future.
You have the power
to effect tremendous
change and growth
both within Adobe
and among the
markets it serves.*

You have put your trust in us, and I am here to tell you that we have never been more committed to you, more trusting of you, and more able to put you in control of our shared destiny. You are here because you engage with change and you know that this place, this time, this opportunity is boundless – boundless precisely because the Magento community pushes against any and all boundaries, setting and resetting the „new normal“ for the rest of our industry.

With your leadership, we lead our futures together. ●



Ben Marks

Ben is a fifteen-year veteran of the open source ecommerce world. He is a developer, educator, and international speaker, having traveled millions of kilometers to speak with and learn from global commerce and developer communities. He is lead evangelist for Magento, an Adobe Company, and a board member of the Magento Association.

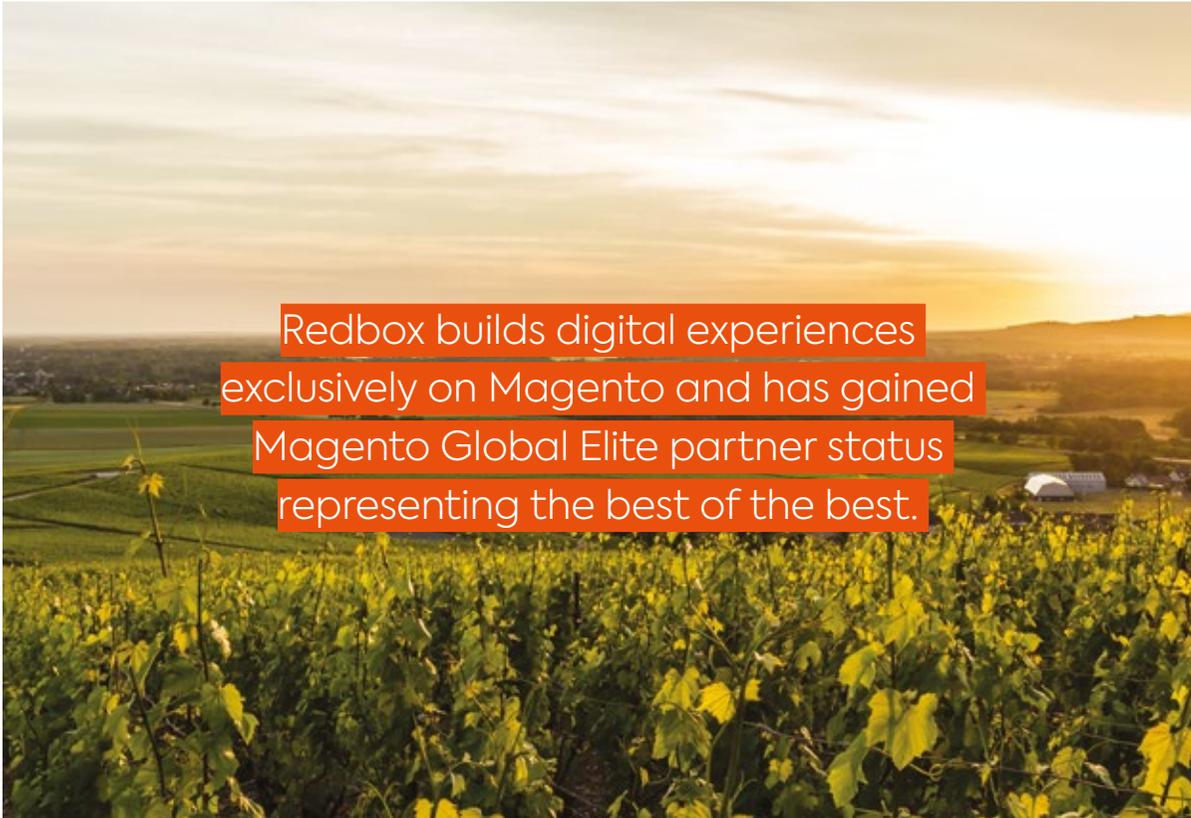
Top Magento events of 2019

The Magento community is growing year by year – we are very happy about it! As a result, the number of events, conferences and meetups is also increasing. We have collected the most important and the most interesting ones for you. See you on one of them!

