

A12 Release 2025.06-ext4

Februar 2026

Grundlegende Informationen zu A12
<https://www.a12.ai>

Was ist A12?

A12 ist eine Open Source-Plattform für die Entwicklung von Unternehmensanwendungen in komplexen IT-Landschaften. Sie setzt auf modellbasiertes Software Engineering (MDSE) und erschließt das Low Code-Prinzip für die Welt der Enterprise Software. Als offene Plattform vereinfacht A12 die Integration von Best-of-Breed-Lösungen und den Einsatz von KI auf allen Ebenen. Die Modellierungsumgebung von A12 stellt Werkzeuge bereit, um Teile einer An-

wendung ohne Programmierkenntnisse zu erstellen und als unabhängige Geschäftslogik-Module langfristig zu pflegen. Die Laufzeitplattform von A12 bietet die nötige Flexibilität, um geschäftskritische Applikationen mit professioneller Individualsoftwareentwicklung, KI-Unterstützung und Systemintegration zu voll integrierten Unternehmensanwendungen zu entwickeln.

ADDITIVE MODEL

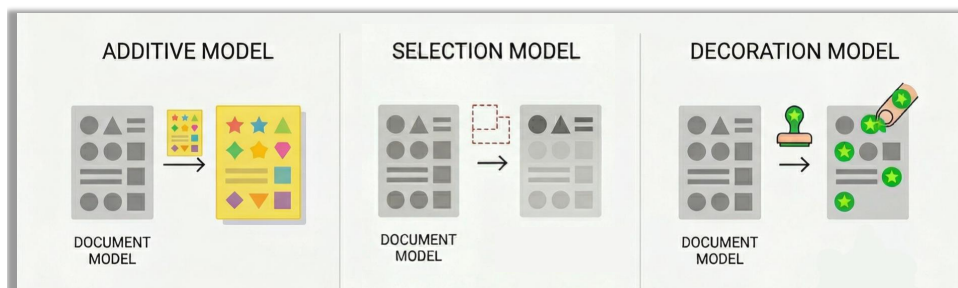
SELECTION MODEL

DECORATION MODEL

Combination Modeling: Geschäftslogik flexibel wiederverwenden

In Geschäftsanwendungen gibt es immer wieder Datenstrukturen, die sehr ähnlich sind und sich nur in bestimmten Aspekten unterscheiden. Um alle wesentlichen Informationen nicht doppelt, sondern nur an einer Stelle zu definieren, bietet A12 schon seit Jahren Mechanismen wie Includes und Type Definitions. Das neu eingeführte Combination Modeling bietet nun zusätzliche Mechanismen, die noch weitergehende Fähigkeiten für eine sauber gekapselte Datenmodellierung ohne Redundanzen mitbringen. Neben der bereits im vorherigen Release eingeführten Möglichkeit, Basismodelle um weitere Aspekte anzureichern (Additive Model),

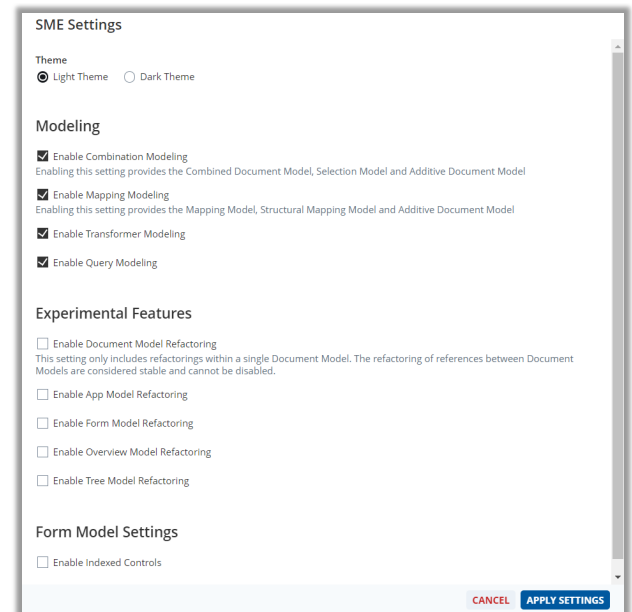
werden jetzt auch zwei weitere Kombinationsschritte unterstützt: das Entfernen nicht benötigter Felder und Gruppen (via eines Selection Models) und die Anreicherung bestimmter Elemente (Decoration Model). Das Additive Model kann man sich wie eine Folie vorstellen, die über ein Basismodell gelegt wird. Das Decoration Model funktioniert eher wie ein Stempel, der an bestimmten Stellen im Modell ergänzt wird. Welche Stellen das sind, bestimmt ein zusätzliches Selection Model. Das Combination Modeling ist bislang noch experimentell, aber nun Feature-complete. Es soll mit dem Release 2026.06 den produktiven Status erreichen.



