



Ocsigen
Applications
Web et mobiles client-serveur



Vincent Balat

Open Source Innovation Spring — Systematic — 12 mai 2016

BeSport



be sport®

10
APR

Soccer at Stade Yves-du-Manoir

Created by **Lex Fleming**
Soccer

Only invitees can see this event. 

Lex Fleming invited you to this event.

RESPOND ▼

FOLLOW

🕒 07:00 – 08:00, Sunday 10 Apr 2016

📍 Stade Yves-du-Manoir, 12 Rue François Faber, 92700 Colombes

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam tempor justo congue, bibendum velit interdum. Quisque felis magnorem ipsum dolor orum ipe ... [Read more](#)

Related to: **Friendly soccer season 2016**



Write a post...

Posts

Media



Nothing yet! **Add a post**

20 Participating



2
Maybe

20
Pending

25
Followers

Latest Photos



Similar Events

12
APR

Soccer at the mount
Created by **Anna Bull**
Soccer

VIEW



Anna Bananya

23 years old | Paris, France



24 shared connection points >

Tennis Rugby Soccer Basketball American Football Badminton +23

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam tempor justo congue, bibendum velit interdum. Quisque felis magnorem ipsum dolor orem ipe ... [Read more](#)

Featured Achievements



2 - 1

Anna Bananya vs Georgia Stai...
Saturday 6 January, 2014
Tennis



Semi-finalist

Australian Open
Friday 5 January, 2014
Tennis

+ CONNECT

FOLLOW



430 Connections

34 Followers



459



23k



3



2503



208

Next Events

Auckland Blues
Southland

Following | Today 19:30

Football with the gang

Feed

Events

Network

Media

Achievements



Lex Fleming posted:

Today at 17:29

Can't wait for the next game of footy, let's go!

♥ 25 💬 10 ➦ 5



Good sport



Comment



Share

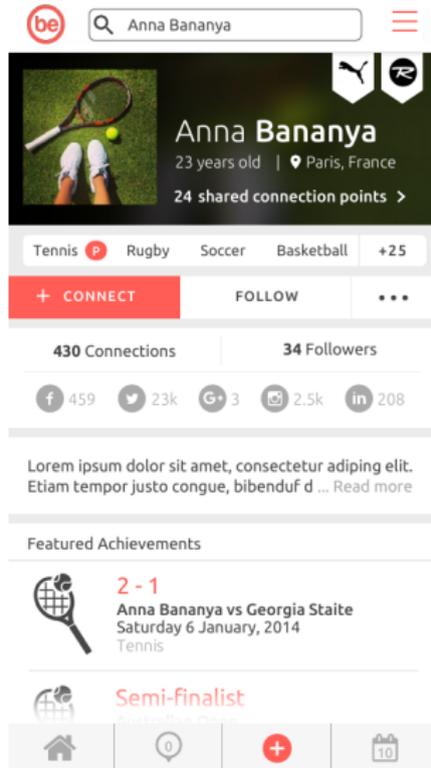


Lex Fleming added an image:

Today at 14:34



BeSport : Application mobile



BeSport : les besoins

- Développement d'une application client-serveur très dynamique
- Application mobile Android, iPhone, Windows,...
- Moyens limités (start-up)
- Fonctionnalités très riches
- Produit évoluant très rapidement (prototype)

Importance du choix de la technologie

Ocsigen : principale caractéristiques

Un ensemble d'outils pour le développement Web en OCaml.

- **Typage statique** pour vérifier de nombreuses propriétés

~> Fiabilité

~> *Refactoring* du code facile

~> Très peu de tests unitaires à écrire

- Une application **client-serveur** 100% en OCaml

~> La puissance d'OCaml sur le navigateur

~> Un seul langage pour les développeurs

~> Génération des pages côté serveur ou client

~> Communication client-serveur très simple

- **NOUVEAU ! Multi-plateforme Web & mobile**

~> Une seule application à développer

Le projet Ocsigen

créé en 2004, plus de 200 000 lignes de code., LGPL

Auteurs:

