< Blog

Best Practices For Managing Application-Level Security with Mendix

by Daniela Field

Mendix applications are implemented by a large variety of companies to support numerous and varied business processes. All these different Mendix users share the critical need for their applications to be secure and accessible.

The Mendix App Platform handles key security aspects out of box. For example, measures against front-end security threats, such as Cross Site Scripting, and server-side security threats, such as SQL Injection and Code Execution, are provided by the platform.

Mendix developers do not need to take these technical security aspects into consideration when building Mendix apps, as the platform handles this as a service. Obviously, this does not mean that developers do not have to consider

Best Practices For Managing Application-Level Security with Mendix

Related posts

The Basics

Application security has never been easier to manage within the Mendix App Platform. With just a few clicks, users can see only their own relevant information and specific parts of the applications. Furthermore, with Mendix, security can be as granular as the business users need it to be.

As you may know, building an app in Mendix takes place in a project. In the project there are 3 types of security levels:

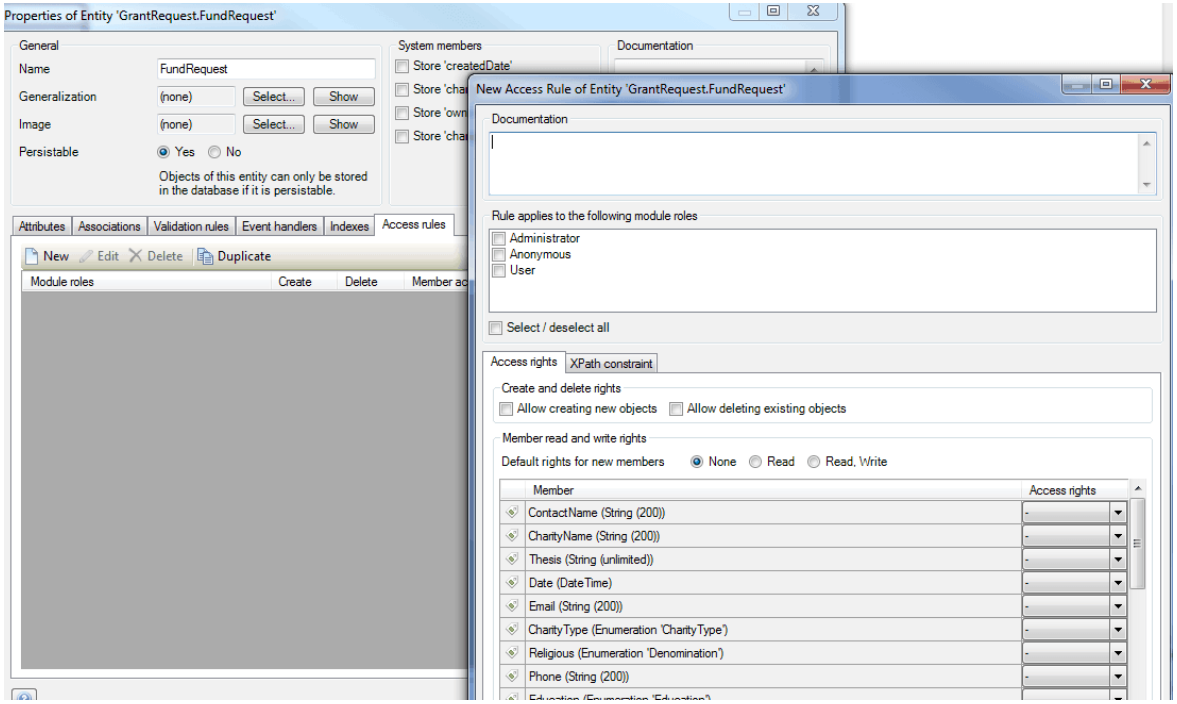
- **Off** – When the security is off, there are no login requirements and the application is open
- **Prototype/ demo** – As the name suggests, this security level is only used for demo purposes or prototypes and the page and microflow access is necessary
- **Production** – If you want to deploy any app into the Mendix Cloud or move it to production; the production level is required and necessary for any enterprise application that is created and used.

The Project security oversees the modules security and you configure the user roles that you wish to assign to the end users. Each user role will have a number of

Best Practices For Managing Application-Level Security with Mendix

Domain First

When developing an enterprise application, always start with production level security. Once you choose the production level security, a new tab shows up in the entities called the Access Rules.



Best Practices For Managing Application-Level Security with Mendix

Subscribe

For each user role, we can grant different access levels. For example, we can allow the anonymous users to create a new grant or be able to delete their own grant. For anonymous users, it is always important to add a restriction using XPath constraint, which allows them to edit their own grant only.

We can go down into more details and allow the user roles to only read or write for a specific attribute.

It is best practice to start applying security at the domain model for various reasons highlighted below:

1. Data security settings supersedes UI and microflow security
2. If you do not allow a specific user role to create or delete objects, then the platform is designed to automatically not show the delete or create buttons in the user interface

Always add XPath constraints to determine which level of security should be applied to each object. For example, a business analyst should be able to see only the reports they created instead of the reports created by other company business analysts.

Administrator Role

users, managing email settings, and monitoring an application. An administrator role should NOT have access to the application at large.

For example, if your company has sensitive data such as finance transactions in a financial app, or physician and patient information in a clinical trial- or healthcare app, the administrator should not be able to view that information.

Implement Security As You Go

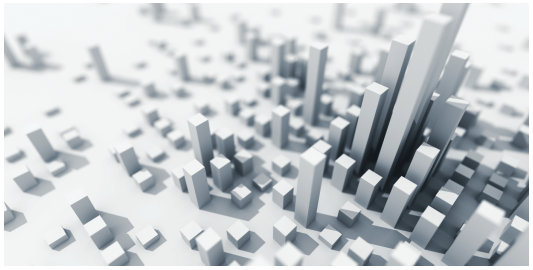
As you are building your application, it is important to keep security in the back of your mind. Always think of who should have access to what. If you are always implementing the security as you are building the application, it will save time when testing. Furthermore, as the application gets more complicated, it can take a while to go back into each page, microflow and entity and determine the correct access and permissions. Again as you are building the domain model and your database – add the security.

As you build pages and microflows – add the security. Each page and microflow will have the **Visible for** property. Adding the user roles here allows for the users with the specific userrole to see the page or microflow. If the page does not have a user role assigned, the Mendix Business Modeler will let you know in the Errors log.

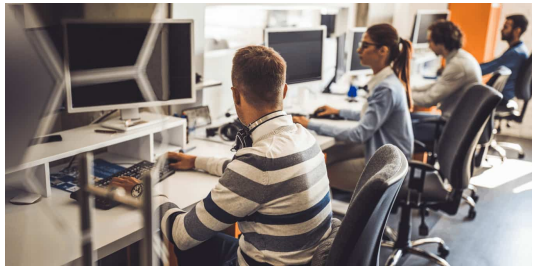
August 25, 2014

Trending articles

Low-Code Principle #2: Collaboration, Where Innovation Happens



Low-Code Principle #1: Model-Driven Development, The Most Important



Introducing the Low-Code Manifesto

Best Practices For Managing Application-Level Security with Mendix

Subscribe

The Platform

Why Mendix?

Made with Mendix™

Community

Resources

Newsroom

Our Company

Best Practices For Managing Application-Level Security with Mendix

Subscribe