



# Paris JUG



Sponsors  
**Platinum**

[www.parisjug.org](http://www.parisjug.org)



Sponsors  
**Gold**





[www.parisjug.org](http://www.parisjug.org)



## Backbone.js en environnement Java

Laurent Dutheil  
Frederic Merizen  
Consultants  
OCTO Technology



# « Comment une équipe Java développe une application JavaScript et l'intègre dans son environnement... »

# Intervenants

## ▪ Laurent DUTHEIL

- Consultant chez OCTO Technology depuis 2010
- Diplômé de l'université de Lille I en Intelligence Artificielle et Génie Logiciel
- Forte expérience des architectures JEE : audit d'architecture, étude de cadrage, supervision métier, tech-lead, Proof of Concept
- Rompu aux méthodes agiles

# Intervenants

## ▪ Frederic MERIZEN

- Consultant chez OCTO Technology depuis 2011
- Diplômé de Supélec
- Technical leader, développeur et agilisateur sur des mission Java et Front
- Co-anime la communauté des technical leaders chez OCTO
- Membre de l'équipe d'organisation de la conférence Agile France

# Motivation

- **Même pour écrire un nouvelle application, on part très rarement d'une page blanche**
  - Équipe en place
  - Environnements de développement
  - Applications existantes
- **Tirer parti des forces en place pour innover**



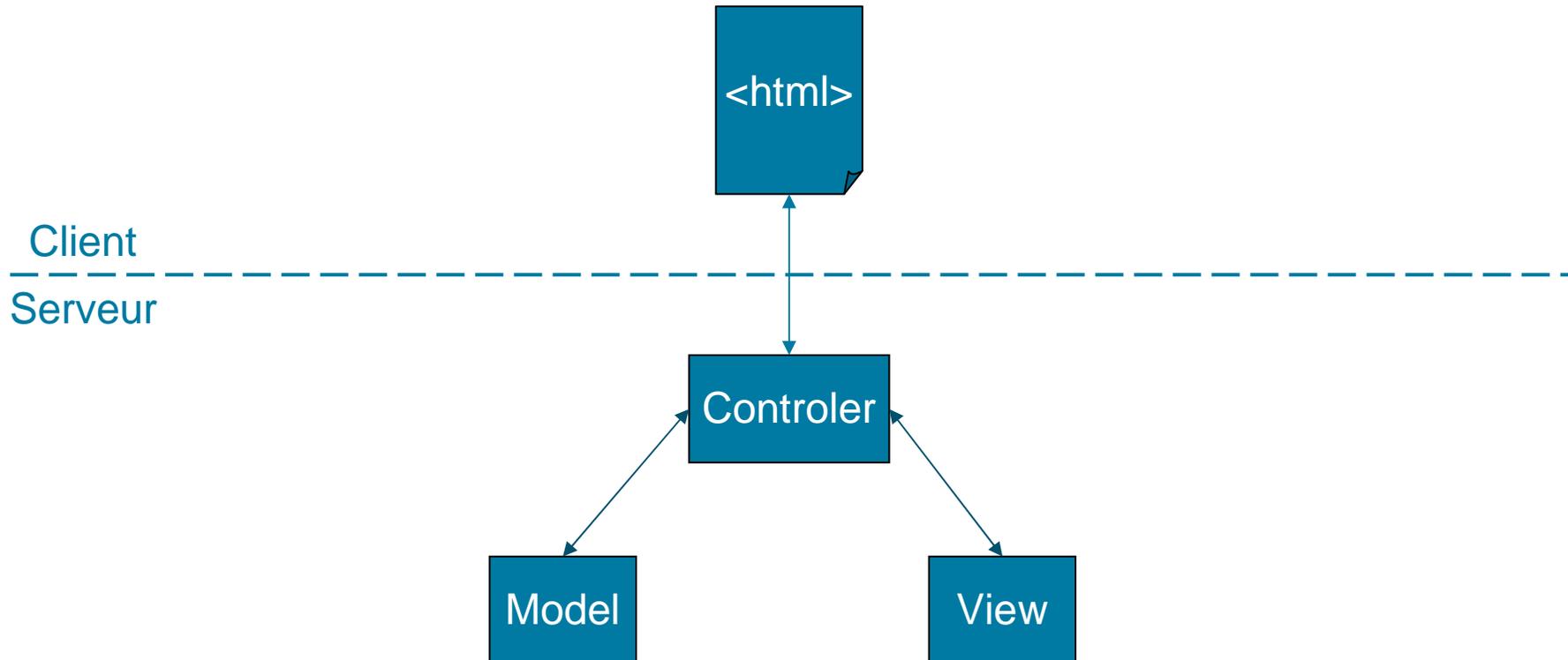
[www.parisjug.org](http://www.parisjug.org)