Vincent Balat, Jérôme Vouillon,

Gabriel Radanne, Vasilis Papavasileiou,

Pierre Chambart, Grégoire Henry, Benedikt Becker, Benjamin Canou, Boris Yakobowski,
Hugo Heuzard, Jérémie Dimino,

Charly Chevalier, Enguerrand Decorne, Romain Calascibetta, Jacques-Pascal Deplaix, Grégoire Lionnet,
Raphaël Proust, Stéphane Glondu, Jérôme Maloberti, Gabriel Kerneis, Arnaud Parant, Christophe
Lecointe, Denis Berthod, Gabriel Cardoso, Piero Furiesi, Jaap Boender, Baptiste Strazzulla, Thorsten
Ohl, Gabriel Scherer, Séverine Maingaud, Simon Castellan, Jean-Henri Granarolo, Archibald Pontier,
Nataliya Guts, Cécile Herbelin, Charles Oran, Jérôme Velleine, Pierre Clairambault ...



The screenshot displays a web application interface for 'BeSport'. At the top, there is a navigation bar with a search bar and a 'Version beta' label. Below this, a user profile for 'William Le Ferrand' is shown, including a profile picture, name, age (26 ANS), and a list of sports interests: Road Running, Urban soccer, Ultimate, Speed flying, Fencing, Sailing, Trailrunning, Tennis, Chess, Stroef hiking, RUNNING, Ice Skating, and Waberg. The profile also indicates '416 AMIS', '33 PARTENAIRES', '00 HEROES', and '18 FANS'. A vertical blue bar on the right side of the profile is labeled 'BEER communication'. Below the profile, there is a section for 'ACTIVITE TROPHÉES' with options to 'POSTER UNE PHOTO OU UNE VIDEO' and 'POSTER UN MESSAGE'. A social feed entry shows a post by 'Aude Hoffelner et William Le Ferrand' with a profile picture and a 'Commenter' button. On the left side of the interface, there is a sidebar with navigation options: 'cette semaine' (10/04/13 - 16/04/13), 'TOUS MON RESEAU', 'MOI', 'PARTENAIRES', 'HEROS', 'FANS', 'AMIS', 'TOUS LES SPORTS', and 'MES SPORTS'. A search bar is located at the bottom of the sidebar.

facebook



Entreprises et projets libres :

BeSport, NYU CGSB Genomics Core, Pumgrana, Facebook, Accret.io, Life.tl, Ashima Arts, Metaweb/Freebase, Hypios, Ocamlcore, Ocamlpro, Baoug, Nleyten...

Quelques projets Ocsigen

ocsigen
fun of in web programming
eliom

More information

Write client/server Web applications in very few lines of OCaml code!

ocsigen
fun of in web programming
js_of_ocaml

More information

An OCaml to Javascript compiler.

ocsigen
fun of in web programming
lwt

More information

A cooperative threading library for OCaml.

ocsigen
fun of in web programming
server

More information

A full-featured and extensible Web server.



And many other projects ...

Typage statique pour le Web

Un système de type puissant peut éviter beaucoup de bugs, par exemple :

- Html : Respect des recommandations du W3C
- Validité des liens et formulaires
- Interactions bien typées avec le DOM
- Requêtes bien typées aux bases de données

Exécuter des programmes OCaml dans un navigateur



Un compilateur qui traduit OCaml vers JS

Exécuter des programmes OCaml dans un navigateur



Un compilateur qui traduit **du bytecode** OCaml vers JS

Exécuter des programmes OCaml dans un navigateur



Un compilateur qui traduit **du bytecode** OCaml vers JS

Fonctionne avec n'importe quel programme OCaml

- même s'il utilise des bibliothèques dont vous n'avez pas le source
- sauf s'il utilise des threads préemptifs (mais on a Lwt)
- à condition d'implémenter en JS toutes fonctions externes

Écriture d'interfaces avec JavaScript très simple (en bien typée).