SME EINSTELLUNGEN

Simple Model Editor führt globale Einstellungen ein

Der Funktionsumfang der A12-Modellierungsumgebung wächst kontinuierlich. Um den Simple Model Editor (SME) als zentrales Modellierungswerkzeug präziser auf die individuellen Bedürfnisse der Nutzer abzustimmen, stehen jetzt globale Einstellungen zur Verfügung. Modellierende können hier beispielsweise zwischen Light- und Dark-Theme wechseln sowie bestimmte Modell-Typen und experimentelle Features aktivieren oder deaktivieren. So können sie Features, die sie im Arbeitsalltag nicht benötigen, gezielt ausblenden und trotz des wachsenden Funktionsumfangs in einer schlanken Nutzeroberfläche arbeiten.



FORM ENGINE MODELLIERUNG

Mehr Kontrolle, weniger Aufwand: flexible Hide Conditions für dynamische Formularsichtbarkeit

Dynamische Formulare müssen häufig Abschnitte je nach Benutzereingabe ein- oder ausblenden – etwa zusätzliche Felder für internationalen Versand bei Angabe einer ausländischen Adresse. Zur Modellierung dieses Verhaltens braucht es einen Mechanismus, der die Sichtbarkeit von Formularelementen an den Wert eines Trigger-Feldes knüpft. Der Simple Model Editor bietet dafür bereits mehrere Mechanismen. Ist ein Feldwert unter bestimmten Bedingungen nicht relevant, kann ein Dependent Field definiert werden, um das Speichern dieses Wertes zu verhindern. Soll ein Formmodellelement wie ein Grid in manchen Fällen ausgeblendet werden, die Daten aber erhalten bleiben, wurde typischerweise eine Dependent Control eingesetzt. Mit

dem neuen Release unterstützen die meisten Formmodell-Elemente zusätzlich eine Hide Condition, die – ähnlich einem Dependent Field – die Auswahl eines Datenfeldes mit dem Ein- oder Ausblenden eines UI-Elements (ähnlich einer Dependent Control) kombiniert und die Flexibilität deutlich erhöht. So können Modellierer Formulare gestalten, in denen Abschnitte basierend auf Benutzereingaben erscheinen oder verschwinden – ohne auf Workaround-Controls zurückgreifen zu müssen. Das Ergebnis: sauberere, besser wartbare Formularstrukturen. Eine Orientierungshilfe zur Wahl zwischen Hide Conditions und anderen Dependency-Mechanismen ist auf [GetA12.com](https://geta12.com) verfügbar.

SME unterstützt neue und erweiterte Refactorings für Document, Overview- und Tree-Models

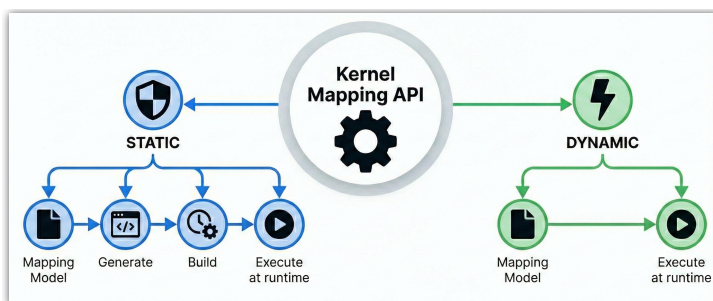
Änderungen an einem Modell können Auswirkungen auf andere Stellen im Modellgeflecht haben. Das Löschen eines Feldes in einem Document Model kann beispielsweise verwaiste Berechnungsregeln oder fehlerhafte Referenzen in Form-Models hinterlassen. Die Refactoring-Funktionen des SME weisen die User automatisch auf solche Auswirkungen hin und helfen ihnen, sie zu beheben. Mit dem Release 2025.06-

