



Webes és mobil-alapú megoldások
hatékonyan és gyorsan a közigazgatásban



Gránicz Ádám, ügyvezető
IntelliFactory Kft.
granicz.adam@intellifactory.com

Infotér Konferencia 2011, Balatonfüred -
Nov 22-23, 2011

Közigazgatás <-- --> Állampolgárok

- Hatékonyság
- Elektronikus kapcsolatteremtés lehetőségeinek bővítése
- Mobil és webes alkalmazások

A mobil piac†

- 3,7 milliárd mobil felhasználó, 4,3 milliárd telefon
- \$1,2 billió (ezer milliárd) cserélt gazdát:
a **leggyorsabban fejlődő iparág** a földön
- \$8,8 milliárd ment hirdetésre/marketingre
- **\$3 milliárdot** költöttek „fogyasztói” alkalmazásokra
- **\$6 milliárdot** költöttek céges alkalmazásokra
- **Természetesen rohamos növekedés várható**

† forrás: <http://communities-dominate.blogs.com/brands/2011/02/all-the-numbers-all-the-facts-on-mobile-the-trillion-dollar-industry-why-is-google-saying-put-your-b.html>

A potenciális mobil ügyfelek elérése†

Csatorna	Költség	Hatásfok	Fő/\$
Mobil website	\$30.000	36,40%	2.840
iPhone alkalmazás	\$30.000	6,75%	527
Mobil alkalmazás a 3 fő platformra	\$90.000	24,04%	599

† source: <http://mashable.com/2011/02/24/mobile-app-dev-cost/>

- iOS – Objective-C „pokol” + Macintosh + egyedi fejlesztési környezet és eszközök
- Android – Java + egy terjedelmes API
- Windows Phone - .NET + egy terjedelmes API
- Windows Mobile - ...

- Lehet-e azonos kódbázisból több platformra alkalmazásokat fejleszteni?
 - Nem igazán, hacsak nem írjuk át az alkalmazást egy másik nyelvre

Mire mondanánk hogy „igen, ez kell nekem!”?

Például a következőkre:

- Kevesebb kódra
- Magasabb szintű absztrakciók használatára
- Erősen tipizált, deklaratív UI-k
- Multi-targetálás
- A webes alkalmazások desktopra vitelére
- Felhő-alapú skálázásra

Mindezekkel óriási megtakarítások érhetőek el mind idő/munka és fenntarthatóság szempontjából

- JavaScript a futtató nyelve ...
 - A kliens oldali web alkalmazásoknak és idővel
 - Desktop alkalmazásoknak: Windows 8
 - Mobil alkalmazásoknak: Android, WP7, etc.
- Néhányan még szerver oldali kódot is fejlesztenek JavaScript-ben (lásd Node.js)

- Írjunk hát mobil alkalmazásokat JavaScript-ben!
- De, mi a helyzet a PhoneGap-el és társaival?
 - Járható, de JavaScript-et jól írni nehéz, mert:
 - Gyengén tipizált
 - Nincs fordításidejű garancia
 - Kódolás közben minimális segítség – kód kiegészítés, stb.
 - És még mindig maradt N-1 ok a váltásra

- **Vállalati erősségű webes keretrendszer**
- A teljes **szerver+kliens kód egy nyelven**
- Ebből teljes **webes alkalmazásokat állít elő**
- Bármely JS könyvtárral együtt tud működni
- **Hatékony és ugrásszerű teljesítménynövekedést eredményező absztrakciók**
- **Automatikus és teljes erőforrás követés**
- **Biztonságos URL-ek**
- És még rengeteg más...

Kevesebb kód (50-90% megtakarítás!)

Gyorsabban fejleszthető

Könnyebben tenntartható, bővíthető alkalmazások

- Mobil képességeket tesz elérhetővé JavaScript-ből
- Csomag készítés automatizálása
- Gyors és hatékony multi-targeting
- Felhős infrastruktúrába is skálázható

Több tucat bővítés itt:

<http://websharper.com/extensions>

- Erősen tipizált
- Egymásbaágyazható
- Első-osztályú

Egy únió típusban írja le a teljes webes funkcionalitást

/// Actions that correspond to the different pages in the site.