Exemple de programme client-serveur

```
[XkShared
]
name Eliom_pervasives
name HTML5_M
let width = 700
let height = 400
type messages = (string * int * (int * int) * (int * int)) deriving (Json)
]

module My_appl = Eliom_output.Eliom_appl (struct
  let application_name = "groffit1"
  end)

let b = Eliom_bus.create ~name:"groff" Json.t:messages)

[XkClient
]
open Event_arrows
let draw ctx (color, size, (x1, y1), (x2, y2)) =
  ctx##strokeStyle <- (Js.string color);
  ctx##strokeWidth <- float size;
  ctx##beginPath();
  ctx##moveTo(float x1, float y1); ctx##lineTo(float x2, float y2);
  ctx##stroke()
]

let main_service = My_appl.register_service ~path:[""] ~get_params:Eliom_parameters.unit
(fun () () -> Eliom_services.onload

[XkClient
]
let canvas = Dom_html.createCanvas Dom_html.document in
let ctx = canvas##getContext (Dom_html_id ) in
canvas##width <- width; canvas##height <- height;
ctx##lineCap <- Js.string "round";
Dom.appendChild Dom_html.document##body canvas;

let slider = jsnew Goog.Ui.slider(Js.null) in
slider##setMinimum(1.); slider##setMaximum(80.);
slider##render(Js.some Dom_html.document##body);

let pSmall = jsnew Goog.Ui.hsvPalette
(Js.null, Js.null, Js.some (Js.string "goog-hsv-palette-se")) in
pSmall##render(Js.some Dom_html.document##body);

let x = ref 0 and y = ref 0 in
let set_coord ev =
  let x0, y0 = Dom_html.elementClientPosition canvas in
  x := ev##clientX - x0; y := ev##clientY - y0 in
let compute_line ev =
  let oldx = !x and oldy = !y in
  set_coord ev;
  let color = Js.to_string (pSmall##getColor()) in
  let size = int_of_float (Js.to_float (slider##getValue())) in
  (color, size, (oldx, oldy), (x, y))
in
let (b : messages Eliom_bus.t) = b in
let line ev =
  let v = compute_line ev in
  let _ = Eliom_bus.write b v in
  draw ctx v
in
ignore (let_stream.iter (draw ctx) (Eliom_bus.stream b));
ignore (run (mousemoves canvas
  (arr (fun ev -> set_coord ev; line ev)
  >>> first [mousemove Dom_html.document (arr line);
  mouseup Dom_html.document >>> (arr line)]))));

let p = pSmall in
let (head
  (title (pcdata "Groffit1"))
```

Exemple de programme client-serveur

```
[%shared
new: Elion_pervasives
new: HTML5_M
let width = 700
let height = 400
type messages = (string * int * (int * int) * (int * int)) deriving (Json)
]

module My_appl = Elion_output.Elion_appl (struct
  let application_name = "groffit1"
end)

let b = Elion_bus.create ~name:"groff" Json.t:messages)

[%client
open Event_arrows
let draw ctx (color, size, (x1, y1), (x2, y2)) =
  ctx##strokeStyle <- (Js.string color);
  ctx##lineWidth <- float size;
  ctx##beginPath();
  ctx##moveTo(float x1, float y1); ctx##lineTo(float x2, float y2);
  ctx##stroke()
]

let main_service = My_appl.register_service ~path:[""] ~get_params:Elion_parameters.unit
(fun () () -> Elion_services.onload

[%client
let canvas = Dom_html.createCanvas Dom_html.document in
let ctx = canvas##getContext (Dom_html_id ) in
canvas##width <- width; canvas##height <- height;
ctx##lineCap <- Js.string "round";
Dom.appendChild Dom_html.document##body canvas;

let slider = jsnew Goog.Ui.slider(Js.null) in
slider##setMinimum(1.); slider##setMaximum(80.);
slider##render(Js.some Dom_html.document##body);

let pSmall = jsnew Goog.Ui.hsvPalette
(Js.null, Js.null, Js.some (Js.string "goog-hsv-palette-se")) in
pSmall##render(Js.some Dom_html.document##body);

let x = ref 0 and y = ref 0 in
let set_coord ev =
  let x0, y0 = Dom_html.elementClientPosition canvas in
  x := ev##clientX - x0; y := ev##clientY - y0 in
let compute_line ev =
  let oldx = !x and oldy = !y in
  set_coord ev;
  let color = Js.to_string (pSmall##getColor()) in
  let size = int.of_float (Js.to_float (slider##getValue())) in
  (color, size, (oldx, oldy), (!x, !y))
in
let (b : messages Elion_bus.t) = %b in
let line ev =
  let v = compute_line ev in
  let _ = Elion_bus.write b v in
  draw ctx v
in
ignore (let_stream.iter (draw ctx) (Elion_bus.stream b));
ignore (run (mousemoves canvas
  (arr (fun ev -> set_coord ev; line ev)
    >>> first [mousemoves Dom_html.document (arr line);
              mouseup Dom_html.document >>> (arr line)]))));
let _ =
  (head
   (head
    (title (pcdata "Groffit1"))
```