# ***Historique des application web***

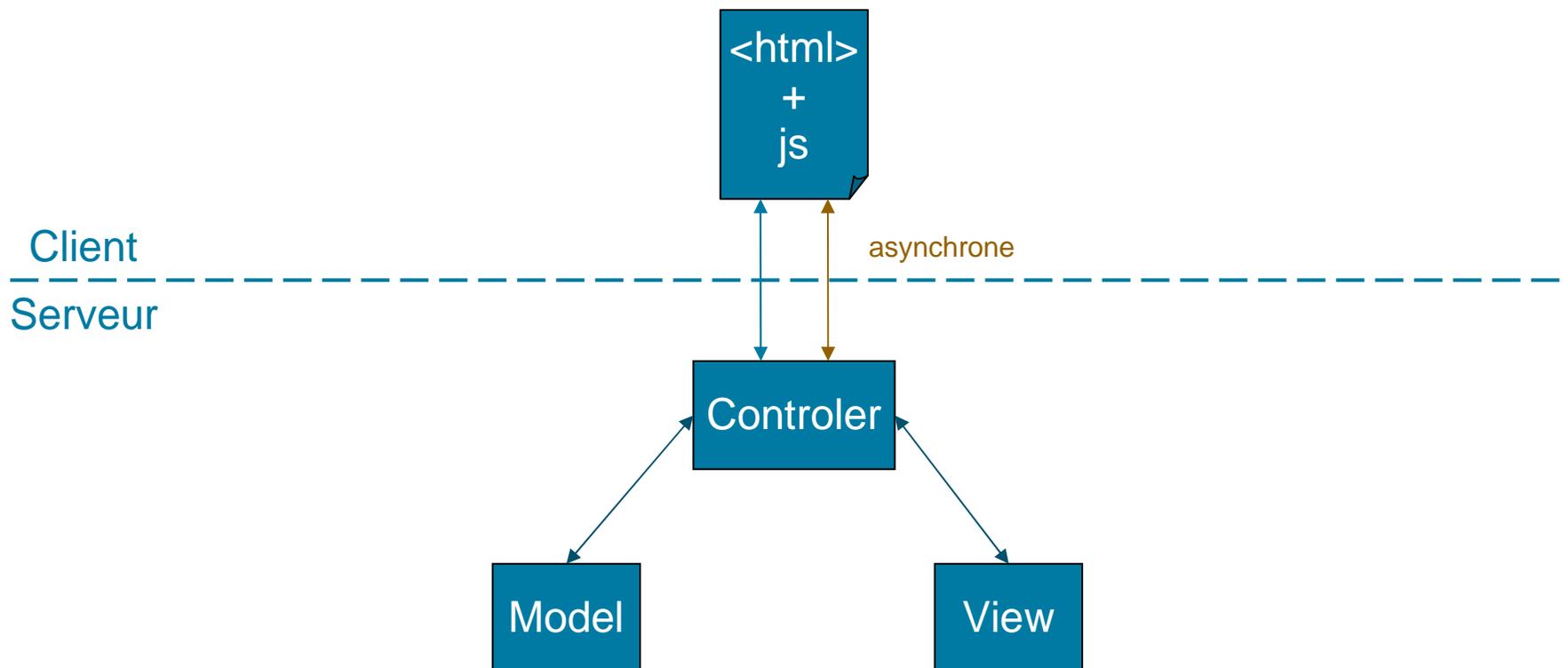
***Toujours plus riche et réactif***



# MVC côté serveur



