

F# Type Providers

...

What are Type Providers

Type Providers are a language feature first introduced in F#3.0:

An F# type provider is a component that provides types, properties, and methods for use in your program. Type providers are a significant part of F# 3.0 support for information-rich programming. <http://bit.ly/2lQJEMW>

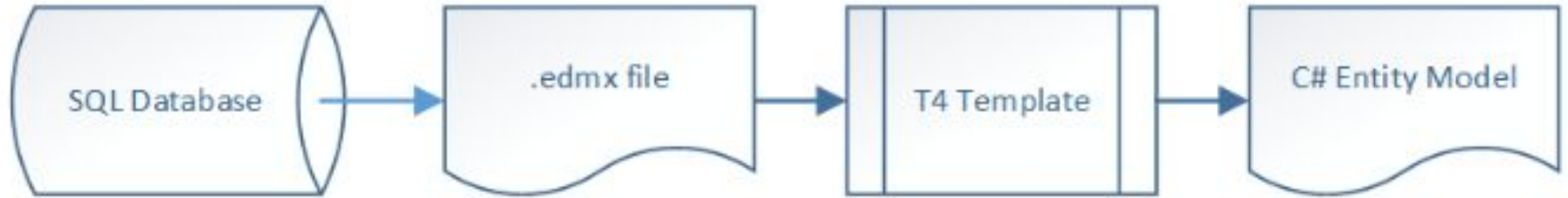
What are Type Providers

A magic thing that saves you from writing hundreds of lines of boilerplate code and updates itself when the data source changes.

Why use Type Providers

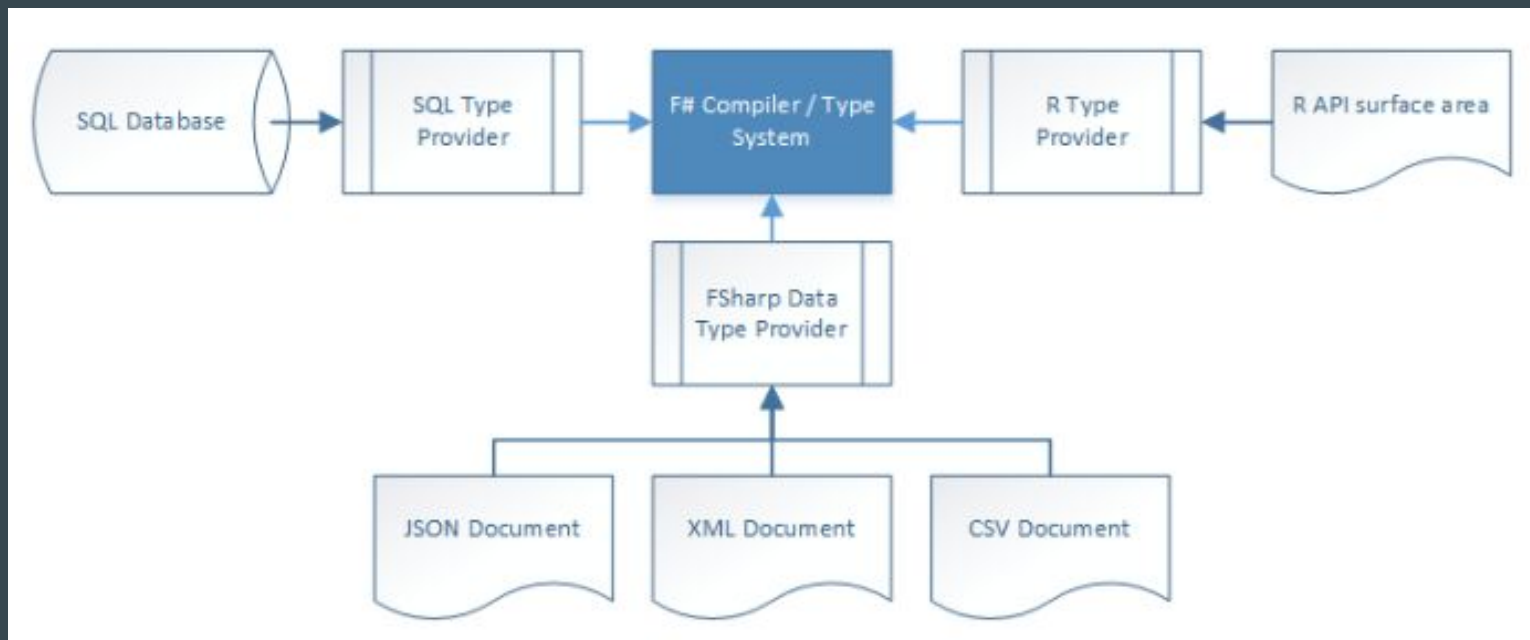
- Statically typed
- Updates with the data source (usually)
- Updates at compile time
- Baked into the language

Understanding Type Providers



Entity Framework Database-First code generation process

Understanding Type Providers



A set of F# Type Providers with supported data sources

Building your own Type Provider (briefly)

- Use the Type Provider starter pack <http://bit.ly/2lnNAkX>
- Check out the pluralsight course Building F# Type Providers
- Two forms of type providers
 - Erased
 - Generated

Building your own Type Provider (briefly)

- Erased
 - Type provider replaces references to the provided type at compile time
 - Not backed by .NET types
 - Erased types won't work with reflection
 - Truly a F# compiler feature

Building your own Type Provider (briefly)

- Generated
 - Type provider creates real types based on the provided type definition
 - Backed by an on-disk assembly
 - Compatible with reflection

Building your own Type Provider (briefly)

- Code Quotations
 - Metaprogramming mechanism similar to expression trees in LINQ
 - Two forms
 - Typed - carries type information e.g. `Expr<'T>`
 - Untyped - carries no type information e.g. `Expr`

Working with Type Providers

Demo Time!

Links

Type Provider starter pack <http://bit.ly/2lnNAkX>

Learn Fsharp <http://bit.ly/2mrNf1L>

MSDN Docs <http://bit.ly/2lQJEMW>

Thank You

@AndreDublin
github.com/andredublin