

A12 Release 2024.06 LTS

July 2024

Basic information about A12
<https://www.mgm-tp.com/a12.html>

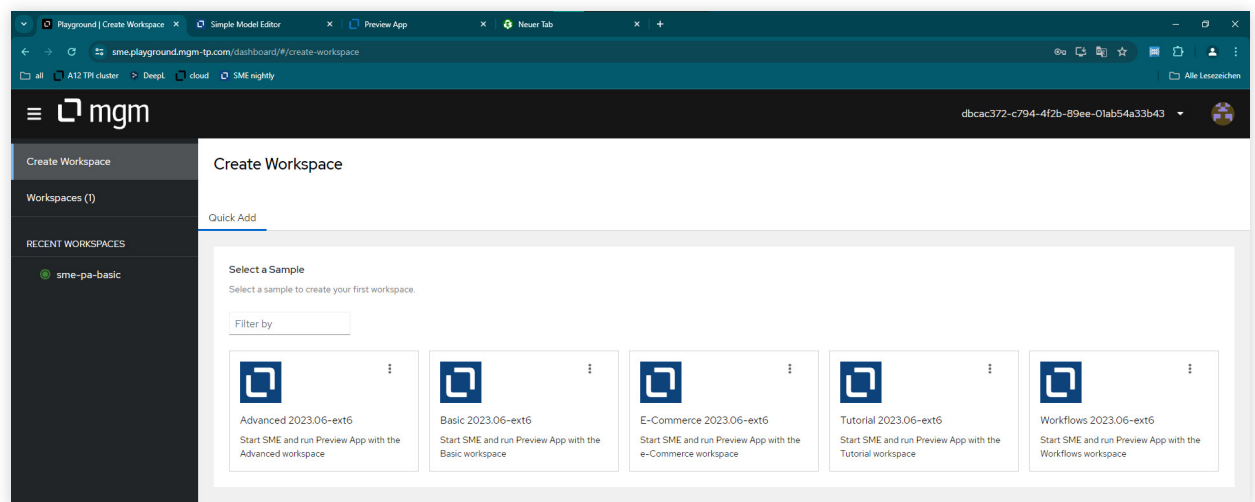
What is A12?

A12 is an enterprise low code platform for developing enterprise applications in complex IT landscapes. A12's modeling platform provides tools to quickly create and maintain parts of an application over the long term without programming experience.

A12's runtime platform provides the flexibility needed to evolve low code apps with professional individual software development and system integration into fully integrated enterprise applications.

CLOUD SME CME

Modeling in the Cloud



The A12 modeling environment is now also available as Software-as-a-Service (SaaS) from the cloud - and can be used without any installation. It can be accessed directly via the browser. Once users have logged in on the start page of the cloud offering, they are greeted by a new administration interface - the *Cloud Modeling Control*. New *Cloud Modeling Environment* workspaces (CME workspaces) can be created here and existing ones can be started. For new CME workspaces, the familiar 5 sample workspaces from the local modeling environment are available

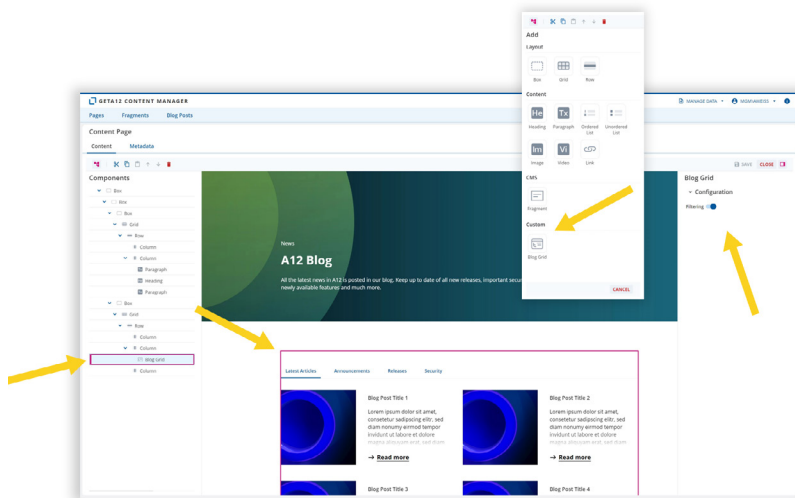
as a basis. After starting a workspace, the Simple Model Editor (SME) opens in a new tab and the Preview Application is available. The SME officially supports the Chrome (Windows, MacOS), Edge (Windows) and Chromium (Ubuntu) browsers. The models themselves are currently not persisted in the CME workspace. Users can simply load them from the local file system into the web application and save the edited models on their devices. For collaborative work on models and workspaces, we recommend using a version control system such as Git.