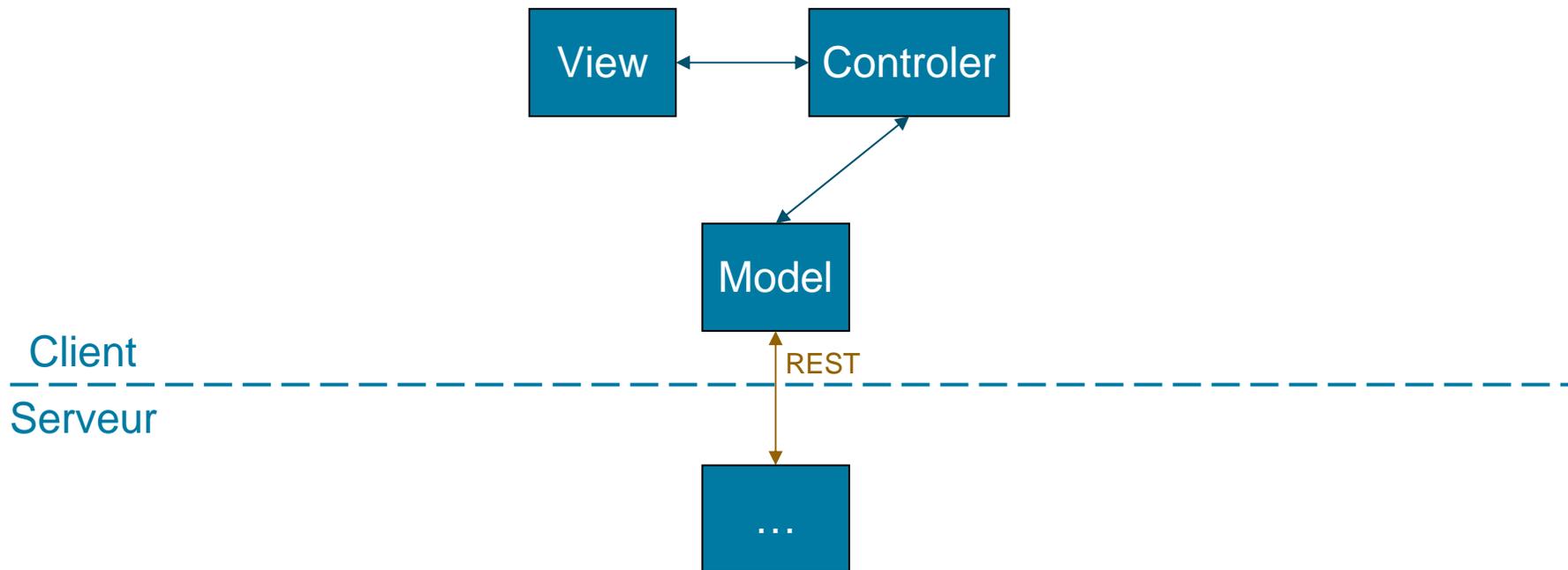
# Ajax



**jQuery**



# MVC côté client





[www.parisjug.org](http://www.parisjug.org)

