Sponsors and Partners

Strategic Sponsors

adform

LUXOFT

Gold Sponsors

DevExpress®

Roche

KMD
KMDPOLAND.PL

VOLVO

Silver Sponsors

it KONTRAKT®

Fast Reports
Reporting must be Fast!

JCommerce

JETBRAINS THE DRIVE TO DEVELOP

labs
software specialists

Millennium
bank

.NET DEVELOPER DAYS

Functional, Reactive Web Abstractions for .NET

Adam Granicz, @granicz

You are in good company

100k+ downloads
60+ mainstream JavaScript libraries (“extensions”)

80+ talks in 35+ cities in 25+ countries

http://try.websharper.com http://websharper.com

WebSharper

WebSharper

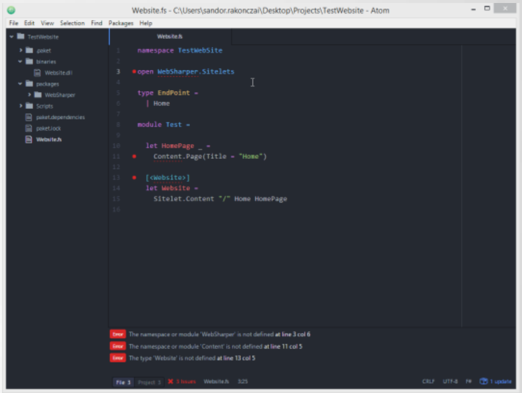
Open source at:

http://github.com/IntelliFactory/websharper

Getting WebSharper

- Downloads for Visual Studio and Xamarin Studio
- Atom via plugin
- In an online IDE – cloudsharper.com
- Using yeoman (generator-fsharp)

```
npm install –g yo
npm install –g generator-fsharp
yo fsharp
```


Website.fs - C:\Users\sandor.rakonczai\Desktop\Projects\TestWebsite - Atom
File Edit View Selection Find Packages Help
TestWebsite
.paket
binaries
Website.dll
packages
WebSharper
Scripts
paket.dependencies
paket.lock
Website.fs

```
namespace TestWebSite

open WebSharper.Sitelets

type EndPoint =
  | Home

module Test =

  let HomePage _ =
    Content.Page(Title = "Home")

  [<Website>]
  let Website =
    Sitelet.Content "/" Home HomePage
```

```
Error The namespace or module 'WebSharper' is not defined at line 3 col 6
Error The namespace or module 'Content' is not defined at line 11 col 5
Error The type 'Website' is not defined at line 13 col 5
```

Project templates

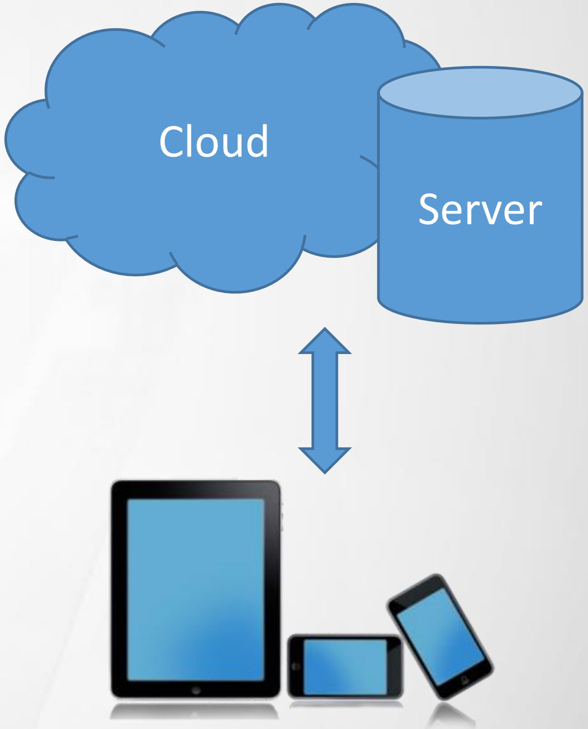
- Single-Page Applications (SPAs) - client-only
- Client-Server Applications - sitelet-based
- HTML Applications - client only, sitelet-based

`http://websharper.com/docs/templates`

Bridging the language mismatch

```
open WebSharper
```

```
module Server =

    [<Rpc>]
    let MyServerFunction(...) = ...

module Client =

    [<JavaScript>]

    let MyClientFunction(...) =

        ...

        let v = MyServerFunction(...)

        ...
```



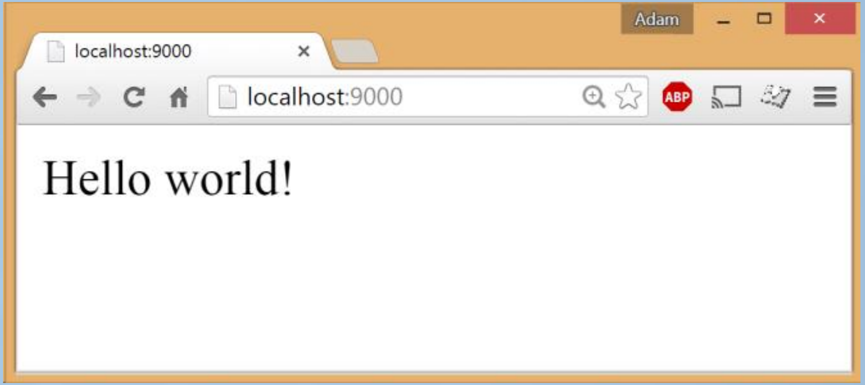
WebSharper

Composable functional programming abstractions

1. Sitelets: web applications
2. Pagelets: dynamic markup and behavior
3. UI.Next: reactive DOM and dynamic dataflow
4. Formlets: complex and dependent web forms
5. Flowlets: sequences of user forms
6. Piglets: formlets on steroids: UIs for any device

Sitelet microservices

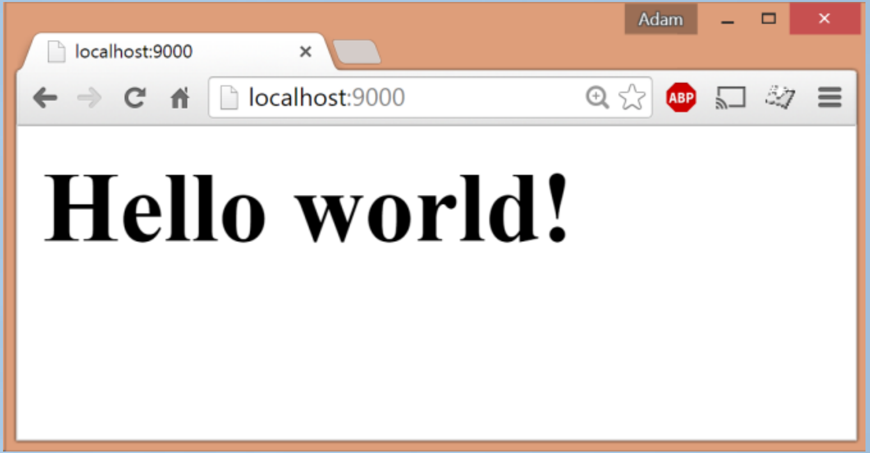
```
module MyApplication

open WebSharper
open WebSharper.Sitelets

[<Website>]
let Main =
    Application.Text (fun ctx ->
        "Hello World!")
```



Hello world!

Sitelet microservices

```
module MyApplication

open WebSharper
open WebSharper.Sitelets
open WebSharper.UI.Next.Html
open WebSharper.UI.Next.Server

[<Website>]
let Main =
    Application.SinglePage (fun ctx ->
        Content.Page(h1 [text "Hello World!"]))
```



Hello world!

Sitelet microservices

```
type EndPoint = int


[<Website>]
let Main =
    Sitelet.Infer (fun ctx (endpoint: EndPoint) ->
        match endpoint with
        | i -> Content.Text (string (i*i))
    )
```

HTML and other responses

- Plain text using `Content.Text`
  - `Content.Text “Hello World!”`
- JSON using `Content.Json`

  -
    ```
    type Person = { First: string; Last: string; Age: int }

    Content.Json { First="John"; Last="Smith"; Age=30 }
    ```
- Files using `Content.File`

  -
    ```
    Content.File("../../Main.fs",
                 AllowOutsideRootFolder=true,
                 ContentType="text/plain")
    ```
- Error codes, etc.

