



HTML5 user interface in Industrial Applications

BY LUIGI DE BERNARDINI | SUN MAR 22 2015

I'm very proud to publish this post written from one of my employees. Alberto is a very good software developer (and an amazing guitar player – click [here](#) to visit his web site), skilled and passionate. He mixes tech skills and the ability to look forward and imagine how new technologies can be applied in our daily activity to bring value to our clients.

“I was recently given the task to build a responsive web application that would be accessible from the mobile barcode readers used to track items in two standard warehouses. The way this had been done in the past had been to develop a program with custom mobile layout, let it run on a PC, and every client had to connect to it through a TS (terminal server) session. This can be quite expensive, especially considering the cost of TS licenses. The more clients that need to connect, the higher the required fees are in terms of licenses—and the two warehouses this application was being designed for needed about 70 total mobile devices each. Plus, the original application was developed for a desktop PC. Even though its layout was optimized for mobile, the end

user experience was poor.

Working with the latest in web development technologies, however, allows for the easy creation of a beautiful and responsive HTML5 user interface in industrial applications through new CSS3 style definitions. This led me to consider the idea of using HTML5 for industrial web applications while working on an internal project. Here is what I chose for each component needed to build an industrial web application, and why.

Working with the latest in web development technologies allows for the easy creation of a beautiful and responsive user interface.

1. Web Server. A web server makes web pages available to many devices. NodeJS provides a simple but complete web server written in Javascript. For development you can use Visual Studio. It comes with NodeJS Tools for Visual Studio, which integrates perfectly with the Visual Studio environment.

2. Database Processing. An Interface is needed for an SQL database and for the server to process the results set. NodeJS offers modules that developers can install on their project with NodeJS Tools. One of these modules is the SQL driver for any SQL database provided by Microsoft. The Javascript language lets you create objects and arrays ready for broadcast to clients.

3. Communication Channel. A protocol to transmit data between client and server is required, and Web Socket is a great way to allow such bidirectional communication. Applications may need to send a request to the server (NodeJS) and receive a response. Other messages may need to be broadcast, without sending any request (for example a timer to upgrade a meter). Web Socket is also useful for identifying warehouse operators using IP addresses, because the Web Socket connection requires a handshake. NodeJS provides a module called Socket.IO to implement this communication channel.

4. Responsive User Interface. A set of frameworks to integrate dynamic variables into HTML pages allows for less writing of code and more functionality. These are my favorite client side frameworks:

- *AngularJS* is a Javascript library that allows web developers to insert variables and structures into the HTML page to perform various functions such as ordering fields. I believe this represents a turning point in web page development, because it replicates the way all software developers think.
- *JQuery* is now used on 90 percent of websites to provide clean access to and manipulation of the DOM. It is a Javascript library that allows web developers to handle events and animate elements of a webpage.
- *Bootstrap* originated as a set of CSS styles to build responsive websites with modern layout. Many web developers use it because of its grid system and simple layout. Its integration with platforms like WordPress makes it easily accessible and commonly used.

With the appropriate integration of these three libraries, a modern and maintainable client side can be developed. AngularJS is developed by Google, so it is easy to find support and suggestions from other people using it for their projects. This can also be said for the other libraries I use. In my experience, the quality of support and the strength of a developing community are very important.

Turning now to the server side, NodeJS appears as a command prompt in DOS. This may seem old school, but it makes it easy to develop in Javascript, and allows importation of various kinds of modules. Observing operations in a DOS window is useful for monitoring what tasks the server is performing in real time. If you need a history log, you can implement a simple function to create a .txt file to record what the server is doing.

Another benefit is that you don't need a server to run it, you can just use a desktop PC and manage it like a process. This is useful in case of error, because you can automatically force a restart. More experienced users can use the extension provided by NodeJS to install it on an IIS server, letting IIS supply the web page while NodeJS processes everything else, as in the case of Web Socket requests.

Using the simple tools mentioned above, it is possible to develop each of the four separate components of a custom web application. In a very short period of time, I was able to develop a web application with clean client and server sides. The client was so satisfied that they have asked us to develop another new module in the same way."

Alberto Ziliotto is a software developer for [Autoware](#), a [CSIA](#) certified member based in Vicenza, Italy.

Published on [Automation World](#) – Mar. 16th 2015