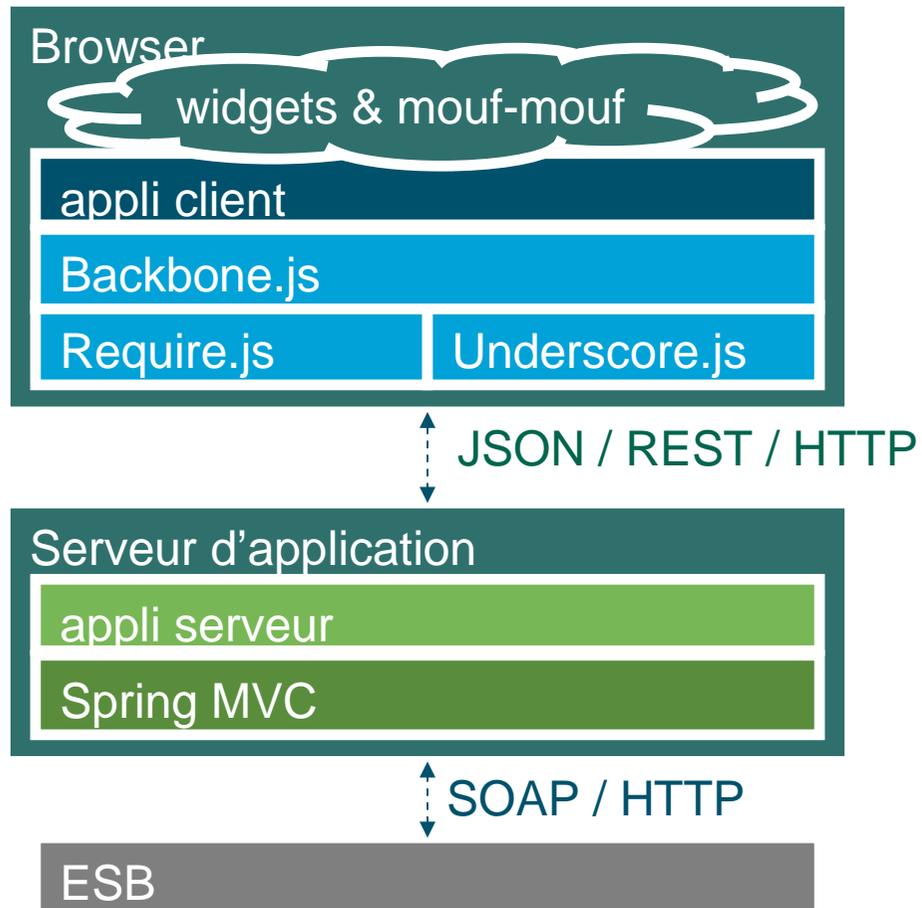
# *L'application*

<https://github.com/fredZen/java-backbone-demo>

*(d emo)*



# Une stack efficace avec Java



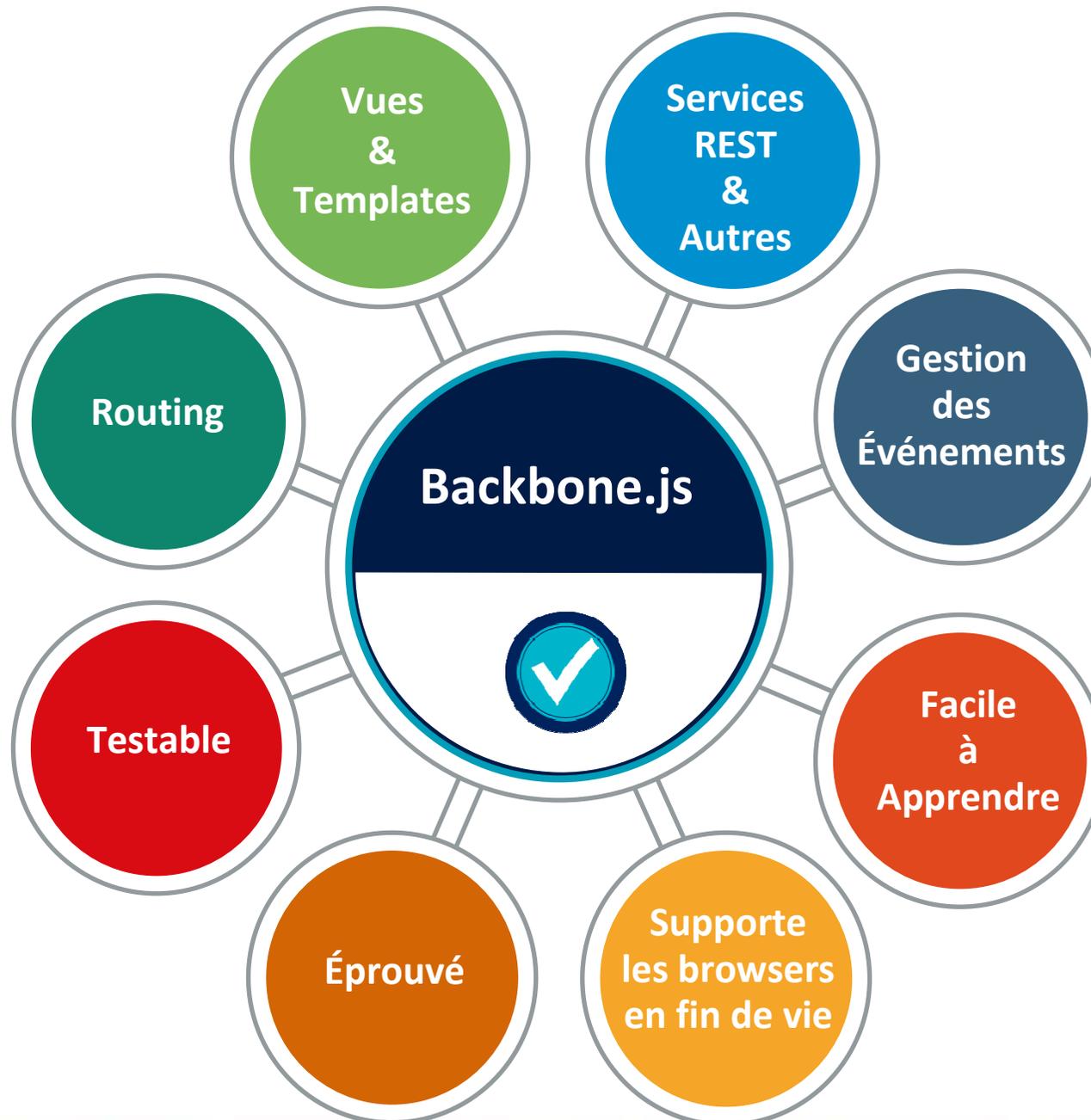


[www.parisjug.org](http://www.parisjug.org)

# *Côté client*



# Pourquoi Backbone.js



# Exploiter le templating

- **Variabiliser le html avec un langage de templates**
  - Un moteur est inclus dans Underscore
  - D'autres : Mustache, Handlebars, ...

# Exploiter le templating

## • Tips : Associer une Vue à chaque template

```
var MyView = Backbone.View.extend({
  el: "#container",
  ...
  render: function() {
    var content = //votre méthode de templating
    this.$el.html(content);
  },
  ...
});
```

# Synchroniser vue et modèle

## Utiliser les événements backbone

```
var MyView = Backbone.View.extend({
  ...
  setModel: function(model) {
    this.model = model;
    this.model.on("change", this.modelChanged);
  },
  events: {
    "change input, select, textarea": "viewChanged",
  },
  viewChanged: function(event) {
    //mise à jour du model
    ...
  },
  modelChanged: function() {
    //mise à jour de la vue
    ...
  },
  ...
});
```

# Synchroniser vue et modèle

- **Systeme d'événement très flexible**
  - Pas restreint à la vue : un modèle peut observer un modèle, etc...
  - Permet un couplage lâche
  - Tips : Penser à mettre en place un bus d'événements

# Communiquer avec le serveur

## ▪ La synchro backbone en 1 ligne !

```
var MyView = backbone.View.extend({  
  ...  
    submit: function() {  
      this.model.save();  
    },  
  ...  
});
```

- model.fetch => GET
- model.save => POST
- model{id}.save => PUT
- model.destroy => DELETE

```
var myModel = backbone.Model.extend({  
  urlRoot: baseUrl + "/myResource",  
  ...  
});
```