CMS

Content Management With Custom Elements

The A12 Content Management System (CMS) introduces new options to address project-specific requirements for the content design of web applications. It is now possible for development teams in A12 projects to define custom elements. They are made accessible to Content Managers in the Content Model Editor and can be used there in the same way as the standard elements for page design. The Content Engine renders the content according to the respective implementation. This extension mechanism allows A12 projects to create their own dynamic overviews for blog posts, for example. Existing components can now also be customized as required. The documentation for the CMS component is available at getA12.com. It summarizes how the editor works for content managers and shows development teams how they



can integrate the CMS as a full-stack component and how the content engine can also be used separately.

PERFORMANCE RENDERING

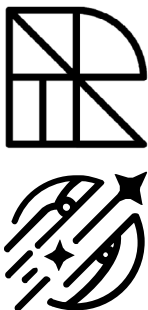
Improved Rendering Performance

The rendering of various A12 UI components and interactions has been optimized so that less rendering is required, client side performance is improved and applications react faster to user interaction. This can e.g. be observed when opening a

form on a master/detail screen from an overview or a tree. The faster responsiveness of the interface is also evident when dragging and dropping elements within a tree.

MIGRATION A12 UPGRADE

OpenRewrite & Codemod Reduce Effort When Upgrading to New A12 Versions



To benefit from new features and ensure the long-term security of their application, A12 projects should carry out a major upgrade to the latest A12 release line once a year. As part of the upgrade, several migration steps are necessary, which can affect the code base, the models and the underlying infrastructure. To reduce the manual effort involved in adapting the code base, A12 now

utilizes the auto-refactoring tools OpenRewrite for backend and Codemod for the frontend. They help to make changes such as renaming classes, packages and methods automatically. A tool for the automated migration of models has long been part of the A12 modeling environment and has been directly integrated into the Simple Model Editor (SME) since last year. The Technical Professional Services (TPS) team offers further support for the A12 upgrade and can be contacted at any time via the usual channels.



WORKFLOWS | BPMN | MIGRATION

New Command Line Tool for BPMN Migration

With the newly provided Workflows-Model-Migration-Tool, project teams can automatically migrate their BPMN models to new versions of the Workflows component of A12 and thus avoid manual effort. While not every new version of the Workflows component requires a migration of the BPMN models per se, this is necessary when upgrading to Workflows version 11 in release 2024.06 LTS. The reason for this are improvements

to the delegates provided - specifically the name change from `setDocumentFieldDelegate` to `setStatusDelegate`. With the new tool, the change is made automatically. The tool is provided as a jar file in the A12 Artifactory. A number of options are available to users to control the migration process. For example, the „-b“ flag is used to create a backup. A complete documentation of all options can be found in GetA12.

