3 Tips for a Performant User Interface

Nothing annoys a user more than a slow application. And unfortunately, no developer (not even a Mendix developer) is immune to this problem. But there are ways to boost application performance and it's easy to manage this process in the Mendix App Platform.
In my next Expert Webinar on performance optimization, I'll focus on preventing user frustration by answering the question: How does UI impact performance?
To help get the community excited, I've included three tips below to ensure your User Interface remains as fast as possible. For my other tips, you'll have to join the webinar! The live event will take place on June 26, 2015 (@ 10am ET / 16:00 CET), but the recording will be available on our website soon after.

1. Limit the number of objects showing on a page

When it comes to UI design – less is more. Show the users the right information only when it is needed. I've seen new Mendix developers build a single page that includes over 100 objects. But there's no reason to do this. Why would a user ever need to see over 100 records at one time? This amount of information overloads the user and slows down the system.
Retrieving and rendering that volume of information takes time even for the fastest internet connection. Ideally, users shouldn't have to waste time scrolling through a page to find what they need. Instead of bogging the system down with one massive page, consider breaking up the information across multiple pages.

2. Avoid complicated XPath constraints

Retrieving information from the database takes time. The more complex your XPath, the longer the retrieval will take. And while we just released a product update that increases performance, there's always additional components to adjust per each unique application. You can avoid complicated XPath constraints in your widgets by using Microflows to populate your datagrids. To learn about XPath constraint improvements, check out the articles here and here.
Microflows are more straightforward to optimize compared with direct retrievals from the database with complicated XPath constraints. It's true that optimizing Microflows takes a bit of practice and is more work than using the Connector dock to add datagrids within dataviews. The improvement on performance, however, will be worth it.

3. Avoid overloading your page with widgets

Rendering a JavaScript widget takes time and resource. And the more widgets present on a page, the longer it will take to load. If you are seeing performance issues on a page, review the number of widgets you are using.
Ask yourself the following questions:

- Do I really need 20+ widgets on this page?
- Do I need to have tables within tables within tables?

Consider replacing your widgets with snippets for ease of use and faster load times.