type Action =

| Home

| Contact

| Protected

| Login of option<Action>

| Logout

| Echo of string

Erősen tipizált template-k, biztonságos URL-ek

```
let Template title main : Content<Action> =  
  let menu (ctx: Context<Action>)=  
    let (!) action = ctx.Link action |> HRef  
    [  
      A [!Action.Home] -< [Text "Home"]  
      A [!Action.Contact] -< [Text "Contact"]  
      A [!(Action.Echo "Hello")] -< [Text "Say Hello"]  
      A ["~/LegacyPage.aspx" |> ctx.ResolveUrl |> HRef] -< [Text "ASPX Page"]  
    ]  
  |> List.map (fun link ->  
    Label [Class "menu-item"] -< [link]  
  )
```



Templates.Skin.Skin (Some title)

```
{  
  LoginInfo = Widgets.LoginInfo  
  Banner = fun ctx -> [H2 [Text title]]  
  Menu = menu  
  Main = main  
  Sidebar = fun ctx -> [Text "Put your side bar here"]  
  Footer = fun ctx -> [Text "Copyright (c) 2011 YourCompany.com"]  
}
```

```
/// The pages of this website.
```

```
module Pages =
```

```
/// A helper function to create a hyperlink
```

```
let ( => ) title href = A [HRef href] -< [Text title]
```

```
/// The home page.
```

```
let HomePage : Content<Action> =
```

```
  Template "Home" <| fun ctx ->
```

```
    [
```

```
      H1 [Text "Welcome to our site!"]
```

```
      "Contact" => ctx.Link Action.Contact
```

```
    ]
```

```
  ...
```


// A simple sitelet for the root of the site.

let home =

Sitelet.Content "/" Action.Home Pages.HomePage

// An automatically inferred sitelet for the basic parts.

let basic =

Sitelet.Infer <| **fun** action ->

match action **with**

| Action.Contact -> Pages.ContactPage

| Action.Echo param -> Pages.EchoPage param

| Action.Login action -> Pages.LoginPage action

| Action.Logout ->

 // Logout user and redirect to home

 UserSession.Logout ()

 Content.Redirect Action.Home

| Action.Home -> Content.Redirect Action.Home

| Action.Protected -> Content.ServerError

```
// A sitelet for the protected content.
```

```
let authenticated =
```

```
    let filter : Sitelet.Filter<Action> =
```

```
    {
```

```
        VerifyUser = fun _ -> true
```

```
        LoginRedirect = Some >> Action.Login
```

```
    }
```

```
Sitelet.Protect filter
```

```
<| Sitelet.Content
```

```
    "/protected"
```

```
    Action.Protected
```

```
    Pages.ProtectedPage
```

```
let EntireSite =
```

```
// Compose the above sitelets into a larger one.
```

```
Sitelet.Sum
```

```
[
```

```
  home
```

```
  authenticated
```

```
  basic
```

```
]
```

- Erősen tipizált
- Egymásbaágyazható
- Első-osztályú

- Dependens formlet-ek
- Flowlet-ek

```
let TB label msg =  
    Controls.Input ""  
    |> Validator.IsNotEmpty msg  
    |> Enhance.WithValidationIcon  
    |> Enhance.WithTextLabel label
```

Formlet.Yield (fun $v_1 v_2 \dots v_n \rightarrow$ <compose all v_i 's>)

<*> formlet₁

<*> formlet₂

...

<*> formlet_n

```
type Person = { Name: string; Email: string }
```

```
[<JavaScript>]
```

```
let PersonFormlet () : Formlet<Person> =
```

```
  let nameF = TB "Name" "Empty name not allowed"
```

```
  let emailF = TB "Email" "Please enter a valid email address"
```

```
    Formlet.Yield (fun name email -> { Name = name; Email = email })
```

```
  <*> nameF
```

```
  <*> emailF
```

```
  |> Enhance.WithSubmitAndResetButtons
```

```
  |> Enhance.WithLegend "Add a New Person"
```

```
  |> Enhance.WithFormContainer
```

Az eredmény:



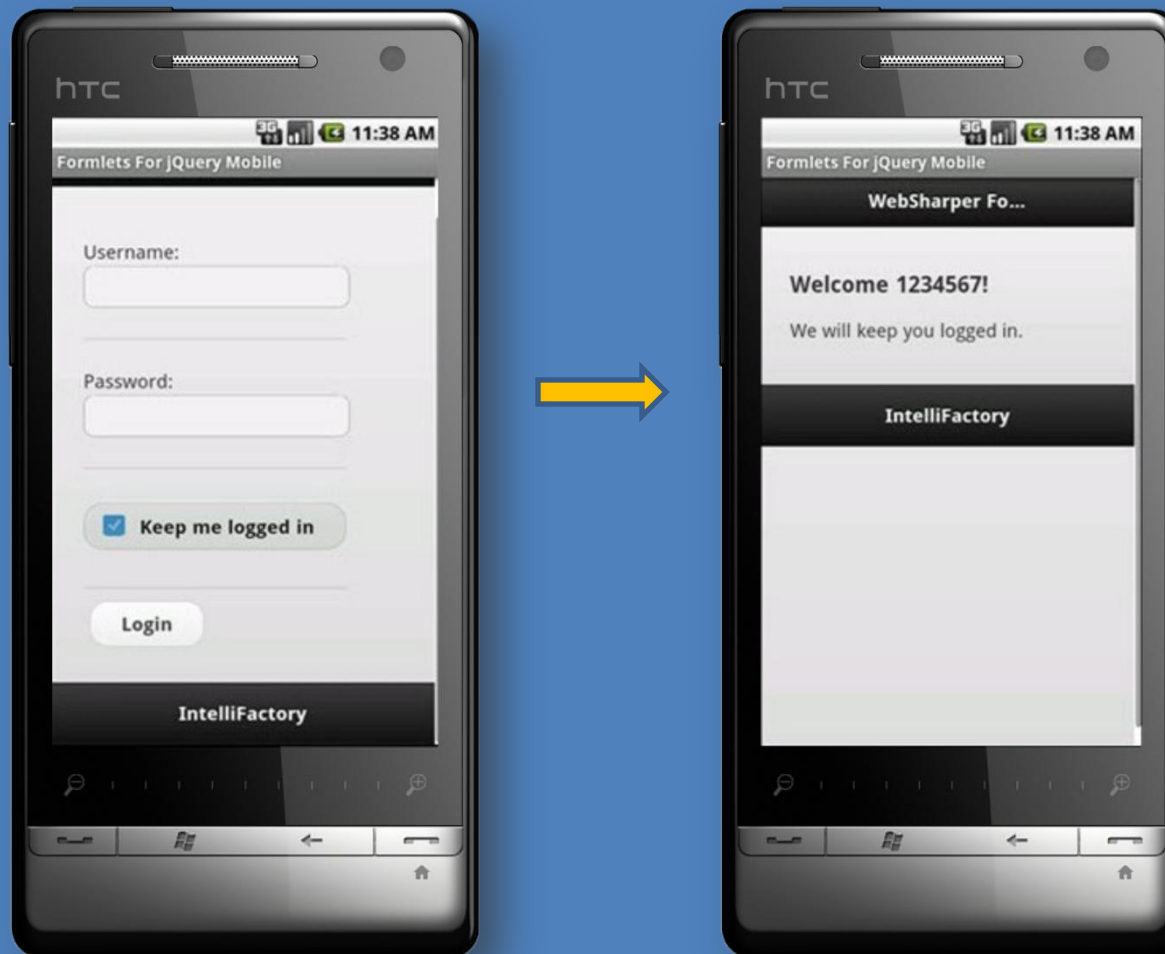
The image shows a web form titled "Add a New Person" with a white background and rounded corners. It contains two input fields: "Name" and "Email". Both fields have a red "X" icon to their right, indicating an error. Below the fields are two buttons: "Submit" and "Reset". The form is centered on a blue background.

- Több kurrens UI technológiára húzható:
 - jQuery UI
 - Yahoo UI
 - Ext JS
 - jQuery Mobile