Meet Magento DE	June 3 – 4	Leipzig, Germany
Meet Magento UK	June 18 – 19	London, UK
Meet Magento Indonesia	August 1	Jakarta, Indonesia
Meet Magento Singapore	August 22	Marina Bay Sands, Singapore
Meet Magento New York	September 5 – 6	New York City, NY
Meet Magento Poland	September 16 – 17	Katowice, Poland
Meet Magento Brazil	September 17	São Paulo, Brazil
Meet Magento Romania	September 26 – 27	Bucharest, Romania
Meet Magento Sweden	October 14 – 15	Stockholm, Sweden
MageTitans Austin	September 13	Austin, Texas
MageTitans Valencia	29 June	Valencia, Spain
MageUnconference	November 30	Koln, Germany
MageTitans Groningen	7 June	Groningen, Netherlands
Magento Meetup Twin Cities	20 June, 18 July	Minneapolis, MN
Los Angeles Magento Group	23 May	Los Angeles, CA
Magento Minds of Manchester	5 June	Manchester, UK
Magento-Stammtisch Aachen	17 June	Aachen, Germany
Magento User Group Overijssel	6 June	Zwolle, Netherlands
Magento Stammtisch Hamburg	22 May	Hamburg, Germany
Magento Meetup Kiel	12 June	Kiel, Germany
Magento Meetup Wroclaw	18 June	Wroclaw, Poland
Magento Meetup Austria	28 May	Vienna, Austria
Auckland Magento User Group	5 June, 7 August	Auckland, New Zeleand
Magento Meetup Kansai	11 June	Osaka, Japan
Magento Meetup Tokyo	23 May, 27 June, 25 July	Tokyo, Japan
MagentoLive Europe	22–23 October	Amsterdam, Netherlands



Redbox is a global digital commerce and design consultancy.



Redbox builds digital experiences exclusively on Magento and has gained Magento Global Elite partner status representing the best of the best.

Over the years, Redbox has helped some of the most iconic and ambitious brands digitally transform their business, including Nespresso, Sephora, Universal Music Group, Les Ambassadeurs, Paperchase, Nestle, The Folio Society, Bucherer, Chapel Down, AXA and Screwfix.

The company supports clients across the globe with operations in London, New York, Los Angeles, Modena, Dubai, Mauritius, Cape Town, Sydney and Auckland.

AWARDS

- 2019/18 Finalist Magento Imagine Excellence Awards - Universal Music Group
- 2018 Finalist Magento Imagine Excellence Awards - Redsea
- 2017 Spirit of Excellence Award - Magento Imagine
- 2017 Best Commerce Launch - Oliver Sweeney - Magento Imagine
- 2016 Magento Digital Commerce Partner of the Year
- 2016 Magento Great Explorer for International Expansion
- 2015 Magento Partner of Excellence
- 2014 Magento Partner of Excellence

• EMEA

London
+44 207 183 1833
Modena
+39 059 788 0144

• MENA

Dubai
+971 4 319 9327

• SOUTHERN AFRICA

Cape Town
+27 21 205 0002
Mauritius
+230 465 5777

• ASIA PACIFIC

Sydney
+61 2 8399 0594
Auckland
+64 9 638 4250

• THE AMERICAS

New York
+1 917 900 1833
Los Angeles
+1 310 807 0183



Every business has a unique story, unique set of challenges, goals and ways to achieve them, and we approach each of our clients in a unique way. By combining consulting, development and design services aligned with the culture based on integrity, technical excellence and passion, we are able to prepare and execute a high-performing commerce strategy.

INCHOO — A MERCHANT'S BEST FRIEND



Inchoo is an eCommerce agency from Croatia, providing high quality solutions for our clients on (and with) Magento for the last 11 years.

Our strong team of certified UX designers, developers, digital marketing analysts and consultants work together to help our B2C and B2B clients in various industries achieve continuous growth across all channels.

MAIN STRENGTHS

- **complex integrations with outside systems (ERP, PIM, CRM...)**
- **digital marketing services (SEO, PPC, Google Shopping, Advanced Analytics)**
- **no-nonsense, customer-focused UX design**
- **development of custom features and workflows**

WHAT OUR CLIENTS SAY?

They're very insightful, and typically offer us plenty of ideas when we're trying to solve problems.

I've worked in this industry for 15 years and Inchoo is as good a vendor as they come.

Inchoo's efforts have improved our platform's conversion rates, sales, and average page load time.

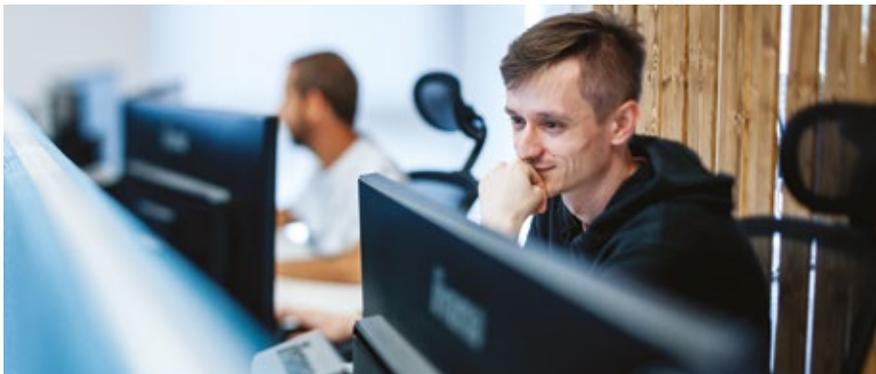
We like working with them, and they're more than an agency for us. We've become friends, and we can go to them with any problem we might have.

