



Xtext & Xtend

Xtext

Randy Paredis



What is Xtext

- Open-source
- Grammar driven
- Continuous Integration
- Based at Eclipse.org

When should I use Xtext LSP instead of a native editor integration?

As of today if you are looking for an Eclipse Plug-in my answer is clearly, go with the traditional Xtext Eclipse Plug-in. With Eclipse LSP4E there is Eclipse support for language servers, but it is not even close to what our native Eclipse support does. I also doubt that this will change any time in the future. **The native Eclipse integration of Xtext is here to stay.**

For IntelliJ IDEA the situation is different. Neither the Xtext integration has been updated with the last release, nor has JetBrains yet started to work on LSP support. The code for the IDEA integration is quite extensive and deep. So deep that we get regularly broken because we use non-public API. Since the demand for IDEA integration is not high, maintaining it doesn't make sense to us. That is why I asked JetBrains to work on LSP integration last year already. So far they don't seem to be convinced, but you could add your 2cents or +1 to **this ticket** if you think LSP would be a good IDEA.

For the rest of this post, I want to talk about Xtext Web and why you should not use it anymore and prefer the LSP integration instead.

The Xtext Web support was our first attempt to generalize language features over multiple editors. At that time we only abstracted over the web editors Ace, CodeMirror and Eclipse Orion (the editor widget, not the IDE). We did it over a REST interface and focussed on single code editors, only. The LSP integration works with any editor supporting it and while Eclipse Orion is still working on supporting it, the Monaco code editor from Microsoft **fully supports it already**. So here are my four reasons why you should use LSP for web applications:



Demo

In Eclipse Photon



Next Steps

- Creating a grammar for Bmod
- Export it to html and show results
- Try getting the web-editor to work