Sitelet endpoints

Endpoint Type	Sample Request	Parsed Request
`Int`	`/12`	`12`
`Float`	`/12.34`	`12.34`
`String`	`/abc1234`	`"abc1234"`
`System.Net.HttpStatusCode`	`/200`	`HttpStatusCode.OK`
`System.DateTime`	`/2015-08-24-12.55.14`	`System.DateTime(2015,8,24,12,55,14)`
`string * int`	`/abc/1234`	`("abc", 1234)`
`{ Name: string; Age: int }`	`/john/12`	`{ Name="John"; Age=12 }`

`string option`	`/None`
	`/Some/abc`
		`None`
		`Some "abc"`

`int list`
`float list`
`string list`
	`/2/1/2`
	`/2/1.1/2.2`
	`/2/abc/1234`
		`[1; 2]`
		`[1.1; 2.2]`
		`["abc"; "1234"]`

`int array`
`float array`
`string array`
	`/2/1/2`
	`/2/1.1/2.2`
	`/2/abc/1234`
		`[|1; 2|]`
		`[|1.1; 2.2|]`
		`[|"abc"; "1234"|]`

Sitelet endpoint modifiers

- `[<EndPoint ...>]`: Specifying URL/method pairs

```
type EndPoint =
  | [<EndPoint "GET /about">] About
```

Sitelet endpoint modifiers

- `[<Query("param1", ...)>]`: specifying query parameters

```
type EndPoint =
    | [<EndPoint "/doc"; Query "version">] Document of int * version: int option
```

Sample Request	Parsed Request
`/doc/1234?version=1`	`Document(1234, Some 1)`
`/doc/1234`	`Document(1234, None)`

Sitelet endpoint modifiers

- `[<Json "param">]`: Specifying arguments to be passed as JSON (on POST)

```
type EndPoint =
    | [<EndPoint "POST /create"; Json "order">]
      CreateOrder of data: OrderData

and OrderData =
    { item: string; quantity: int }
```

Sample Request	Parsed Request

`/create`
`{ item:"Book", quantity:1 }`
	`CreateOrder({`
	`item="Book";quantity=1 })`

Sitelet endpoint modifiers

- `[<FormData("param1", …)>]`: Specify arguments to be passed as form data

Client-side templating

```
type MainTemplate = Templating.Template<"Main.html">

let Main ctx action title body =
    Content.Page(
        MainTemplate.Doc(
            title = title,
            menubar = ...,
            body = body
        )
    )
```

HTML templates

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta data-replace="meta" />
    <title>${title}</title>
    <meta data-replace="styles" />
</head>
<body>
    <div data-replace="body"></div>
    <script data-replace="scripts"></script>
</body>
</html>
```

- data-replace
- data-hole
- ${var}

Reactive template placeholders

`data-var`: bind the value of an input control to a reactive variable

`data-attr`: assign an attribute

`data-event-xxx`: bind an event handler for xxx

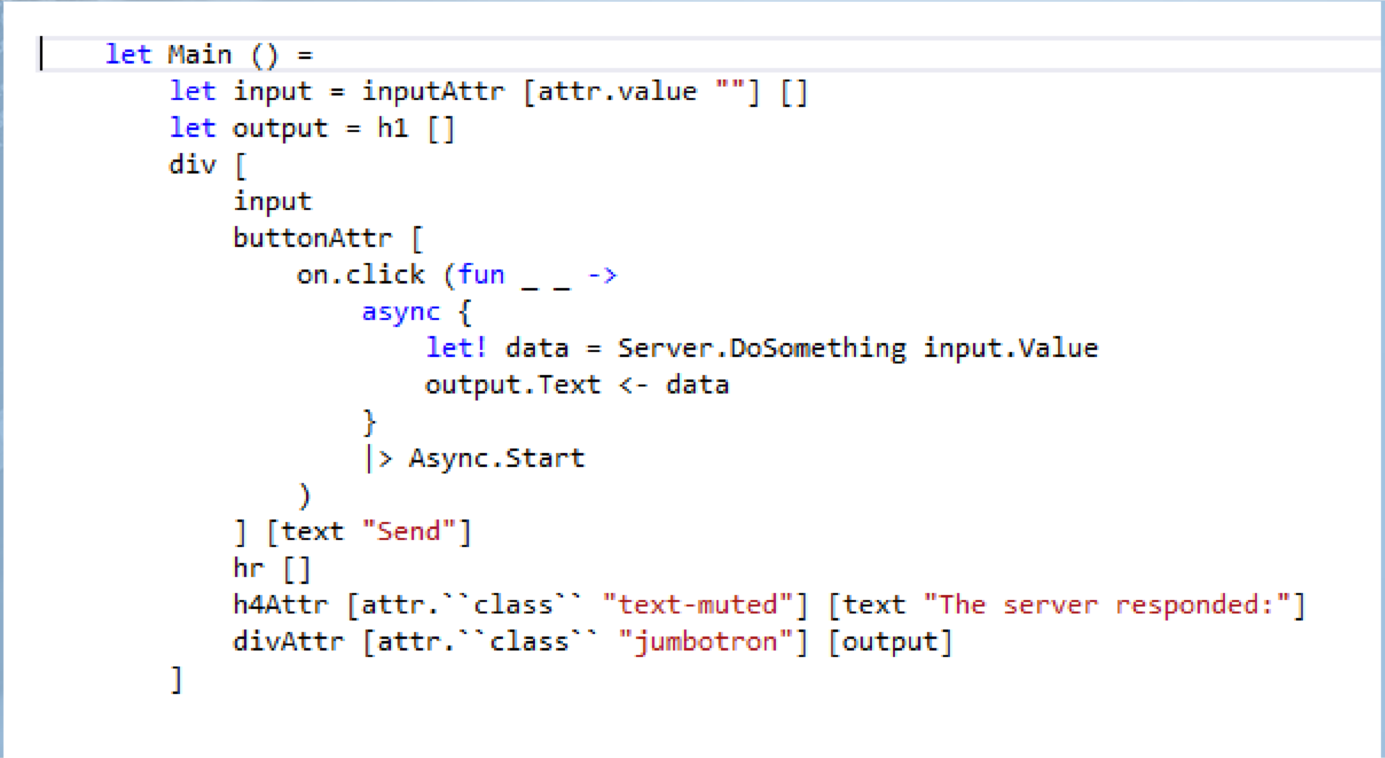
`data-template`: use the given node as a template

`data-children-template`: use the contents of the given node as a template

`$!{var}`: the view of a reactive variable

`http://try.websharper.com/example/todo-list`

Constructing HTML with UI.Next



```fsharp
let Main () =
    let input = inputAttr [attr.value ""] []
    let output = h1 []
    div [
        input
        buttonAttr [
            on.click (fun _ _ ->
                async {
                    let! data = Server.DoSomething input.Value
                    output.Text <- data
                }
                |> Async.Start
            )
        ] [text "Send"]
        hr []
        h4Attr [attr.``class`` "text-muted"] [text "The server responded:"]
        divAttr [attr.``class`` "jumbotron"] [output]
    ]
```

Event handling

```
[<JavaScript>]
let Main () =
    let input = inputAttr [attr.value ""] []
    ...
    buttonAttr [
        on.click (fun _ _ ->
              async {
                   let! data = Server.DoSomething input.Value
                   output.Text <- data
              } |> Async.Start
        )
    ] [text "Send"]
```

Write your server-side code

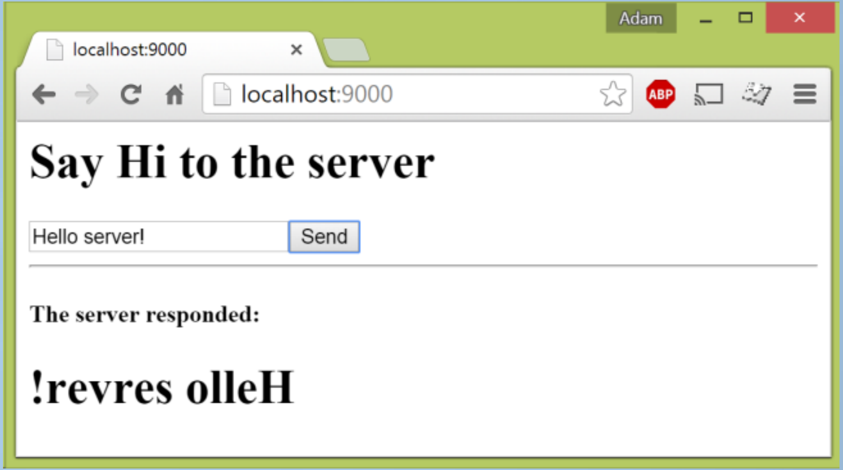
```
module Server =

    [<Rpc>]
    let DoSomething input =
        let R (s: string) = System.String(Array.rev(s.ToCharArray()))
        async {
            return R input
        }
```

Put your pages together

```
module Site =
    let HomePage ctx =
        Templating.Main ctx EndPoint.Home "Home" [
            h1 [text "Say Hi to the server!"]
            div [client <@ Client.Main() @>]
        ]

    let AboutPage ctx =
        Templating.Main ctx EndPoint.About "About" [
            h1 [text "About"]
            p [text "This is a WebSharper client-server application."]
        ]
```

Client-server applications