ext4 wird der Refactoring-Support deutlich ausgebaut. Bislang gab es die Unterstützung für Document-, Form- und Application-Models. Jetzt ist sie für Document-Models erweitert und als experimentelles Feature auch für Overview- und Tree-Models verfügbar. Nach Aktivierung über die globalen SME-Einstellungen erscheinen Refactoring-Dialoge automatisch, wenn eine Modelländerung Folgeeffekte hat.

Mapping API des A12 Kernel führt Code-Generierung zur Laufzeit ein

Die Mapping API des A12 Kernels erreicht einen neuen Reifegrad. Bisher unterstützte die API als experimentelles Feature das Mapping mit statischem Java-Code, der aus Mapping-Modellen generiert wurde. Dabei muss der Code zunächst zur Build-Zeit generiert, kompiliert und eingebunden werden, bevor er in der Anwendung ausgeführt werden kann.

Diese statische Variante ist nun ausgereift und nicht mehr experimentell. Gleichzeitig führt die API als neues experimentelles Feature die dynamische Variante ein. Damit ist es jetzt möglich, die Mapping-Funktionalität zu verwenden, bei der Groovy Code zur Laufzeit generiert und direkt ausgeführt wird.



Datensparsamkeit mit weniger persistierten Metadaten

A12 setzt auf eine dokumentenorientierte Datenhaltung, bei der neben den Datensätzen für Dokumente auch Metadaten persistiert werden. Bislang waren alle der folgenden sieben Metadaten-Felder obligatorisch: `docRef`, `modelReference`, `creator`, `modifier`, `createdAt`, `modifiedAt` und `modelVersion`. Jetzt ist es möglich, diese Felder durch die Anpassung des Metadaten-Modells zu reduzieren. Nur `docRef` und `modelReference` sind unverzichtbar. Nicht gewünschte Felder können nun von der

Persistierung ausgeschlossen werden. A12 bietet damit neue Optionen für Datensparsamkeit. In manchen Fällen ist das neue Feature auch entscheidend für eine einfache DSGVO-Compliance, da die Metadaten-Felder `creator` und `modifier` personenbezogene Daten enthalten können. Im Standard besteht das Metadaten-Modell aus allen sieben Feldern und ist damit rückwärtskompatibel. Bei Anpassungen des Modells müssen bestehende Daten migriert werden.

UAA KEYCLOAK OIDC

Mehr Flexibilität bei Authentifizierung via OIDC

UAA unterstützt jetzt benutzerdefinierte Search-/Query-Parameter beim OIDC-Login. Damit können Clients zusätzliche Kontextdaten direkt an die Keycloak-Login-Seite übergeben und so das Verhalten und die Darstellung der Anmeldeseite steuern. Der Parameter `ui_locales=de` sorgt zum Beispiel dafür, dass das Login-UI auf Deutsch angezeigt wird. `acr_values=mfa` kann ein stärkeres Authentifizierungsniveau wie die Multifaktorauthentifizierung anstoßen. Neben den neu unterstützten Login-Parametern lässt sich jetzt auch die *Current User URL* server- und clientseitig konfigurieren, was die Integration in unterschiedliche Deployment-Topologien erleichtert.

➤ Using `uaaClient`

```
uaaClient.login({
  extraQueryParams: {
    login_hint: "user@example.com",
    ui_locales: "de",
    acr_values: "mfa"
  }
});
```

➤ Using `UaaActions`

```
reduxStore.dispatch(UaaActions.loggingInOIDC(
{
  extraQueryParams: {
    login_hint: "user@example.com",
    ui_locales: "de",
    acr_values: "mfa"
  }
}
));
```