LOAD TESTS | DATA MIGRATION

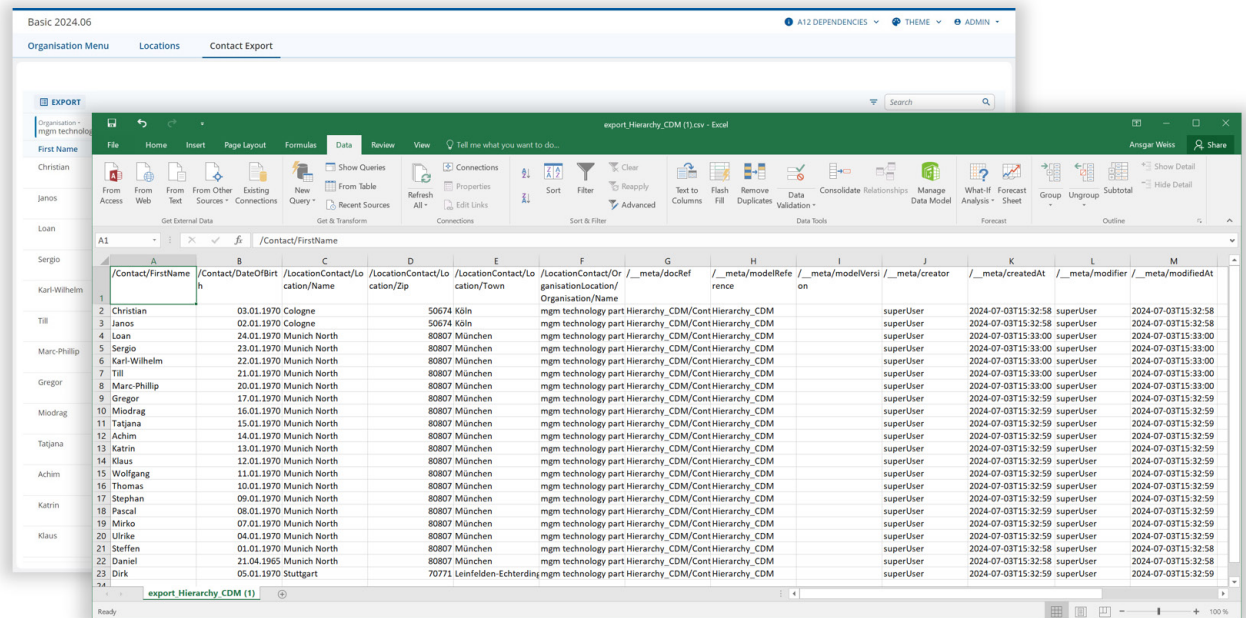
Data Migration Performance Put to the Test

In order to gradually improve the performance of the A12 platform, it is continuously put through systematic load and end-2-end tests. They help to identify and eliminate bottlenecks and develop best practices for the high-performance configuration of A12 applications. For example, the performance of data migration was examined in detail for the A12 LTS release 2024.06. Tests were carried out for two key steps - DocumentMetaDataMigrationStep and indexing (with a batch size of 10k) - each with around 2.9 million Documents. The former took a total of

458 minutes (~9.4ms/Document), the latter around 136 minutes (~2.8ms/Document). To speed up the migration, a new option has already been introduced which, once activated, excludes faulty documents from the process and flags them accordingly. In addition to examining specific processes, the load tests also brought previously undetected bottlenecks to light. For example, it turned out that uploading image files took a very long time due to an error in thumbnail generation. This problem has also been fixed in the current A12 release.