```
type EndPoint =
    | [<EndPoint "/">] Home
    | [<EndPoint "/about">] About


[<Website>]
let Main =
    Application.MultiPage (fun ctx endpoint ->
        match endpoint with
        | EndPoint.Home -> HomePage ctx
        | EndPoint.About -> AboutPage ctx
    )
```



Say Hi to the server

Hello server!

The server responded:

!revres olleH

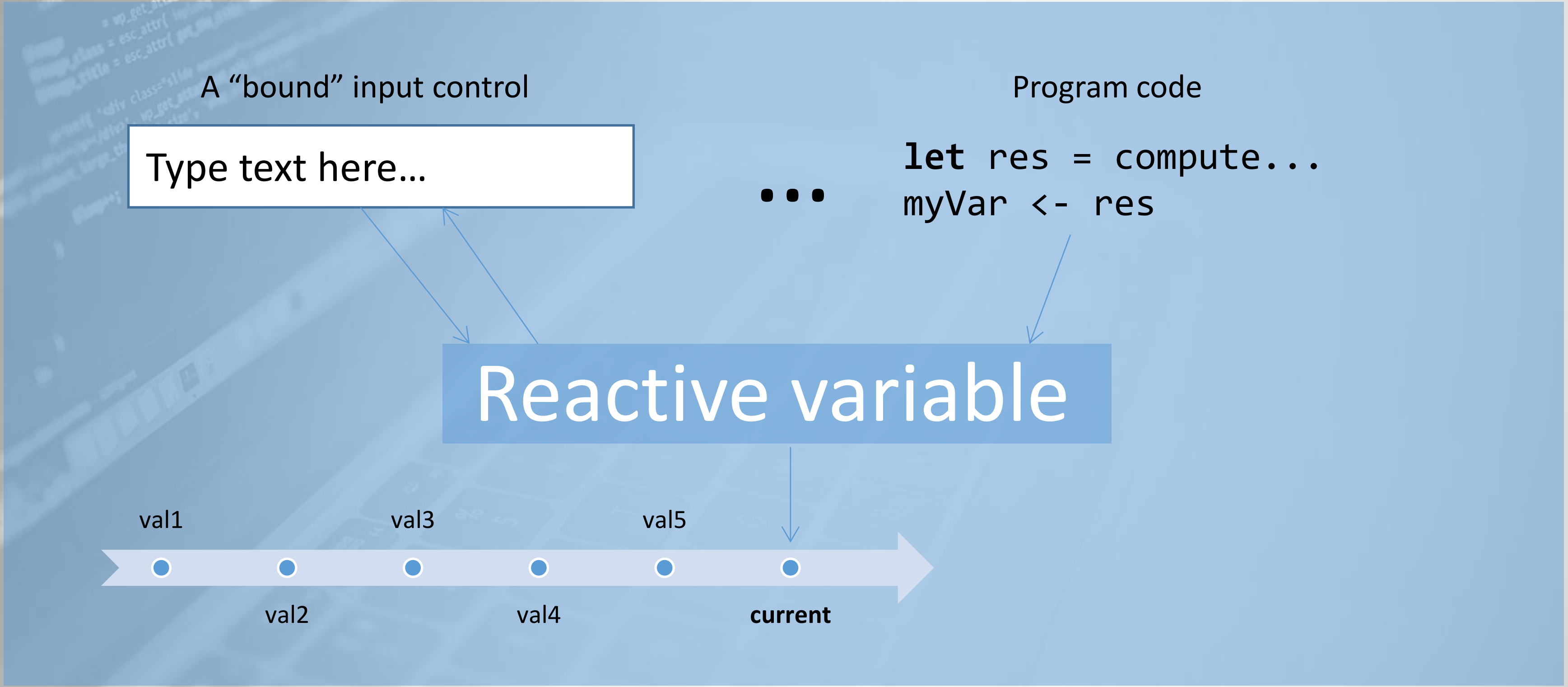
YES!
THAT'S WHAT I'M TALKING ABOUT.
imgflip.com

REST services

```
type Action =
| [<EndPoint "GET /person"; Query "id">] GetPerson of id:int
| [<EndPoint "POST /person"; Json "data">] PostPerson of data:Person
| [<EndPoint "PUT /person"; Query "id"; Json "data">] PutPerson of id:int * data:Person
| [<EndPoint "DELETE /person"; Query "id">] DeletePerson of id: int
```

http://www.websharper.com/tutorials#web_development/implementing_a_rest_api_(f-sharp)

Two-way data binding



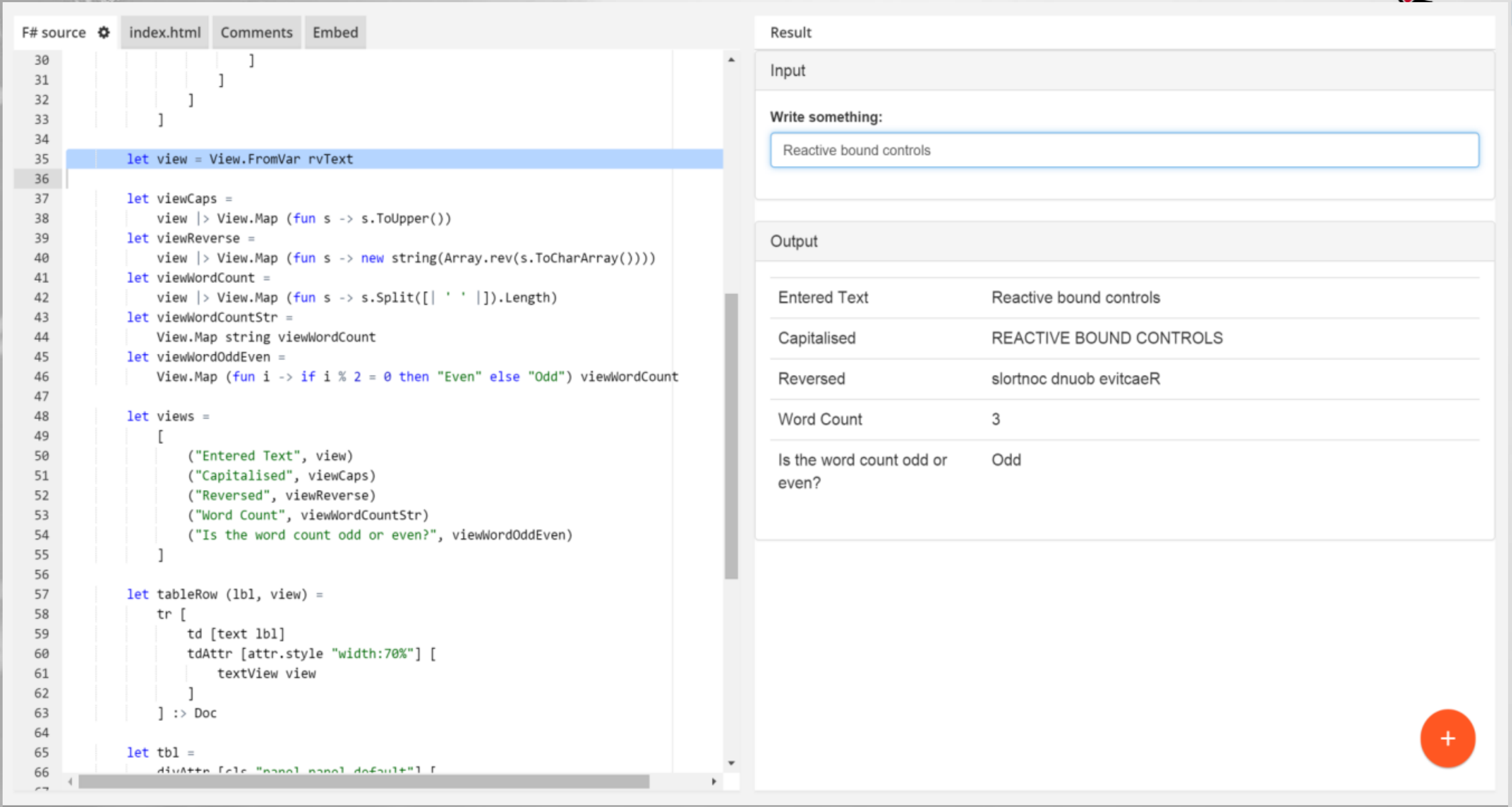
```
let res = compute...
myVar <- res
```

Reactive variables, bound controls, and views

```
open WebSharper.UI.Next
open WebSharper.UI.Next.Client

let v = Var.Create “first value”

let textbox = Doc.Input [] myVar

let view = View.FromVar v
```

Doc: a representation for a reactive DOM fragment (empty, or single or multiple node)

http://try.websharper.com/snippet/adam.granicz/00001u



