

Contents

- [Launch the app](#)
- [Watch the videos](#)
- [Model](#)
- [Domain Services](#)
- [Common classes](#)
- [Main project](#)
- [Screen harness](#)
 - [Video](#)
- [Web project](#)

[Punch](#) ships with a reference application called **TempHire**, a line-of-business application for an imaginary temporary hiring agency. TempHire is available as Silverlight and WPF solutions, sharing most of the code between the two solutions. [Download](#) the very latest Source Code to stay up-to-date as the application evolves. TempHire demonstrates a best practice architecture and design for real world enterprise and line-of-business applications. In order to keep the scope to a manageable size, TempHire currently only implements the resource management module, but one could easily imagine other modules like a scheduler, customer management etc.

Despite only having a single module, TempHire demonstrates the common pieces that make up a real application, including login, view and view model composition for loosely-coupled, reusable UI components, multi-screen navigation, eventing, validation, concurrency checking, multiple EntityManagers for sandbox editing and cache management, etc.

TempHire is written in Code-First style and utilizes multiple models to separate concerns such as security.

Launch the app

The Silverlight version of TempHire can be seen and experienced online and the source code is included in the Punch Samples folder.

Demo URL: apps.ideablade.com/TempHire

Watch the videos

- The [Punch YouTube playlist](#) contains all of the Punch videos listed below.
- [Punch: TempHire overview](#)
- [Punch: TempHire in Action](#)
- [Punch: Building TempHire](#)
- [Punch: TempHire solution structure \(WPF\)](#)
- [Punch: TempHire solution structure \(Silverlight\)](#)
- [Punch: TempHire domain model](#)
- [Punch: TempHire domain services](#)
- [Punch: TempHire main project](#)
- [Punch: TempHire main project \(the shell\)](#)
- [Punch: TempHire main project \(view composition\)](#)
- [Punch: TempHire main project \(UI and miscellaneous components\)](#)

Model

TempHire contains two separate code-first models. DomainModel is the main model for all the application data and the Security model is a very simple model for user authentication against a user table in a database. Separating the security concern from the main application concerns like this, gives us the flexibility to later change to something like Active Directory or Windows security much easier than if the security was baked into the main model.

The two models are not only independent in the code, but they are also mapped to two different physical databases, which will be automatically created for you using the embedded SQL Server Compact.

Domain Services

Contains the [Unit of Work, Repositories, Entity Factories and Business Services](#) used by the ViewModels. This layer implements the business processes and provides [persistence ignorance](#) to the UI.

Common classes

Common classes and functionality that is most likely going to be used across multiple modules, which may live in their own assemblies, are broken out into a Common project.

Main project

The main project is TempHire, which contains all the views and view models, along with the entire UI logic. The UI is decomposed into smaller views and view models that then get composed together into what we ultimately see on the glass. This approach maintains flexibility in how the UI is arranged together and allows for many different orchestration scenarios. It also facilitates the distribution of work among many developers in a team. Each developer can independently work on an entire screen or parts thereof.

Screen harness

A completely optional project is the screen harness to increase developer productivity. Learn more about the [screen harness in Punch](#) or see the TempHire harness [in action](#).

Video

Web project

Finally, the web project implements the server-side of TempHire for Silverlight or an N-tier deployment of the WPF version. It hosts the Silverlight application and the DevForce Application Server. The two landing pages either launch the Silverlight main application (default.aspx) or the Silverlight screen harness (harness.aspx)