DATA EXPORT | CSV | EXCEL

Excel Export: A12 Simplifies the Further Use of Data From Overview Tables

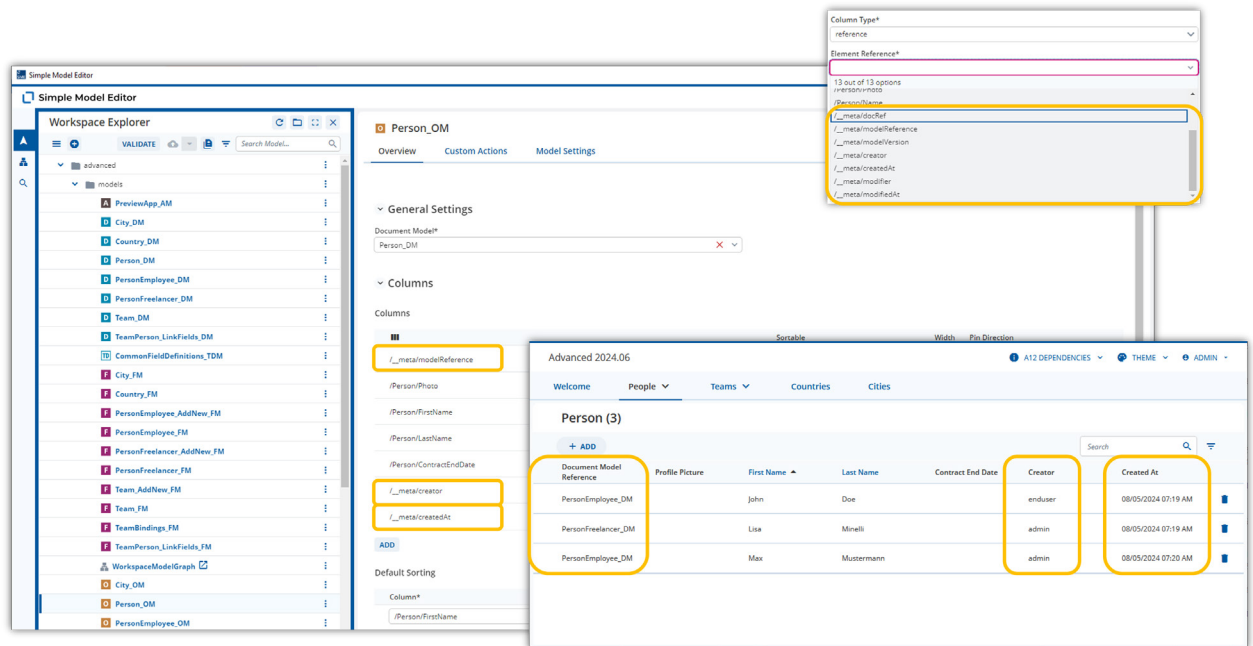


Overview Models can be used to represent extensive overview lists and data tables in A12 applications. They are brought to life by Overview Engines. In order to enable users to process this data outside the A12 application, new extension points are now

available for development teams. They enable the implementation of individual export mechanisms. For example, a selection filtered by the user can be easily exported to a CSV file, which can then be further processed in Excel.



Metadata Opens up New Possibilities for A12 Data Modeling



Metadata provides important information about A12 Documents - for example, when and by whom a Document was created or changed. As system-driven fields, they have long been an integral part of all Documents, but are now made accessible to all layers of the application. For example, they can be used in the Client and are also available for modelers in the Simple Model Editor (SME). Each Document Model that represents business content of the application now contains the group „_meta“ by default. The server-side A12 component Data Services automatically adds a set of metadata to each created or modified

data set (referred to as a „Document“ in A12 terminology - the instance of a „Document Model“). This set consists of 7 fields: docRef, modelReference, modelVersion, creator, createdAt, modifier and modifierAt. The field modelVersion is optional for now and opens the door to future features relating to model versioning. There is also an extensions group for project-specific metadata. In previous A12 releases, the Documents were persisted in XML format. Now this is done in JSON format for better performance.

PROJECT TEMPLATE

A12 Project Template Now Supports Trees, CDMs and Relationships as Standard

The A12 Project Template enables development teams to set up A12 projects quickly and in a standardized way. The standard configuration of the template now also supports three advanced functionalities of A12 out-of-the-box: the Tree Engine, Composed Document Models (CDMs) and Relationships. The Tree Engine is used for tree-like, hierarchical overviews that are defined using a Tree Model. CDMs make it possible to combine several Document Models in one Engine. Relationships are an important part of data modeling that define relationships between different Document Models. All these features are now supported in the Project Template standard. Manual setup

steps are no longer necessary. The previously available template variants for these features are obsolete. However, new template variants are already in the starting blocks. The team is currently working on ready-made configurations for a standardized integration of the Print Engine and the new Content Management System. Another benefit of the new template is that the integration of A12 Model Workspaces has been greatly simplified. For example, the development team can integrate additional modules with A12 models during the course of the project without having to write a single line of code in the Client.



Backend

KERNEL API

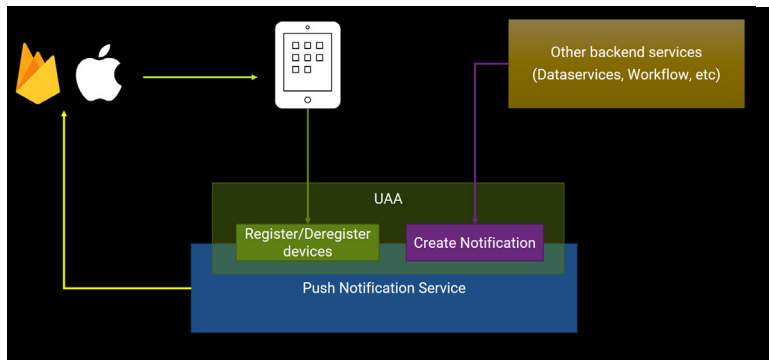
Document API V2 Simplifies Access to Data

So-called Document Models are at the heart of A12's data modeling. They are structured hierarchically and represent the domain-specific content of the application. We refer to instances of these models that are filled with specific data as „Documents“. To access this data, backend developers can now use a new Java API of the A12 component Kernel - the Document

API V2. It was introduced as *experimental* in October 2023 and has now reached *stable* status. This significantly simplifies data access, as (repeatable) subgroups are now also explicitly represented, for example, and a more comprehensive set of utility methods is available. The documentation is available in GetA12.