```
                        ]
                    ]
                ]
            ]

        let view = View.FromVar rvText


        let viewCaps =
            view |> View.Map (fun s -> s.ToUpper())
        let viewReverse =
            view |> View.Map (fun s -> new string(Array.rev(s.ToCharArray())))
        let viewWordCount =
            view |> View.Map (fun s -> s.Split([| ' ' |]).Length)
        let viewWordCountStr =
            View.Map string viewWordCount
        let viewWordOddEven =
            View.Map (fun i -> if i % 2 = 0 then "Even" else "Odd") viewWordCount

        let views =
            [
                ("Entered Text", view)
                ("Capitalised", viewCaps)
                ("Reversed", viewReverse)
                ("Word Count", viewWordCountStr)
                ("Is the word count odd or even?", viewWordOddEven)
            ]

        let tableRow (lbl, view) =
            tr [
                td [text lbl]
                tdAttr [attr.style "width:70%"] [
                    textView view
                ]
            ] :> Doc

        let tbl =
            divAttr [cls "panel panel-default"] [
```

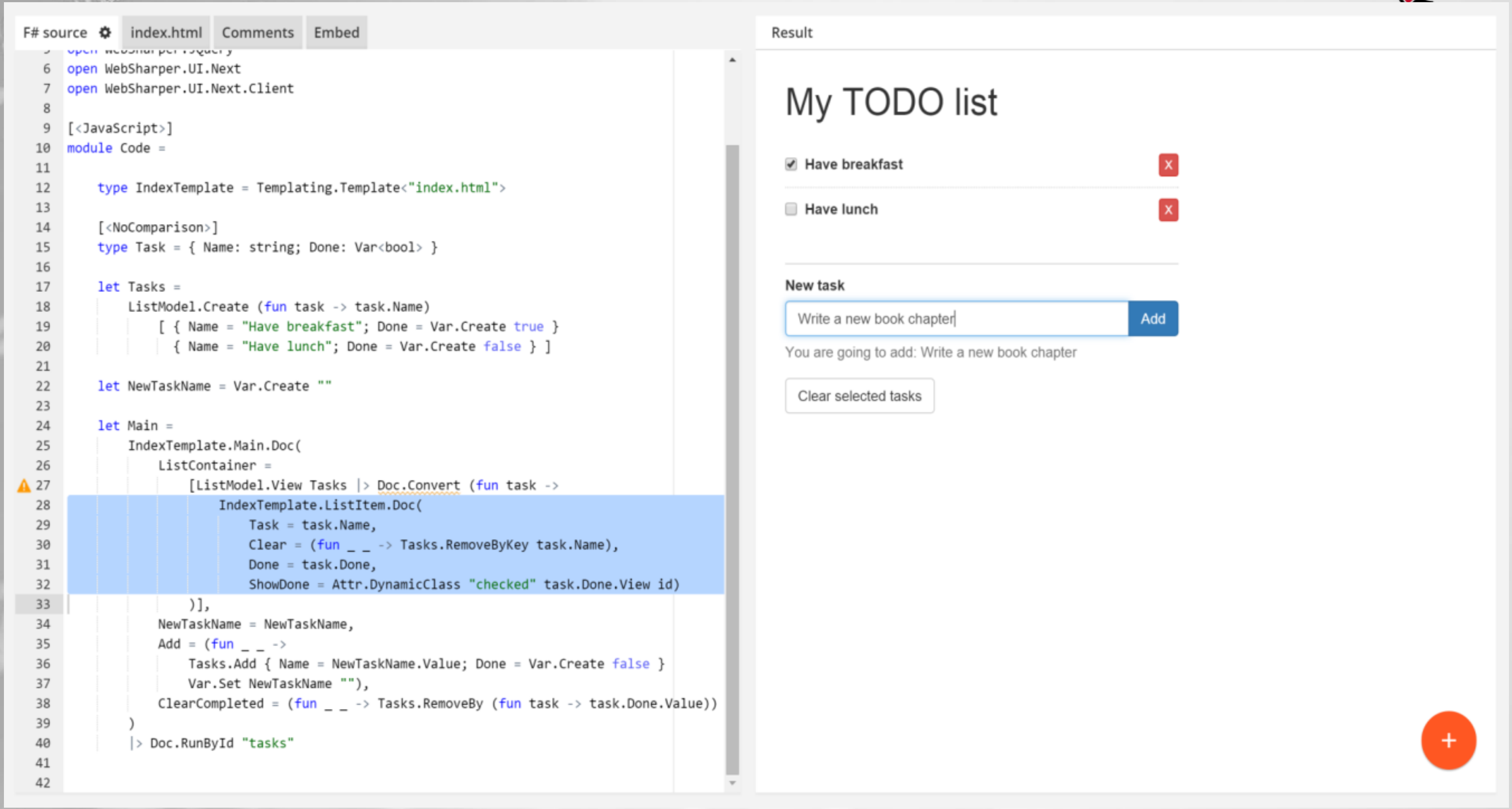
Input

Write something:

Reactive bound controls

Output

Entered Text	Reactive bound controls
Capitalised	REACTIVE BOUND CONTROLS
Reversed	slortnoc dnuob evitcaeR
Word Count	3
Is the word count odd or even?	Odd



```fsharp
open WebSharper.JQuery
open WebSharper.UI.Next
open WebSharper.UI.Next.Client

[<JavaScript>]
module Code =

    type IndexTemplate = Templating.Template<"index.html">

    [<NoComparison>]
    type Task = { Name: string; Done: Var<bool> }

    let Tasks =
        ListModel.Create (fun task -> task.Name)
            [ { Name = "Have breakfast"; Done = Var.Create true }
              { Name = "Have lunch"; Done = Var.Create false } ]

    let NewTaskName = Var.Create ""

    let Main =
        IndexTemplate.Main.Doc(
            ListContainer =
                [ListModel.View Tasks |> Doc.Convert (fun task ->
                    IndexTemplate.ListItem.Doc(
                        Task = task.Name,
                        Clear = (fun _ _ -> Tasks.RemoveByKey task.Name),
                        Done = task.Done,
                        ShowDone = Attr.DynamicClass "checked" task.Done.View id)
                )],
            NewTaskName = NewTaskName,
            Add = (fun _ _ ->
                Tasks.Add { Name = NewTaskName.Value; Done = Var.Create false }
                Var.Set NewTaskName ""),
            ClearCompleted = (fun _ _ -> Tasks.RemoveBy (fun task -> task.Done.Value))
        )
        |> Doc.RunById "tasks"
```

My TODO list

- [x] Have breakfast X
- [ ] Have lunch X

New task

Write a new book chapter | Add

You are going to add: Write a new book chapter

Clear selected tasks

Formlets

A compositional abstraction for constructing web forms:

```
Formlet.Return (fun fn age -> { FirstName=fn; Age=age })
<*> Controls.Input "First name"
<*> (Controls.Input "20"
    |> Validation.IsMatch "^[1-9][0-9]*$" "Need an int"
    |> Formlet.Map (int))
```

http://try.websharper.com/snippet/adam.granicz/00003G

Dependent formlets and flowlets

Enhance flowlets with dynamic composition

Use the bind operator (`let!` in an F# computation expr)

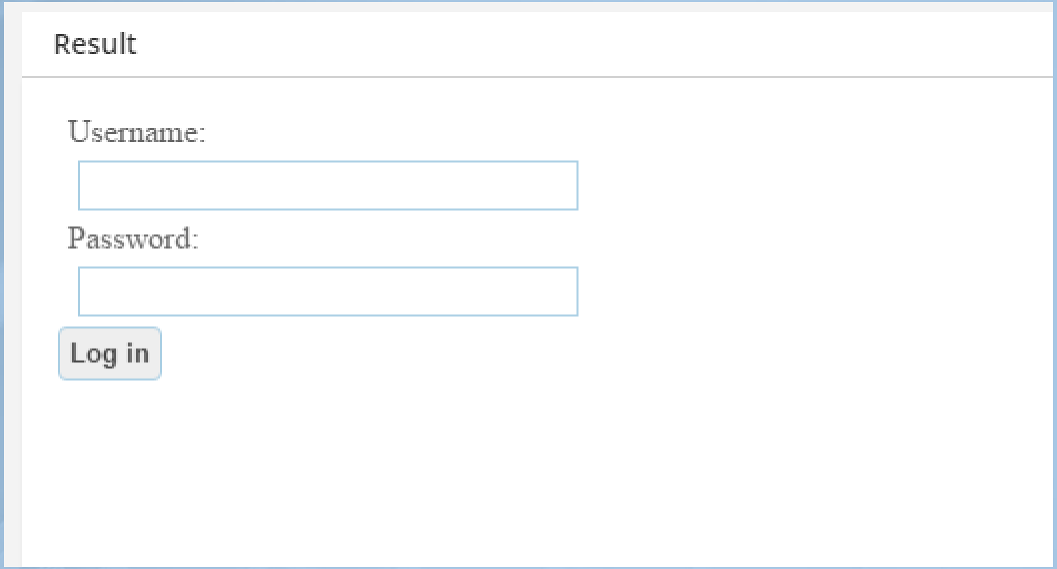
J. Bjornson, A. Tayanovskyy, A. Granicz. *Composing Reactive GUIs in F# using WebSharper*. IFL 2010.

