Component Reuse

URL: https://www.mendix.com/evaluation-guide/app-lifecycle/reuse

1 How Does Mendix Enable Component Reuse?

One important aspect to consider when building multiple applications is whether you can reuse components of your app models across your applications. The Mendix Platform enables such reuse on different levels, both within a single app and across multiple apps. This allows you to achieve the following:

- Create consistency across your app landscape - Reusing UI elements, logic, and data will ensure your applications look and behave consistently. This will both improve the usability of your applications and ensure the correctness of your information.
- Increase your productivity - Reusability means you can reuse proven components in multiple locations, reducing the time to implement and test similar behavior.

The sections below will discuss the reuse options you have in the Mendix Platform for different aspects of your application models.

2 How Can I Reuse Components Across Mendix Applications?

Reuse across applications can occur in the UI, logic, and data. Dive into the sections below for details.

2.1 User Interface

UI reuse is divided between components and styling, centralized theming, and referencing user flows from other apps.

2.1.1 Components & Styling

With the Mendix Atlas UI framework, developers can rapidly build consistent user interfaces through reuse while maintaining control and flexibility per app. With Atlas UI, it is possible to create a UI resource package that combines reusable elements, templates, and theming to reflect the brand guidelines of your company.

The philosophy behind Atlas UI is that designers can create a company design language and implement best practices that developers can easily reuse. This helps to make designers more productive in that they only need to design UI components once, and it makes developers more productive in that they can focus on building functionality. The Mendix Platform uses several concepts for creating consistently great looking apps that can be encapsulated in the UI resources package. For an explanation of these concepts, see UI Design.

You can share your UI resources through the Mendix private App Store or integrate them in company starter apps so that other developers can quickly reuse the design language when starting a new app project. This helps to increase development velocity and create consistency across apps. New updates to the design language can be easily published via your company's private App Store.

For each application, it is possible to extend the design language in order to achieve full control over the look and feel while retaining consistency. Developers can extend SASS files to include app-specific styling with specific layouts, page templates, building blocks, and design properties.

2.1.2 Centralized Theming

In addition to this, it is also possible to share theming across applications by referencing a centrally hosted theme. By hosting the theme in a central Mendix app or in a content delivery network (CDN), your company can make changes on one location that are applied to multiple apps. It is also possible to combine a central theme with an app-specific theme, providing both reuse and control.

2.1.3 Referencing User Flows from Other Apps

Mendix enables deep-linking from one app to another app. In turn, this enables developers to separately develop user interaction flows that can be reused from other apps.

Another benefit here is that users are can be automatically logged in via single sign-on (SSO) technologies like SAML, OAuth, or OpenID Connect.
2.2 Logic

For reusing logic across different apps, there are several different methods:

- **Keep your logic in a central place** - The best way to reuse logic across different applications is to not distribute it across all your apps. This can be achieved by exposing your microflow logic through web services or REST APIs.
- **Export & import app modules** - This method is similar to using libraries in traditional development platforms. Next to logic, modules can also contain reusable data and UI components. The Mendix App Store enables you to publish your modules into a central repository for easy reuse by other developers.
- **Reuse third-party Java libraries via modules** - With this method, you can include Java-based microflow actions in a module and include any Java library you need. This enables you to easily reuse a wide range of Java libraries in your Mendix applications in a way that does not require any Java knowledge from the developers using your modules.

All of these methods provide you with the benefit of having to implement and test your logic only once, which will help to improve productivity and consistency.

2.3 Data

Your approach to reusing a data model across different applications will depend on your requirements. Sometimes you just need to reuse the model, but every app will have its own data. Alternatively, you may need to reuse the model in addition to the actual data across apps.

Mendix supports both reusing a data model and the actual data across apps:

- **Reusing a data model while retaining different data in every app** - To accomplish this, you can export your domain model in a module, as described for exporting & importing app modules in the Logic section above.
- **Reusing a data model and the data** - To accomplish this, creating an API in a central application is a better approach. This can be achieved through web, REST, or OData services. Web services are a good choice when sharing data with legacy systems. REST and OData services both have the benefit of being better suited for sharing data through defining a standard set of operations on the data. REST will enable you to share data with other Mendix apps as well as with most other modern application development platforms. OData has the additional benefit of being well supported as a way to reuse data in business intelligence and reporting tools.

3 How Can I Reuse Components Within Mendix Applications?

Reuse within applications also occurs in the UI, logic, and data, as presented in the sections below.

3.1 User Interface

Mendix applies a UI component-based approach based on several concepts for supporting reuse within apps while still enabling extensibility per use case. For details on these concepts, see UI Design.

Mendix offers layouts and snippets that can be used as references, meaning that if a layout or snippet is changed, this change is propagated globally in the project. This workflow optimizes reuse, but in some cases, propagating a change may not be preferred if some pages or parts of the app are intentionally designed differently. To support these use cases, Mendix also offers page templates and building blocks for a template approach. These templates serve as design inspiration while helping to increase productivity by providing app pages (or parts of pages) that are already created. After a template is used, changes are only applied on that location, so unforeseen side effects can be avoided during the development process.

This combination of using layouts, snippets, and templates empowers developers with the ability to reference and update globally while maintaining the freedom to make changes locally in their app projects.

3.2 Logic

To reuse pieces of your logic within an app, you can use these methods:

- **Extract reusable steps from a microflow into a new microflow** - This extraction is as easy as selecting the actions you want to reuse and entering a name for your new microflow. In the originating microflow, the old steps will automatically be replaced by a call to your new microflow. When extracting your new microflow, Mendix will automatically determine what parameters are required for your new microflow.
- **Create new microflow activities using Java actions** - After defining a new Java action and writing the implementation with a regular Java development environment, you can use the activity in any microflow. You can provide an easy-to-understand name, category, and icon that will be used in the microflow editor toolbox in Mendix Studio Pro. Other users will not see any difference between using Mendix-provided activities and your custom activities.
meaning, there is no learning curve for reusing your custom activities.

3.3 Data

Mendix domain models enable reusing the definition of an entity for similar entities through inheritance. This means you can define the attributes that need to be defined on multiple entities once, and then inherit these attributes in another entity. This works best when the entity inheriting the attributes is a specialization of the original entity (for example, a delivery address entity and billing address entity can both inherit from a generic address entity).