CONTACT US TO SEE WHAT WE CAN DO FOR YOU!

web: inchoo.net
email: b2b@inchoo.net
phone: +385 31 200 729



Strix is a team of 100+ experienced consultants and engineers, specializing in consulting and implementation of leading open-source technologies, such as Magento Enterprise Edition. Our activities are focused on developing business solutions for B2C and B2B companies and helping them to migrate from traditional sales models to the omnichannel model.



OMNICHANNEL MAGENTO CONSULTING

What matters is good handling of marketing communications and sale strategy, but only when aided by efficient product range and resource management. Many are the factors that can tip the scales to your advantage. And we happen to know the ones that can truly add value to your business. Selling over the Internet is a tough business, one that imposes refining your sale model. We help our clients to take the decisions, through which they will best fit sale strategies to their brand, organization and clients themselves. Used by thousands of businesses around the world, analytical and concept tools (Business Model Canvas) help us gear up to sell in the fast-moving Internet environment.

UX & DESIGN

We believe a sound design of an e-commerce platform is just like a well designed car. It blends this fine shape, an unparalleled sense of satisfaction that driving it gives you, and the efficient cutting-edge technology. And to achieve it, we need to be highly specialized. A product and business strategy, client's needs analysis, drafting, designing, prototyping, you name it – this variety calls for comprehensive knowledge and skills. And a variety of professions, for that matter. Engineers, architects, designers, sociologists and humanists – all of them are on our team. A

multidisciplinary team, we aim at solving specific business challenges. We leave no room to chance and have each decision discussed through and through. Our experience tells us best solutions are born through hard work and selected from a good few dozen of concepts. We do not hide behind catchy presentations but focus on iterating projects so that interactive prototypes can prove their worth for real users.

MAGENTO DEVELOPMENT

What makes clients pick us is the expertise of our implementation team, not its size. Our consultants and developers boast both certificates and hands-on experience; entrusted with tasks, they will deliver solutions, regardless of how difficult they are. There is more to the implementation process than just programming – it is also making decisions that translate into development, sales and scalability of the e-commerce business. Therefore, people who make up our implementation team are developers, but also business consultants and managers. Our cooperation with the Client thrives on our partnership. We realize mutual respect, understanding and trust condition the success of the implemented project. How swiftly we work is all down to simplicity. We do not overdo bureaucracy but concentrate on efficient prototyping of implemented solutions so that their business value can be quickly noticed and evaluated. ●



LOCATIONS

POLAND

- **Krakow**
Milkowskiego 5/3U
Krakow, 30-349
- **Rzeszow**
Litewska 4A/9
Rzeszow, 35-302
- **Poznan**
Sarmacka 11
Poznan, 61-616

CZECH REPUBLIC

- **Prague**
Prosecká 855/68
Prague, 190 00

LITHUANIA

- **Vilnius**
Smalines st. 21-3
Vilnius, Vilniaus
Apskritis LT-06225

CLIENTS

- **Castorama,**
- **Tous,**
- **Decathlon,**
- **Inter Cars,**
- **Nestlé,**
- **Super-Pharm,**
- **Mint of Poland**

THE INCHOOERS



COMIC BY HRVOJE JURISIC

MOBILE

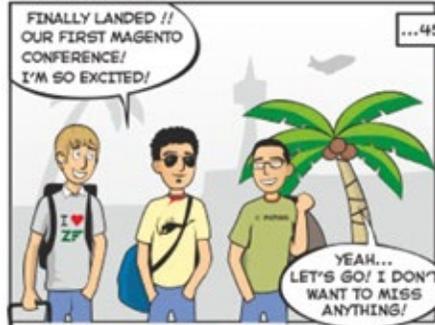
THE INCHOOERS



HRVOJE JURISIC

THE CORE

THE INCHOOERS



INCHOOERS COMIC BY HRVOJE JURISIC

MAGENTO CONFERENCE

MAGEZINE STAFF

EDITOR IN CHIEF
Borys Skraba

CONTENT EDITORS
Katarzyna Gądek,
Paulina Jóźwik

PRODUCER
Karolina Gajzler-Polak

DTP DESIGNER
Michał Hojnacki

**PHOTOS
AND ILLUSTRATIONS**
stock.adobe.com

WWW.MAGEZINE.CO

By Inchoo Team



New open-source PIM on the market

- Ready out-of-the-box for 150k+ product references.
 - Like-excel experience product management.
 - Fast and efficient.
 - PWA-Desktop approach.
-

www.ergonode.com

ZINE
EE
G
MA
M

Like it?
Want next issue?
Subscribe to our
magine.co