```
Formlet.Do {
    let! fn = Control.Input "First name"
    let! age = (Control.Input "20" |> …)
    return { Firstname=fn; Age=age }
}
```

Reactive formlets

```
let rvUsername = Var.Create ""
let rvPassword = Var.Create ""
Formlet.Return (fun user pass -> (user, pass))
<*> (Controls.InputVar rvUsername
    |> Formlet.WithLabel (text "Username: "))
<*> (Controls.InputVar rvPassword
    |> Formlet.WithLabel (text "Password: "))
|> Formlet.WithSubmit "Log in"
|> Formlet.WithFormContainer
|> Formlet.Run (fun (user, pass) ->
    JS.Alert ("Welcome, " + user + "!"))
|> Doc.RunById "main"
```

Reactive formlets


Result

Username:

Password:

http://try.websharper.com/snippet/00004B

Reactive piglets – WebSharper.Forms

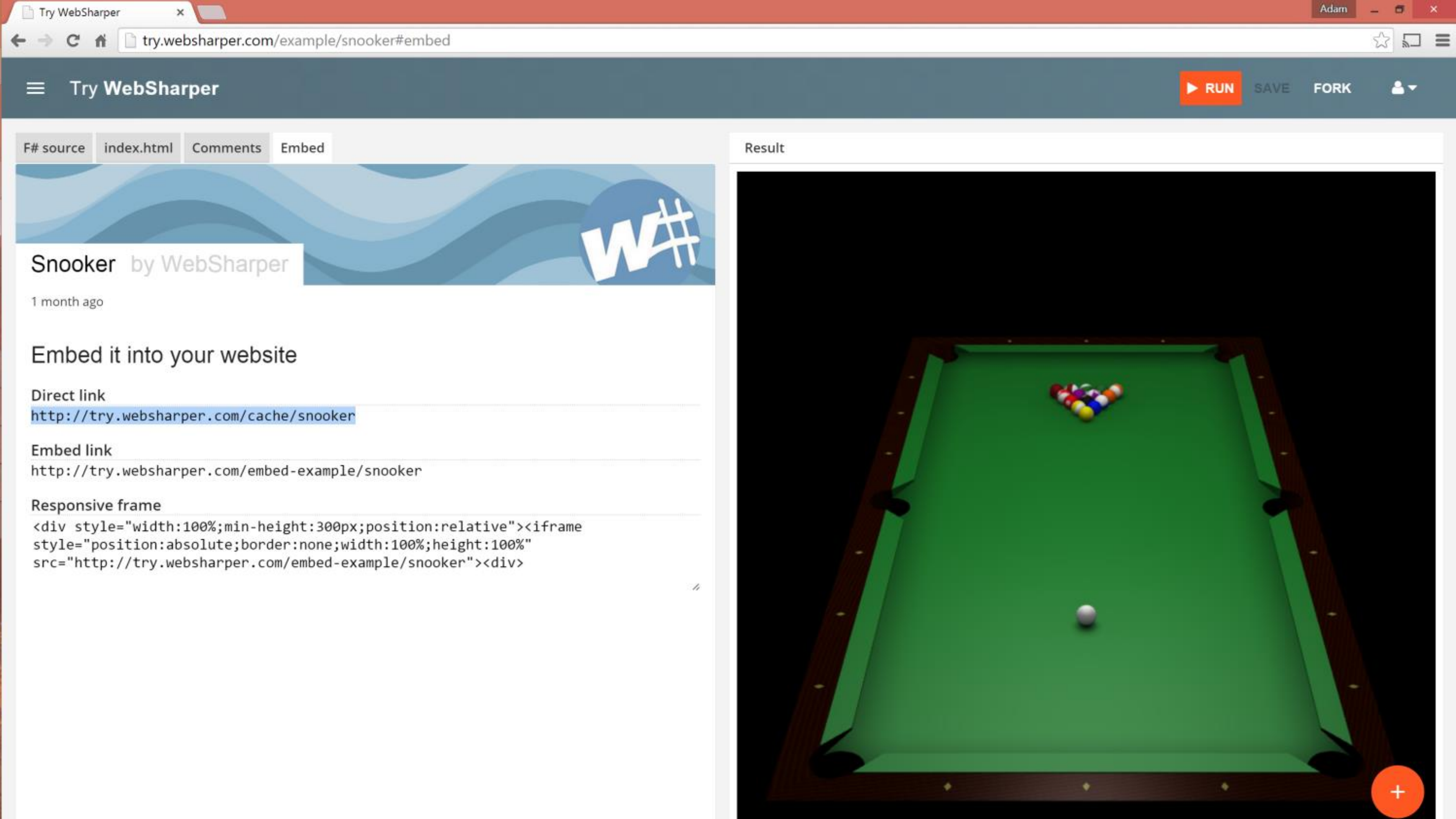
```
Piglet.Return (fun user pass -> (user, pass))
<*> Piglet.Yield ""
<*> Piglet.Yield ""
|> Piglet.WithSubmit
|> Piglet.Run (fun (user, pass) ->
    JS.Alert ("Welcome, " + user + "!"))
|> Piglet.Render (fun rvUsername rvPassword submit ->
    form [
        ...
    ]
)
```

http://try.websharper.com/snippet/00004x

Reactive “sitelets”

Client-side routing

`http://try.websharper.com/snippet/adam.granicz/000033`



Snooker by WebSharper

1 month ago

Embed it into your website

Direct link

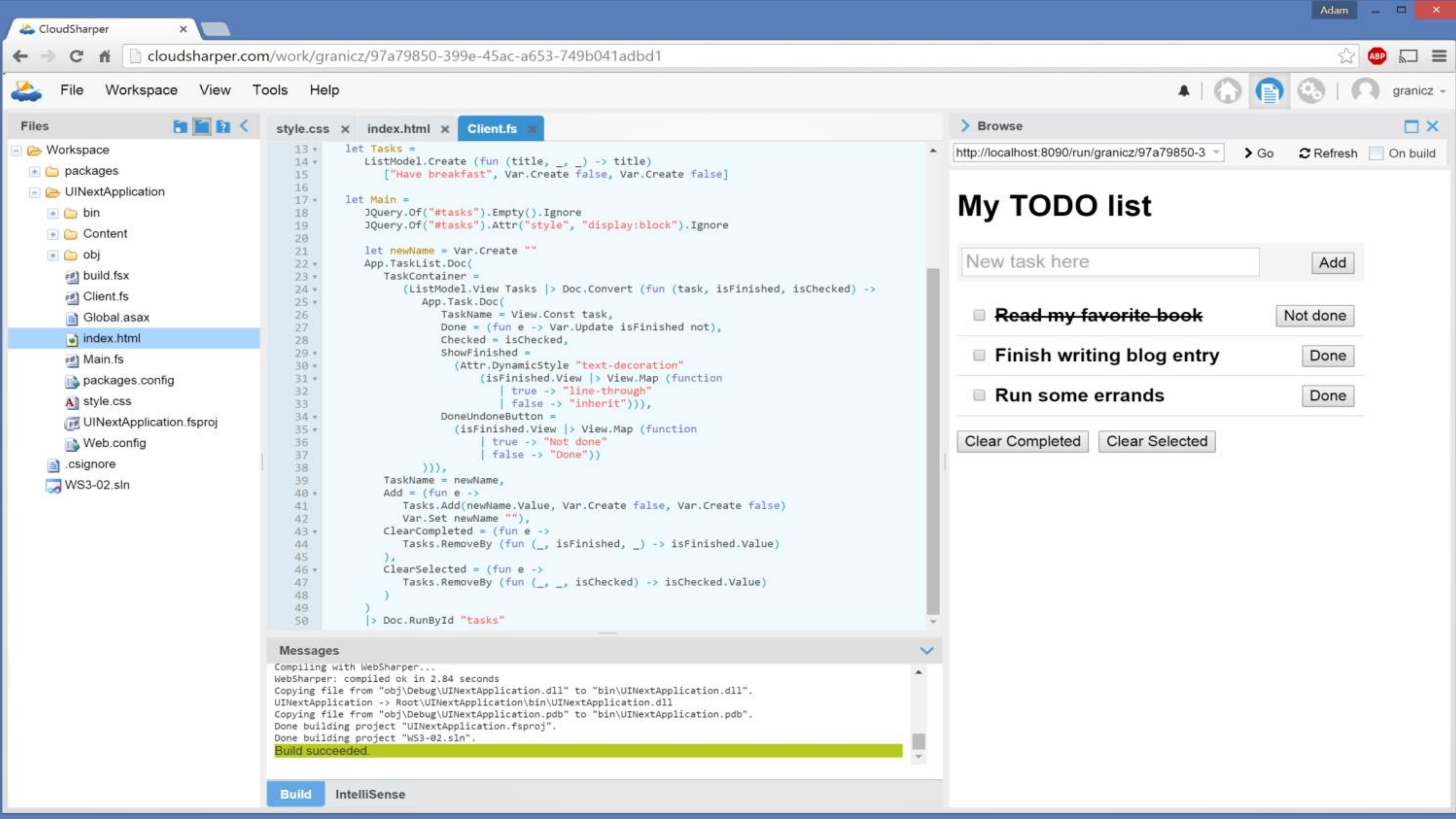
http://try.websharper.com/cache/snooker

Embed link

http://try.websharper.com/embed-example/snooker

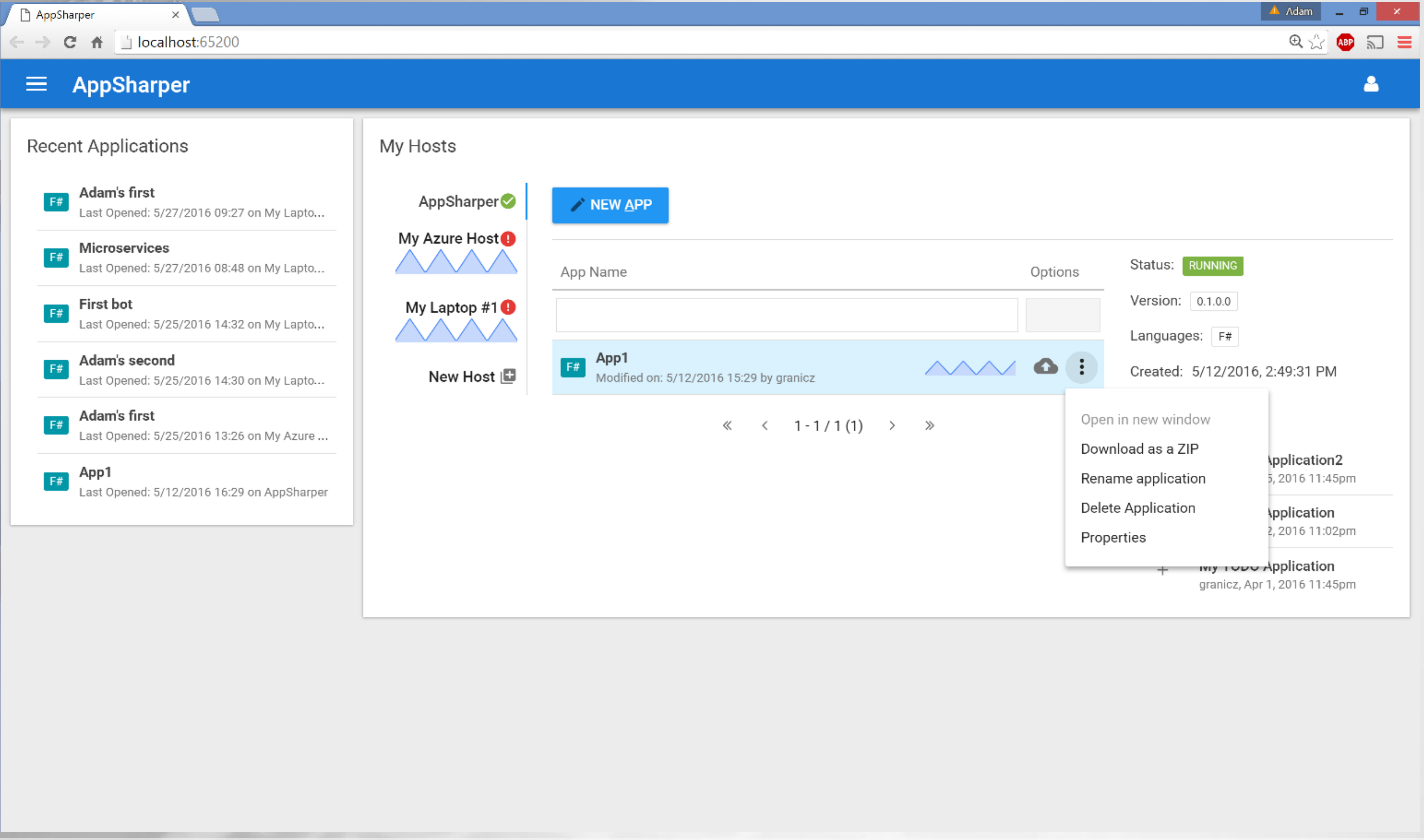
Responsive frame