NOTIFICATION CENTER PUSH NOTIFICATION

Notification Center Now Also Supports Push Notifications



Push notifications are part of everyday life for smartphone users. A notification appearing either on the Lock-Screen or during usage is referred to as a push notification. Push notifications are opt-in messages which can be configured by the users individually for each app. It is up to the users to decide on which events they want to get notified. By using this feature, the users don't have to actively open the app to check for new incoming important messages.

The A12 Notification Center now also supports push notifications, which inform users at operating system level. The app-based push notifications complement the already existing web notifications, which are displayed in the browser via toast messages. To use this feature in a project, a push notification service needs to be registered - for example the Apple Push Notification Service or Firebase Cloud Messaging. The Notification Center then provides a `register/deregister device` endpoint where the app can activate or deactivate the notifications based on the user's preferences. Any service (like A12 Data Services or A12 Workflows) can then call the create notification endpoint to send out push notifications to dedicated users based on your use case.

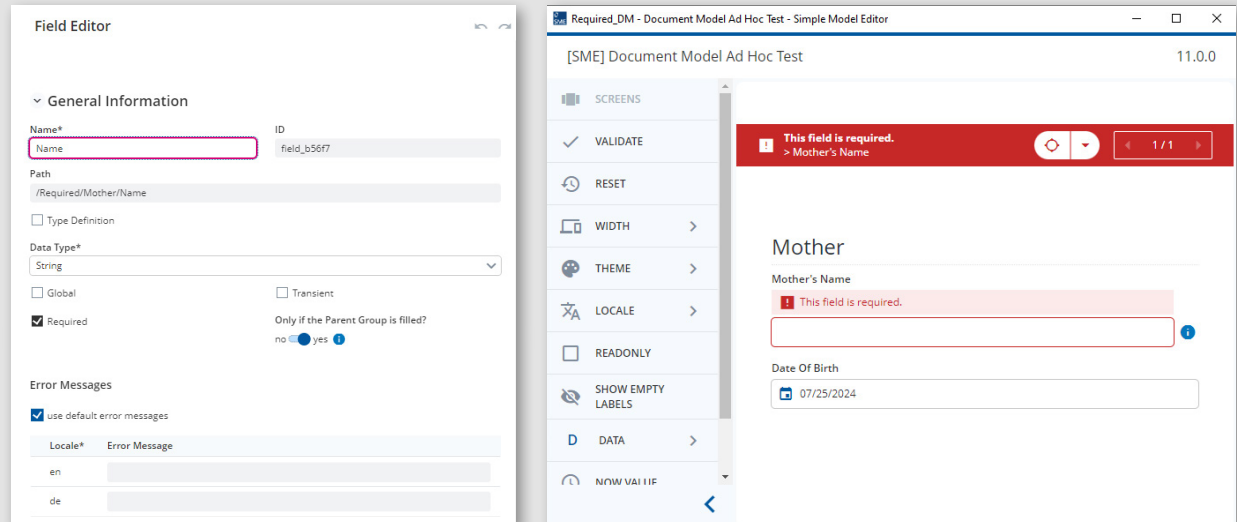
Offering app-based push notifications is a step towards opening the Notification Center to more channels. Support for further notification channels like e-mail, SMS or other messaging services is also planned in the future.



Modeling

SME DOCUMENT MODEL EDITOR REQUIRED FIELDS

Advanced Customization of Required Fields in the Document Model Editor



The Document Model Editor allows the creation and editing of domain-specific Document Models in the Simple Model Editor (SME). Within the Document Model, the “Required Fields” property determines whether a field is mandatory. The new checkbox “Only if parent group is filled” now makes it possible to consider the context of the surrounding group. If set to “Yes,” the field does not need to be filled out initially. However, once another

field in the group is filled, this field must also be completed. This ensures that certain datasets are complete. For example, if someone provides a birthdate, a name must also be provided. This way, it is possible to define the “minimum dataset” that is mandatory for a group. The error message for an empty required field can also be customized for each locale by unchecking the “use default error messages” checkbox.