PROJECT TEMPLATE E2E-TESTS

E2E-Tests: A12 Projekt-Template ersetzt Cypress durch Playwright

Für Ende-zu-Ende Tests (E2E) setzt das A12 Projekt Template jetzt auf die Automatisierungsbibliothek Playwright an Stelle von Cypress. Alle Cypress-Tests wurden nach Playwright migriert – sowohl im Basis-Template, als auch den Varianten mit Workflows und dem Notification Center. Das Kapitel „End-to-End Testing“ in der Dokumentation des Projekt Templates fast kurz zusammen, wie die Tests der Template-Anwendung um projektspezifische Tests erweitert werden

können. Ausgangspunkt für die Entscheidung zum Wechsel des Testframeworks war eine umfassende Evaluierung beider Open Source-Produkte. Playwright punktete sowohl bei der Entwickler- und Debugging-Erfahrung, als auch bei der Performance. Lokal liefen die Playwright-Tests mehr als 60% und in der Jenkins CI-Umgebung 36,9% (1 Worker) bzw. 57,1% (3 Worker) schneller.



Playwright



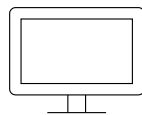
Kurz & Knapp

- Das A12 **Kalender-Widget** unterstützt jetzt mit **Infinite Scrolling** eine **nahtlose Navigation durch Monate und Jahre** - und bietet damit eine noch **bessere Nutzererfahrung**.
- Der **A12 Kernel** stellt **erweiterte Fehlermeldungen** zur **Validierung von Queries** bereit. Dies hilft dem **Entwicklungsteam** dabei, Konfigurations- und Modellierungsprobleme **schneller und komfortabler zu identifizieren**.

Mehr Optionen zur Definition des responsiven Verhaltens durch die Media Query Elemente

Moderne Websites müssen ihre Informationen mit hoher visueller Klarheit für Nutzer auf einer Vielzahl von Endgeräten darstellen können – seien es Mobiltelefone, Tablets oder Desktop-Computer. Die neue Version A12 2025.06-ext4 stellt Modellierern von Content-Modellen hierfür ein neues Werkzeug zur Verfügung, um diese Herausforderung zu adressieren: das Media-Query-Element. Ein Media-Query-Element dient als Container für andere UI-Elemente, wie etwa Tabellen oder Textfelder. Zusätzlich zu diesen Elementen enthält

es eine Definition – die Media Query –, die die Bedingungen beschreibt, unter denen die Elemente dem Endnutzer angezeigt werden. Auf diese Weise kann ein Modellierer festlegen, wie Informationen abhängig von der Bildschirmgröße dargestellt werden sollen, dedizierte UI-Bereiche innerhalb von Media-Query-Elementen erstellen und die entsprechenden Abfragen definieren, sodass Nutzer stets das für ihr Gerät passende Layout sehen.



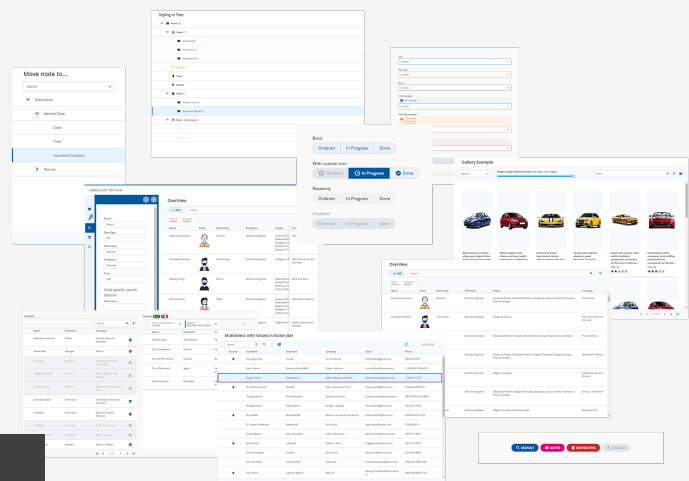
Country	City	Price	
Germany	Frankfurt	300	 
France	Paris	400	 
America	New York	101	 



Country: Germany City: Frankfurt Price: 300
Country: France City: Paris Price: 400
Country: America City: New York Price: 101

A12 Widgets

A12 Widgets ist eine umfassende Sammlung von vorgefertigten Komponenten, die eine ansprechende und intuitive Benutzererfahrung ermöglichen.



A12-Widget-Showcase
<https://www.mgm-tp.com/a12.htmlshowcase/>