```
<div style="width:100%;min-height:300px;position:relative"><iframe
style="position:absolute;border:none;width:100%;height:100%"
src="http://try.websharper.com/embed-example/snooker"><div>
```



```
    let Tasks =
       ListModel.Create (fun (title, _, _) -> title)
          ["Have breakfast", Var.Create false, Var.Create false]

    let Main =
       JQuery.Of("#tasks").Empty().Ignore
       JQuery.Of("#tasks").Attr("style", "display:block").Ignore

       let newName = Var.Create ""
       App.TaskList.Doc(
          TaskContainer =
             (ListModel.View Tasks |> Doc.Convert (fun (task, isFinished, isChecked) ->
                App.Task.Doc(
                   TaskName = View.Const task,
                   Done = (fun e -> Var.Update isFinished not),
                   Checked = isChecked,
                   ShowFinished =
                     (Attr.DynamicStyle "text-decoration"
                         (isFinished.View |> View.Map (function
                            | true -> "line-through"
                            | false -> "inherit"))),
                   DoneUndoneButton =
                     (isFinished.View |> View.Map (function
                         | true -> "Not done"
                         | false -> "Done"))
                ))),
          TaskName = newName,
          Add = (fun e ->
             Tasks.Add(newName.Value, Var.Create false, Var.Create false)
             Var.Set newName ""),
          ClearCompleted = (fun e ->
             Tasks.RemoveBy (fun (_, isFinished, _) -> isFinished.Value)
          ),
          ClearSelected = (fun e ->
             Tasks.RemoveBy (fun (_, _, isChecked) -> isChecked.Value)
          )
       )
       |> Doc.RunById "tasks"
```