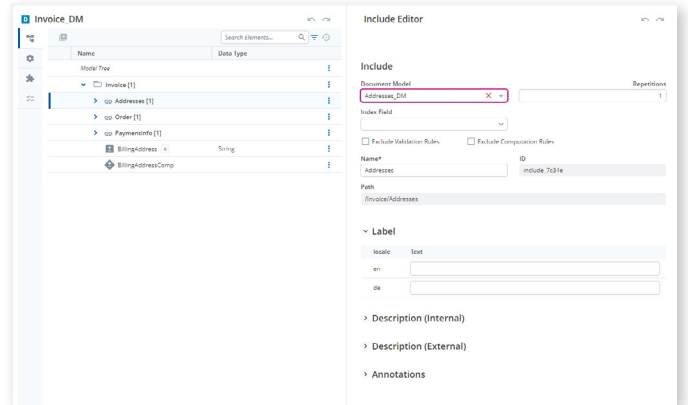
More News in a Nutshell

- The new A12 release line **2024.06 LTS** comes with **support for two years** while the release line **2022.06** has reached its end of life and is **retired**.
- **A12 2024.06 LTS** supports **Java versions 17 and 21**. The next release line **2025.06 LTS** will only support **Java 21**.
- The **A12 Security Guidelines** have been **expanded** and now explain even better from an **end-2-end perspective** how to ensure that **configuration** and **authorization** are **effectively secured**. **Checklists** are available for the most important steps in the project lifecycle.
- The **A12 Developer Tutorials** have been updated and enhanced and are **now available on Github**: <https://github.com/mgm-tp/a12-tutorial-application>
- The **Model Updater** is now obsolete as its functionality has been integrated into the Simple Model Editor (SME) since the 2023.06 release.

SME DOCUMENT MODEL EDITOR INCLUDES

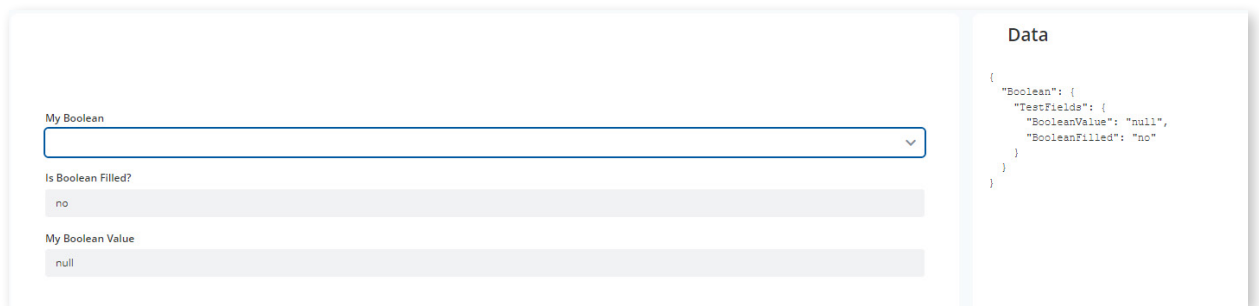
New Features for Include Elements in the Document Model Editor

In the Document Model Editor, you can define Include Elements at any point in the model hierarchy. These elements allow the insertion of another model at a specific location, incorporating all type definitions, fields, groups, rules, and calculations from the inserted model. This simplifies the reuse of existing Document Models without requiring copies. Changes to the original model automatically affect the included models in the main model. However, these elements cannot be directly modified within the main model. The current release now allows setting descriptions, annotations, and index fields for includes. Additionally, rules for validations and calculations can be explicitly excluded, meaning they will not be inherited from the included model.



FORM MODELING

Three-State Logic for Booleans in the Form Engine and Form Model



The Form Engine and Form Model now support a unified three-state logic for Boolean values: undefined, true, and false. This standardization ensures consistency and simplifies usage. Previously, the Form Engine recognized binary values, true and false, per default while the Kernel already supported three

states, leading to unnecessary complexity. The transition phase begins with the A12-Release 2024.06 LTS. Projects can continue using the old method and gradually switch to the three-state logic until 2025.06.