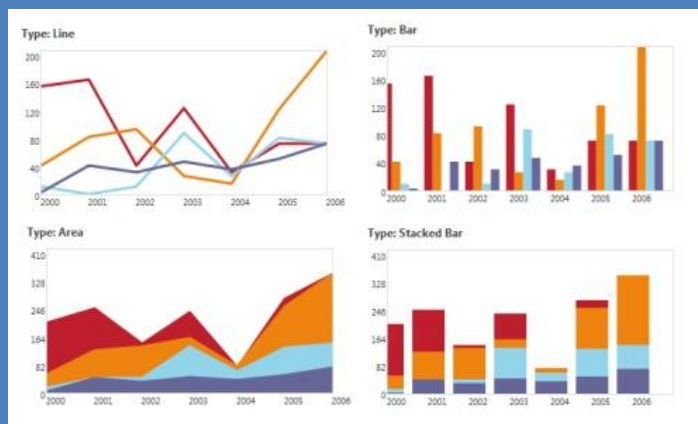
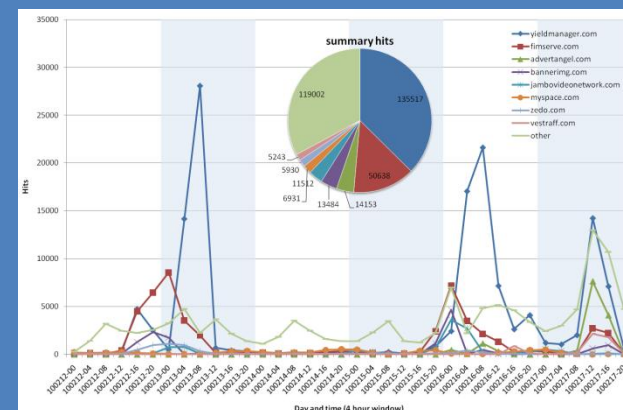
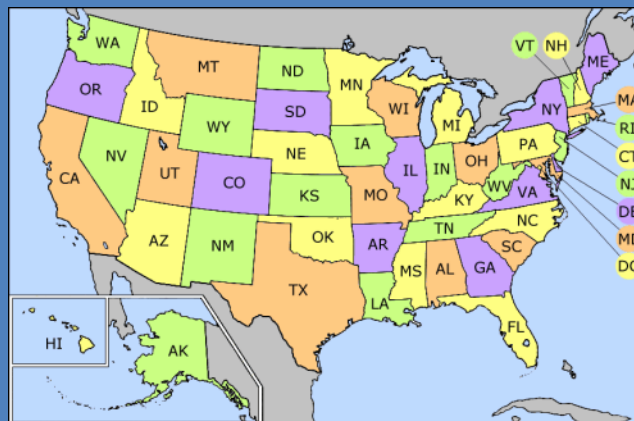
```
let loginSequenceF =
  Formlet.Do {
    let! username, password, remember =
      Formlet.Yield (fun user pass remember -> user, pass, remember)
    <*> (Controls.TextField "" Theme.C "Username: "
      |> Validator.IsNotEmpty "Username cannot be empty!")
    <*> (Controls.Password "" Theme.C "Password: "
      |> Validator.IsRegexMatch "^[1-4]{4,}[0-9]$" "The password is wrong!")
    <*> Controls.Checkbox true Theme.C "Keep me logged in "
      |> Enhance.WithSubmitButton "Log in" Theme.C
    let rememberText =
      if remember then "" else "not "
    do! Formlet.OfElement (fun _ ->
      Div [
        H3 [Text ("Welcome " + username + "!")]
        P [Text ("We will " + rememberText + "keep you logged in.")]
      ])
  }
|> Formlet.Flowlet
```

```
Div [HTML5.Attr.Data "role" "page"] -< [  
  Div [HTML5.Attr.Data "role" "header"] -< [  
    H1 [Text "WebSharper Formlets for jQuery Mobile"]>  
  ]  
  
  Div [HTML5.Attr.Data "role" "content"] -< [  
    loginSequenceF  
  ]  
  
  Div [HTML5.Attr.Data "role" "footer"] -< [  
    P [Attr.Style "text-align: center;"] -< [Text "IntelliFactory"]  
  ]  
]
```

Az eredmény:



- GIS
 - Google Maps
 - Bing Maps
- Vizualizáció
 - Infovis
 - Protovis
 - Google Visualization



```
type CurrentLocationControl() =  
  inherit Web.Control()  
  
[<JavaScript>  
  override this.Body =  
    let screenWidth =  
      JQuery.Of("body").Width()
```

```
  let MapOptions =  
    Bing.MapViewOptions(  
      Credentials = bingMapsKey,  
      Width = screenWidth - 10,  
      Height = screenWidth - 10,  
      Zoom = 16)
```

```
  let label = H2 []
```

```
  let setMap (map : Bing.Map) =  
    let updateLocation() =  
      // Gets the current location  
      let loc = Mobile.GetLocation()  
      // Sets the label to the current location  
      Rest.RequestLocationByPoint(<<your-bingmaps-key>>,  
        loc.Lat, loc.Long, ["Address"],  
        fun result ->  
          let locInfo =  
            result.ResourceSets.[0].Resources.[0]  
          label.Text <-  
            "You are currently at " +  
            JavaScript.Get "name" locInfo  
  
      // Adds a pushpin at the current location  
      let loc = Bing.Location(loc.Lat, loc.Long)  
      let pin = Bing.Pushpin loc  
      map.Entities.Clear()  
      map.Entities.Push pin  
      map.SetView(Bing.ViewOptions(Center = loc))
```