```
Compiling with WebSharper...
WebSharper: compiled ok in 2.84 seconds
Copying file from "obj\Debug\UINextApplication.dll" to "bin\UINextApplication.dll".
UINextApplication -> Root\UINextApplication\bin\UINextApplication.dll
Copying file from "obj\Debug\UINextApplication.pdb" to "bin\UINextApplication.pdb".
Done building project "UINextApplication.fsproj".
Done building project "WS3-02.sln".
Build succeeded.
```



AppSharper

localhost:65200

AppSharper

Recent Applications

F#	Adam's first Last Opened: 5/27/2016 09:27 on My Lapto...
F#	Microservices Last Opened: 5/27/2016 08:48 on My Lapto...
F#	First bot Last Opened: 5/25/2016 14:32 on My Lapto...
F#	Adam's second Last Opened: 5/25/2016 14:30 on My Lapto...
F#	Adam's first Last Opened: 5/25/2016 13:26 on My Azure ...
F#	App1 Last Opened: 5/12/2016 16:29 on AppSharper

My Hosts

AppSharper ✓
My Azure Host !
My Laptop #1 !
New Host

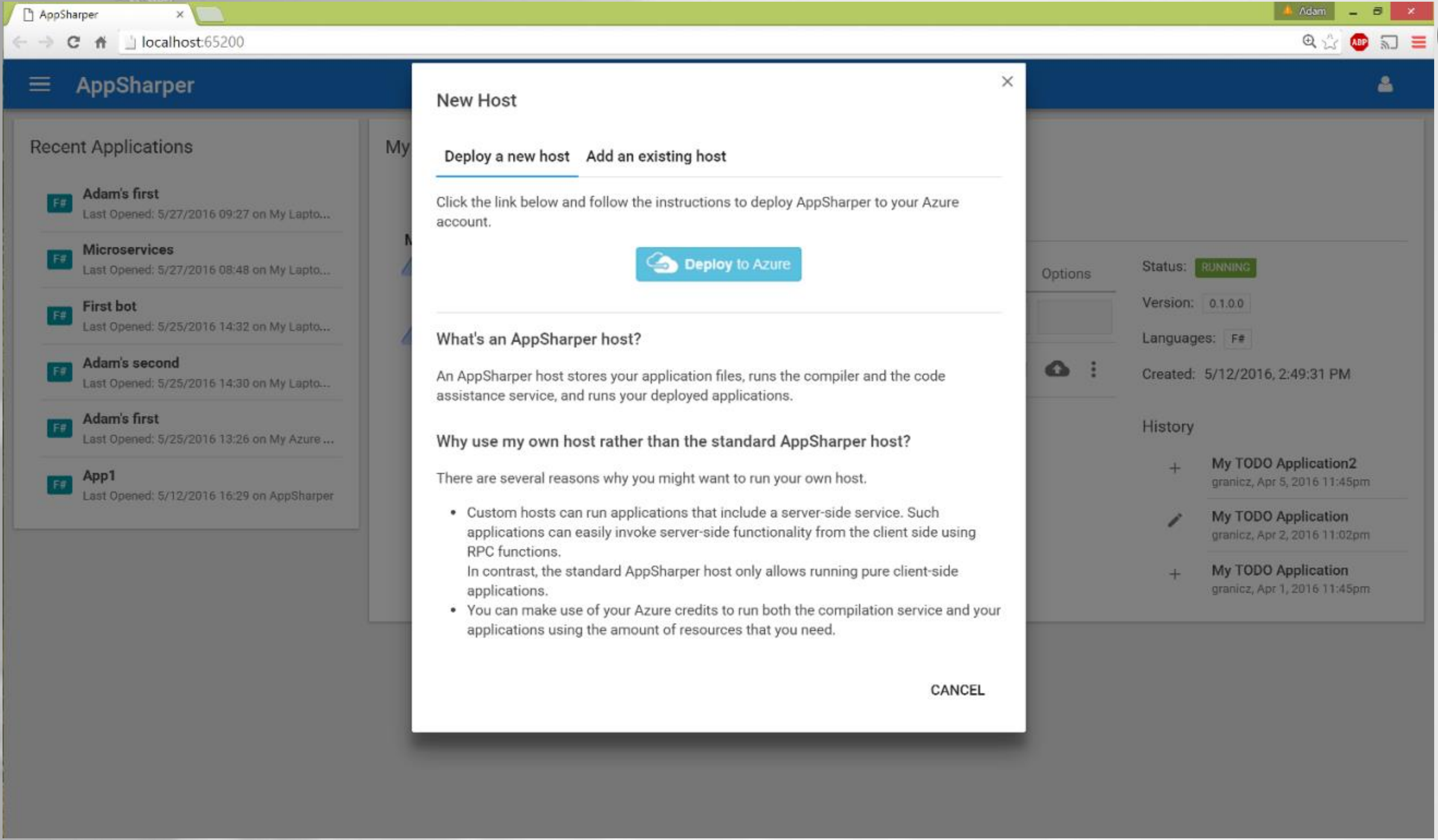
NEW APP

App Name	Options
F# App1 Modified on: 5/12/2016 15:29 by granicz	

1 - 1 / 1 (1)

Status: RUNNING
Version: 0.1.0.0
Languages: F#
Created: 5/12/2016, 2:49:31 PM

Application2
5, 2016 11:45pm
Application
2, 2016 11:02pm
My TODO Application
granicz, Apr 1, 2016 11:45pm



AppSharper

localhost:65200

AppSharper

Recent Applications

F#	Adam's first Last Opened: 5/27/2016 09:27 on My Lapto...
F#	Microservices Last Opened: 5/27/2016 08:48 on My Lapto...
F#	First bot Last Opened: 5/25/2016 14:32 on My Lapto...
F#	Adam's second Last Opened: 5/25/2016 14:30 on My Lapto...
F#	Adam's first Last Opened: 5/25/2016 13:26 on My Azure ...
F#	App1 Last Opened: 5/12/2016 16:29 on AppSharper

Status: RUNNING

Version: 0.1.0.0

Languages: F#

Created: 5/12/2016, 2:49:31 PM

History

My TODO Application2
granicz, Apr 5, 2016 11:45pm

My TODO Application
granicz, Apr 2, 2016 11:02pm

My TODO Application
granicz, Apr 1, 2016 11:45pm

New Host

Deploy a new host | Add an existing host

Click the link below and follow the instructions to deploy AppSharper to your Azure account.

Deploy to Azure

What's an AppSharper host?

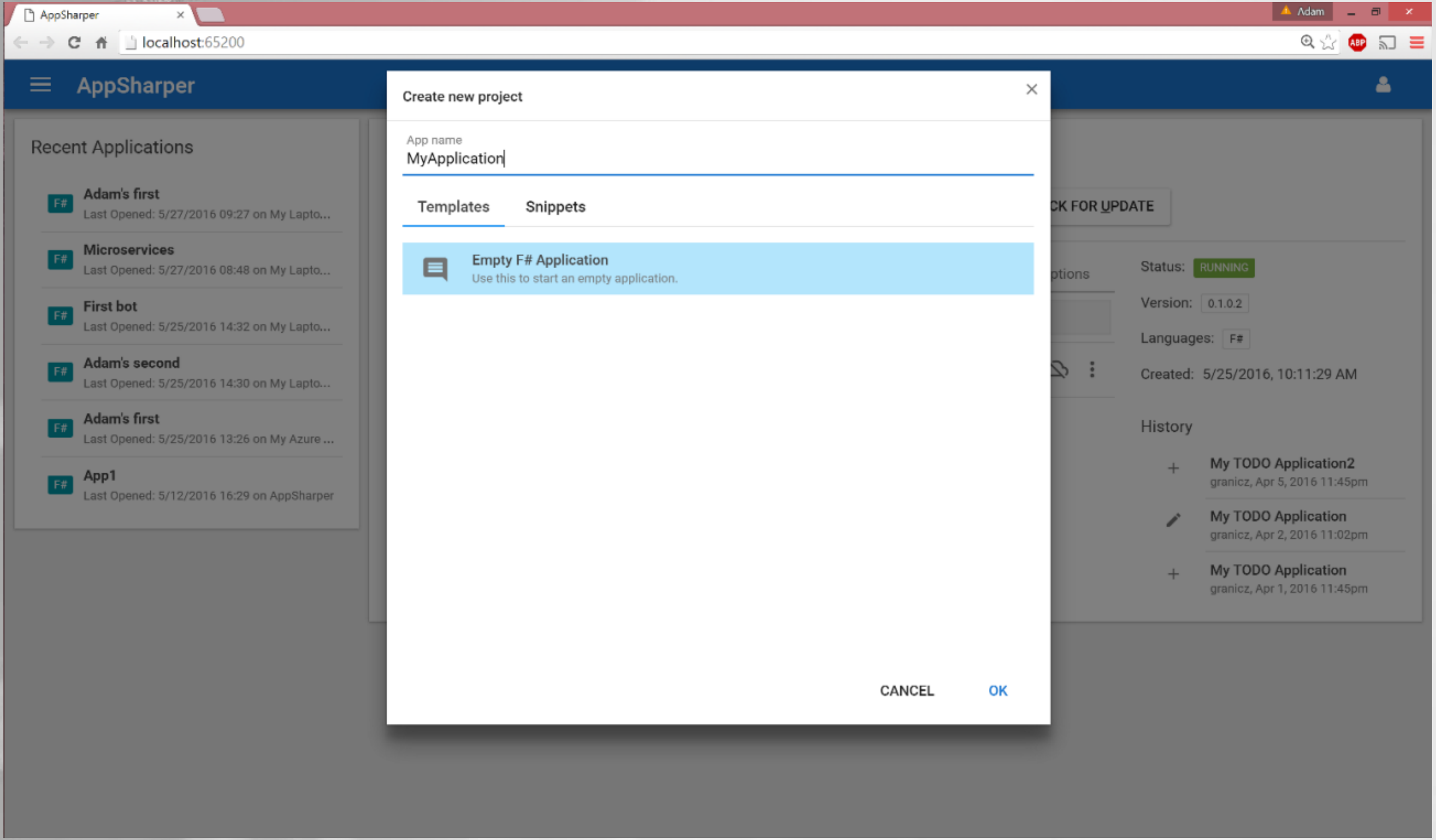
An AppSharper host stores your application files, runs the compiler and the code assistance service, and runs your deployed applications.

Why use my own host rather than the standard AppSharper host?

There are several reasons why you might want to run your own host.

- Custom hosts can run applications that include a server-side service. Such applications can easily invoke server-side functionality from the client side using RPC functions.
  In contrast, the standard AppSharper host only allows running pure client-side applications.
- You can make use of your Azure credits to run both the compilation service and your applications using the amount of resources that you need.

CANCEL



AppSharper

localhost:65200

AppSharper

Recent Applications

F# Adam's first
Last Opened: 5/27/2016 09:27 on My Lapto...

F# Microservices
Last Opened: 5/27/2016 08:48 on My Lapto...

F# First bot
Last Opened: 5/25/2016 14:32 on My Lapto...

F# Adam's second
Last Opened: 5/25/2016 14:30 on My Lapto...

F# Adam's first
Last Opened: 5/25/2016 13:26 on My Azure ...

F# App1
Last Opened: 5/12/2016 16:29 on AppSharper

Create new project

App name
MyApplication

Templates | Snippets

Empty F# Application
Use this to start an empty application.

CANCEL OK

CK FOR UPDATE

Status: RUNNING

Version: 0.1.0.2

Languages: F#

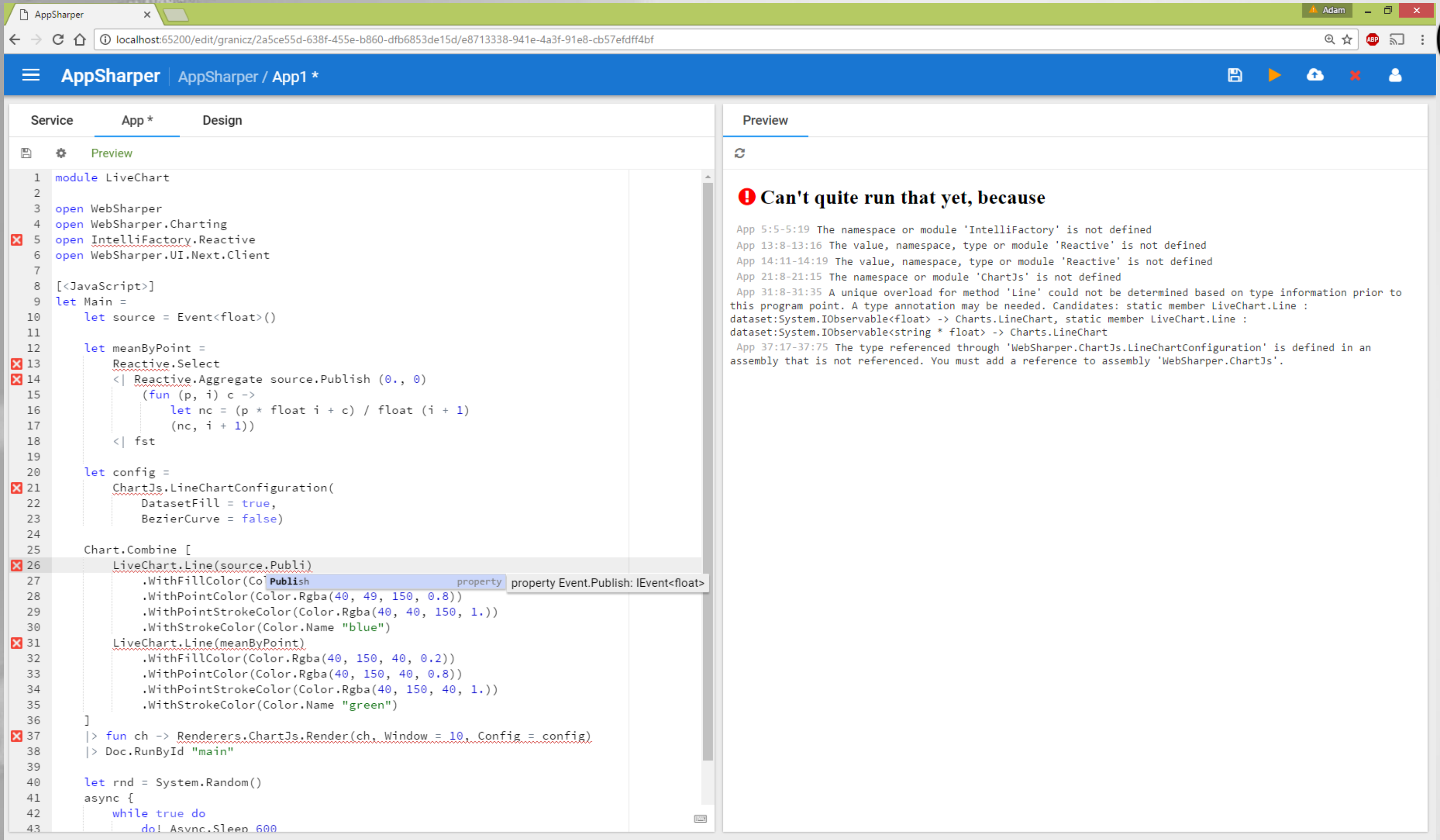
Created: 5/25/2016, 10:11:29 AM

History

My TODO Application2
granicz, Apr 5, 2016 11:45pm

My TODO Application
granicz, Apr 2, 2016 11:02pm

My TODO Application
granicz, Apr 1, 2016 11:45pm


AppSharper
localhost:65200/edit/granicz/2a5ce55d-638f-455e-b860-dfb6853de15d/e8713338-941e-4a3f-91e8-cb57efdff4bf
AppSharper
AppSharper / App1 *
Service
App *
Design
Preview
Preview