# Communiquer avec le serveur

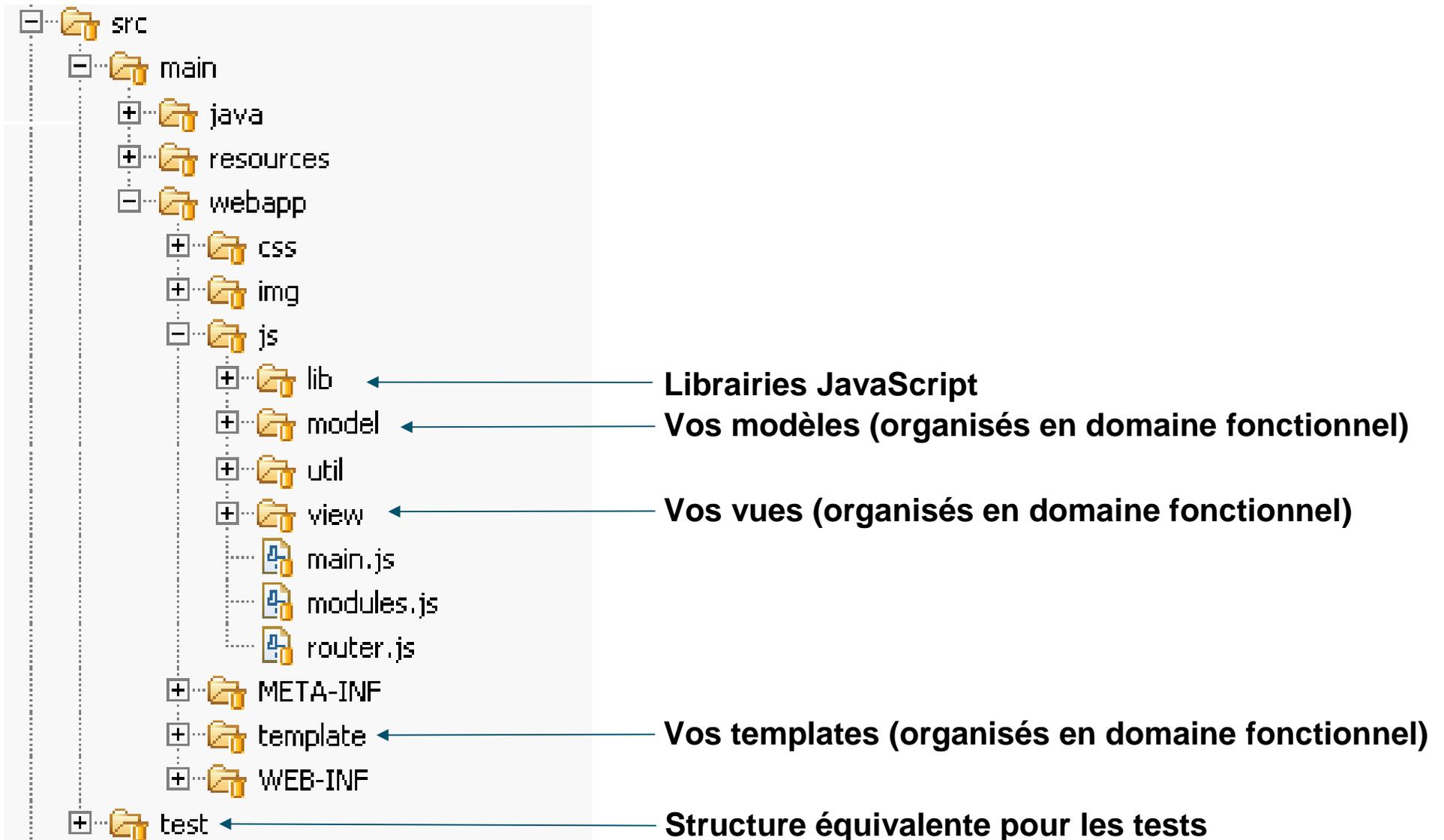
## • Flexibilité de customiser si nécessaire

```
var ProposalModel = backbone.Model.extend({
  ...
  parse: function(response) {
    return null;
  },
  ...
});
```

```
var MyView = backbone.View.extend({
  ...
  submit: function() {
    this.model.save({}, {
      success: this.saveSucceeded,
      error: this.saveError
    });
  },
  ...
  saveSucceeded: function(model, response) { //traitement de la réponse
    ...
  },
  saveError: function(model) { //traitement de l'erreur
    ...
  },
});
```

# Organiser son application

## Organiser ses fichiers dans des répertoires



# Organiser son application

## Requirejs

- Simule l'import Java
- Permet l'injection de dépendances



```
define([ 'backbone', 'jquery', 'underscore' ],  
function(Backbone , $ , _ ) {  
    var ErrorModel = Backbone.Model.extend({  
        ..  
    });  
  
    return new ErrorModel();  
});
```