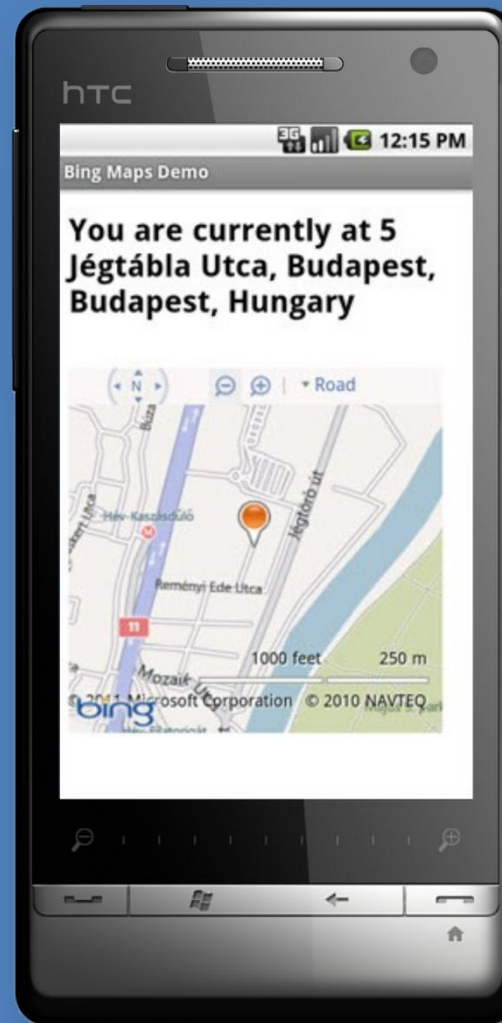
```
  // Keep updating your location regularly  
  JavaScript.SetInterval updateLocation 1000  
  |> ignore
```

```
  let map =  
    Div []  
    |>! OnAfterRender (fun this ->  
      // Renders a map control  
      let map =
```

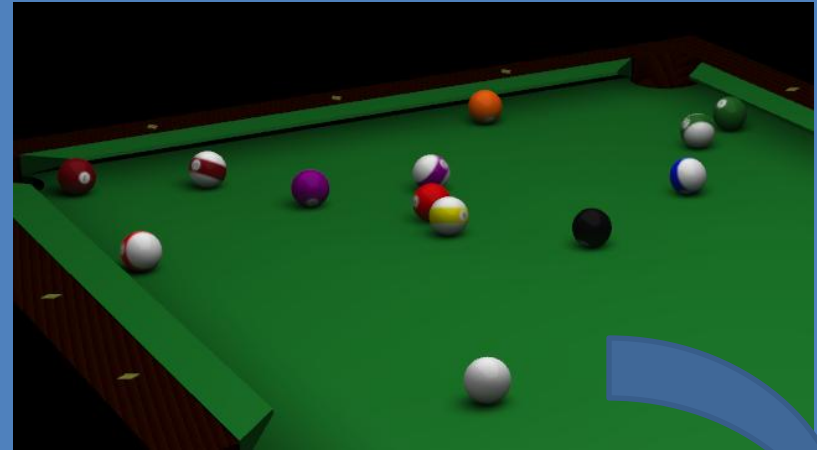
```
        Bing.Map(this.Body, MapOptions)
```

```
        map.SetMapType(Bing.MapTypeId.Road)  
        setMap map)
```

```
  // Returns markup for this control  
  Div [  
    label  
    Br []  
    map  
  ] :> _
```



- HTML5
 - WebGL
 - O3D
 - GIMatrix



F# + WebSharper a következőket nyújtja:

- Óriási **hatékonyságnövekedés**
- Egy **folyamatosan bővülő piac** lehetősége
- Gyors út **a többplatformos megoldásokhoz**
- Skálázhatóság **desktop-ra** és a **felhőbe**

Funkcionális programozás:

- Hatékonyabb absztrakciók
- Gyorsabb fejlesztés
- Rövidebb, fenntarthatóbb kódbázis





Kérdések?



Több információ itt:



<http://intellifactory.com>

<http://websharper.com>

<http://infoq.com/articles/WebSharper>

Infotér Konferencia 2011, Balatonfüred –
Nov 22-23, 2011