```
 1  module LiveChart
 2
 3  open WebSharper
 4  open WebSharper.Charting
 5  open IntelliFactory.Reactive
 6  open WebSharper.UI.Next.Client
 7
 8  [<JavaScript>]
 9  let Main =
10      let source = Event<float>()
11
12      let meanByPoint =
13          Reactive.Select
14          Reactive.Aggregate source.Publish (0., 0)
15              (fun (p, i) c ->
16                  let nc = (p * float i + c) / float (i + 1)
17                  (nc, i + 1))
18          <| fst
19
20      let config =
21          ChartJs.LineChartConfiguration(
22              DatasetFill = true,
23              BezierCurve = false)
24
25      Chart.Combine [
26          LiveChart.Line(source.Publi)
27              .WithFillColor(Co  Publish                property   property Event.Publish: IEvent<float>
28              .WithPointColor(Color.Rgba(40, 49, 150, 0.8))
29              .WithPointStrokeColor(Color.Rgba(40, 40, 150, 1.))
30              .WithStrokeColor(Color.Name "blue")
31          LiveChart.Line(meanByPoint)
32              .WithFillColor(Color.Rgba(40, 150, 40, 0.2))
33              .WithPointColor(Color.Rgba(40, 150, 40, 0.8))
34              .WithPointStrokeColor(Color.Rgba(40, 150, 40, 1.))
35              .WithStrokeColor(Color.Name "green")
36      ]
37      |> fun ch -> Renderers.ChartJs.Render(ch, Window = 10, Config = config)
38      |> Doc.RunById "main"
39
40      let rnd = System.Random()
41      async {
42          while true do
43              do! Async.Sleep 600
```

Can't quite run that yet, because

```
App 5:5-5:19 The namespace or module 'IntelliFactory' is not defined
App 13:8-13:16 The value, namespace, type or module 'Reactive' is not defined
App 14:11-14:19 The value, namespace, type or module 'Reactive' is not defined
App 21:8-21:15 The namespace or module 'ChartJs' is not defined
App 31:8-31:35 A unique overload for method 'Line' could not be determined based on type information prior to
this program point. A type annotation may be needed. Candidates: static member LiveChart.Line :
dataset:System.IObservable<float> -> Charts.LineChart, static member LiveChart.Line :
dataset:System.IObservable<string * float> -> Charts.LineChart
App 37:17-37:75 The type referenced through 'WebSharper.ChartJs.LineChartConfiguration' is defined in an
assembly that is not referenced. You must add a reference to assembly 'WebSharper.ChartJs'.
```

```
module Client =

    open IntelliFactory.WebSharper.Html

    [<Sealed>]
    type Control() =
        inherit Web.Control()
        [<JavaScript>]
        override this.Body =
            Div []
            |>! OnAfterRender (fun _ ->
                App.Ini())
            :> _

module Pages =

    type Index =
        {
            Body : Content.HtmlElement list
        }
```

Questions?

Get in touch
@granicz
@intellifactory
@websharper

http://intellifactory.com
http://websharper.com
http://cloudsharper.com

.NET DEVELOPER DAYS

net.developerdays.pl
@DeveloperDaysPL

Sponsors and Partners

Strategic Sponsors

adform

LUXOFT

Gold Sponsors

DevExpress®

Roche

KMD
KMDPOLAND.PL

VOLVO

Silver Sponsors

it KONTRAKT®

Fast Reports
Reporting must be Fast!

JCommerce

JETBRAINS THE DRIVE TO DEVELOP

labs
software specialists

Millennium
bank