# Organiser son application

## ▪ Centralisation des dépendances



```
require.paths = {
  backbone: 'lib/backbone/backbone.min',
  underscore: 'lib/lodash/lodash.min',
  jquery: 'lib/jquery/jquery.min',
  json2: 'lib/json2',
  ...
};

require.shim = {
  'backbone': {
    deps: ['underscore', 'jquery', 'json2'],
    exports: 'Backbone'
  },
  ...
};
```

# Retrouver ses repères java

```
define([ 'jquery', 'underscore', 'backbone', 'router' ],  
function($, _, backbone, router) {  
    var singleton = null,  
        MyView = Backbone.View.extend({ ← Classe  
    ...  
        setModel: function(model) { ← Méthode  
            this.model = model;  
            this.model.on("change", this.modelChanged);  
        },  
    ...  
    }, {  
        showMe: function() { ← Méthode statique  
            router.navigate("myPage");  
            singleton.setModel(new MyModel());  
            singleton.render();  
        }  
    });  
    singleton = new MyView();  
    return MyView;  
});
```

# Gérer les URLs

- Le routeur ventile les URLs entrantes vers les vues

```
var AppRouter = Backbone.Router.extend({
  routes: {
    // Pages
    'myPage': 'myPage',
    ...
  },

  init: function() {
    Backbone.history.start({
      root : baseUrl,
      pushState: true ← Gestion transparente des
                        hashtags vs pushstate
    });
  },

  ...

  myPage: function() {
    require([ 'view/myView' ], function(MyView) {
      MyView.showMe();
    });
  },

  ...

  return AppRouter;
});
```

# Gérer les URLs

- Les vues doivent mettre à jour l'URL

```
MyView = Backbone.View.extend({
  ...
}, {
  showMe: function() {
    router.navigate("myPage");
    ...
  }
});
```



[www.parisjug.org](http://www.parisjug.org)

# *Côté serveur*



# Spring MVC sur le serveur

## ▪ Spring MVC pour s'interfacier au SI

- Couche d'adaptation avec le back (Web Services, Hibernate, ...)
- Exposer des services REST en JSON avec Jackson
- Hybrider REST et RPC si le back est trop procédural
- Gérer la sécurité

# Spring MVC sur le serveur

```
@Controller
public class MyController extends StandardExceptionHandler {

    @RequestMapping(value = "/myResource", method = RequestMethod.POST)
    @ResponseBody
    public MyResource persiste(@RequestBody MyResource myResource) {
        ...
    }

    @RequestMapping(value = "/proposal/{id}", method = RequestMethod.GET)
    @ResponseBody
    public MyResource getById(@PathVariable Long id) {
        ...
    }

    ...
}
```

# Spring MVC sur le serveur

## ▪ **Tips : pour migrer en douceur**

- « MVC client » aux parties très dynamiques du site
- « MVC serveur » pour les parties plus statiques  
facilite le SEO

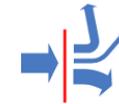
# Une page conteneur

## • Une JSP

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set value="${pageContext.request.contextPath}/js" var="js" />
<c:set value="${pageContext.request.contextPath}/resource" var="resource" />
<c:set value="${pageContext.request.contextPath}/monAppli" var="monAppli" />
...
    <div id="container"/>
...
<script>
    var require = {
        baseUrl: '${js}',
        config: {
            'util/resource': {
                baseUrl: '${resource}'
            },
            'router': {
                baseUrl: '${monAppli}'
            }
        }
    };
    <%@ include file="../../js/modules.js"%>
</script>
<script data-main="main" src="${js}/Lib/require/require.min.js"></script>
```

# Une page conteneur

## ▪ **UrlRewriteFilter**



*UrlRewriteFilter*

- Toutes les URL utilisateur aboutissent à la même JSP
- Ajouter UrlRewriteFilter dans web.xml
- Paramétrer urlrewrite.xml

```
<urlrewrite default-match-type="wildcard">
...
  <rule>
    <from>/monAppli/**</from>
    <to>/WEB-INF/jsp/monAppli.jsp</to>
  </rule>
</urlrewrite>
```



[www.parisjug.org](http://www.parisjug.org)

# *JavaScript à l'ère industrielle*



# Tests unitaires avec Jasmine

```
localhost:8234

Jasmine 1.2.0 revision 1337005947

.....