[%shared ...]

[%server ...]

[%client ...]

Utiliser des valeurs serveur côté client

```
let b = ...
```

```
%%Server
new Elion_pervasive
new HTML5.M
let width = 700
let height = 400
type messages = (string * int * (int * int) * (int * int)) deriving (Show)
}

module My_appl = Elion_pervasive.Elion_app (struct
  let application_name = "graffiti"
end)

let b = Elion_bus.create ~name:"graff" %son.tmessages)

%%Client
open Event_arrows
let draw ctx (color, size, (x1, y1), (x2, y2)) =
  ctx#strokeStyle <- %s.string color;
  ctx#lineWidth <- %float size;
  ctx#beginPath();
  ctx#moveTo(float x1, float y1); ctx#lineTo(float x2, float y2);
  ctx#stroke()
}

let main_service = My_appl.register_service ~path:[""] ~get_params:Elion_parameters.unit
(fun () {} -> Elion_services.onload

[%%Client
  let canvas = Dom_html.createCanvas Dom_html.document in
  let ctx = canvas#getContext (Dom_html._id_1 in
  canvas#width <- width; canvas#height <- height;
  ctx#lineCap <- %s.string "round";
  Dom.appendChild Dom_html.document#body canvas;

  let slider = jsnew Goog.Ul.slider(%s.null) in
  slider#setMinimum(1.); slider#setMaximum(50.);
  slider#render(%s.some Dom_html.document#body);

  let pSmall = jsnew Goog.Ul.hsvPalette
    (%s.null, %s.null, %s.some (%s.string "goog-hsv-palette-sa")) in
  pSmall#render(%s.some Dom_html.document#body);

  let x = ref 0 and y = ref 0 in
  let set_coord ev =
    let x0, y0 = Dom_html.elementClientPosition canvas in
    x := ev#clientX - x0; y := ev#clientY - y0 in
  let compute_line ev =
    let oldx = !x and oldy = !y in
    set_coord ev;
    let color = %s.to_string (pSmall#getColor()) in
    let size = %float (js.to_float (slider#getValue())) in
    (color, size, (oldx, oldy), (x, y))
  in
  let (b : messages Elion_bus.t) = %b in
  let line ev =
    let v = compute_line ev in
    let _ = Elion_bus.write b v in
    draw ctx v
  in
  ignore (Lwt_stream.iter (draw ctx) (Elion_bus.stream b));
  ignore (run (mousedown canvas
    (arr (fun ev -> set_coord ev; line ev)
    >>> first [mousedown Dom_html.document (arr line);
    mouseup Dom_html.document >>> (arr line)])) ());
]; Lwt.return
```


Utiliser des valeurs serveur côté client

```
[%%Server]
open Elixir_pervsives
open HTML5.M
let width = 700
let height = 400
type messages = (string * int * (int * int) * (int * int)) deriving (Show)
}

module My_appl = Elixir_routes.Elixir_appl(struct
  let application_name = "graffiti"
end)

let b = Elixir_bus.create ~name:"graff" ~son.t:messages}

[%%Client]
open Event_arrows
let draw ctx (color, size, (x1, y1), (x2, y2)) =
  ctx#strokeStyle <- (3; string color);
  ctx#lineWidth <- float size;
  ctx#beginPath();
  ctx#moveTo(float x1, float y1); ctx#lineTo(float x2, float y2);
  ctx#stroke()
}

let main_service = My_appl.register_service ~path:[""] ~get_params:Elixir_parameters.unit
(fun () {} -> Elixir_services.onload

[%%Client]
let canvas = Dom_html.createCanvas Dom_html.document in
let ctx = canvas#getContext (Dom_html_id) in
canvas#width <- 700;
canvas#height <- 400;
ctx#lineCap <- (1; string "round");
Dom.appendChild (Dom_html.document#body) canvas;

let slider = js%> Dom_html.getElementById "slider" in
slider#setMinValue 0; slider#setMaxValue 100;
slider#render();

let pSmall = js%> Dom_html.getElementById "pSmall" in
pSmall#setStyle (["color", "red", "font-size", "12px"]);

let x = ref 0 and y = ref 0 in
let set_coord ev =
  let x0, y0 = Dom_html.elementClientPosition canvas in
  x := ev#clientX - x0; y := ev#clientY - y0 in
let compute_line ev =
  let oldx = !x and oldy = !y in
  set_coord ev;
  let color = !ctx.strokeStyle in
  let size = int_of_float (3; to_float (slider#getValue())) in
  (color, size, (oldx, oldy), (x, y))
in
let (b : messages Elixir_bus.t) = b in
let line = "" in
let v = compute_line ev in
let v = Elixir_bus.write b v in
draw ctx v
in
ignore (Lwt_stream.iter (draw ctx) (Elixir_bus.stream b));
ignore (run (mousedown canvas
  (arr (fun ev -> set_coord ev; line ev)
    >>> first [mousedown Dom_html.document (arr line);
      mouseup Dom_html.document >>> (arr line)])) ());
}; Lwt.return

[%%HTML]
<script src="/js/graffiti.js" type="text/javascript"></script>
<script src="/css/style.css" type="text/css"></script>
<script src="/js/graffiti_closure.js" type="text/javascript"></script>
</script>
</html>
```

let b = ...

~%b ...

Si b est une fonction :
Appel d'une
fonction côté serveur !

Pages réactives client-serveur

Des pages qui se mettent à jour toutes seules

quand les variables dont elles dépendent changent.

Basé sur la *programmation fonctionnelle réactive*.

- Presque exactement le même code que pour des pages statiques
 ~> Réduction importante du nombre de lignes à écrire !
- Pas besoin de DOM virtuel !
- Possibilité de générer ces pages réactives **côté serveur** !

Applications mobiles

BIENTÔT DANS OCSIGEN ELIOM 6.0

1 seul code pour les applications Web et mobile

Application mobile exécutée dans une Webview (Android, iPhone, Windows...)

Démarrage de l'application côté client (mobile) ou serveur (Web)

Pages générées **côté serveur** (première page de l'application Web, requêtes HTTP)
ou **côté client** (pages suivantes, ou mobile)

~> Votre appli mobile multi-plateforme sans une ligne de plus

~> Bonnes performances

⚠ L'ordre d'exécution est différent

Beaucoup d'autres fonctionnalités

- Mécanisme d'identification de services (Très simplifié dans **ELIOM 6.0**)
- Sessions avancées avec scope (browser, tab, user ...)
- Communication serveur vers client
- **BIENTÔT Eliom-base-app**: une bibliothèque de haut niveau et un template d'application clé en main
- ...

BeSport : l'entreprise



- Créée en octobre 2015
- 6 développeurs
- Très concis ! (50 000 lignes de code), 100% OCaml

BeSport : retour d'expérience

Bilan de l'utilisation d'Ocsigen :

- + Ça marche :) Malgré des changements incessants du produit.
- + Recrutement facilité !
 - Bibliothèque de widgets client-serveur à étoffer.
 - Manque une bonne bibliothèque pour le bases de données.

BeSport est le premier contributeur externe au projet libre Ocsigen.