Running 48 of 48 specs

fetchPlugin
  require le modele

un modele avec des sous-modeles
  set
    appelle le wrapper quand on lui passe un objet nu
    sait wrapper un objet isolé dans un tableau
    marche aussi quand on sette plusieurs propriétés en même temps
    laisse les autres propriétés tranquilles
    retourne true quand il a setté une propriété
    ne rewrappe pas un objet déjà wrappé

une collection de propositions
  getPremierSejour
    retourne le premier sejour de la première proposition

router
  navigate
    change le texte

DateHelper
  getAge
    calcule l'age lors de la date anniversaire
    calcule l'age lors du mois anniversaire
    calcule l'age lors du mois anniversaire après la date anniversaire
    calcule l'age après la date anniversaire
    calcule l'age avant la date anniversaire
```

```
define(['util/dateHelper'],
function(DateHelper) {
  describe("DateHelper", function() {
    describe("getAge", function() {
      it("calcule l'age lors de la date anniversaire", function() {
        // given
        var today = new Date("2012","2","20"); // 20/03/2012
        var dateNaissance = ["1974","3","20"];

        // when
        var age = DateHelper.getAge(dateNaissance, today);

        // then
        expect(age).toBe(38);
      });

      it("calcule l'age lors du mois anniversaire", function() {
        // given
        var today = new Date("2012","2","10"); // 10/03/2012
        var dateNaissance = ["1974","3","20"];

        // when
        var age = DateHelper.getAge(dateNaissance, today);

        // then
        expect(age).toBe(37);
      });
    });
  });
});
```

# Tests dans l'usine de dev

## ■ jasmine-maven-plugin

**maven**

- <http://searls.github.com/jasmine-maven-plugin/>
- Mode interactif : `mvn jasmine:bdd + browser`
- Mode headless avec `htmlunit`

## ■ Jenkins



**Jenkins**

- Générer un surefire report à l'aide du jasmine maven plugin

# Saga pour la couverture

## Total coverage report

File	Statements	Executed	Coverage	
<b>Total</b>	<b>960</b>	<b>414</b>	<b>43 %</b>	
target/jasmine/src/view/savedProposalView.js	89	6	6 %	
target/jasmine/src/view/proposal/singleProposalView.js	44	4	9 %	
target/jasmine/src/view/mainMenuView.js	35	4	11 %	
target/jasmine/src/view/modalClientNumberView.js	30	6	20 %	
target/jasmine/src/view/service/selectionnableServiceView.js	20	4	20 %	
target/jasmine/src/view/startBookingView.js	60	12	20 %	
target/jasmine/src/view/service/availableServicesView.js	33	7	21 %	
target/jasmine/src/model/errorModel.js	13	3	23 %	
target/jasmine/src/view/proposal/proposalMainView.js	21	5	23 %	
target/jasmine/src/view/aboutView.js	16	4	25 %	
target/jasmine/src/view/footerView.js	17	5	29 %	
target/jasmine/src/view/service/serviceMultidatesView.js	13	4	30 %	
target/jasmine/src/model/loginModel.js	29	10	34 %	
target/jasmine/src/view/availableAccomodationsView.js	94	33	35 %	
target/jasmine/src/util/templatingContext.js	13	5	38 %	
target/jasmine/src/router.js	34	14	41 %	
target/jasmine/src/view/loginView.js	44	19	43 %	
target/jasmine/src/view/homeView.js	9	4	44 %	
target/jasmine/src/view/proposal/proposalRequestView.js	9	4	44 %	
target/jasmine/src/model/lieuDeSejourDescriptionModel.js	4	2	50 %	
target/jasmine/src/model/service/selectionnableServiceModel.js	32	16	50 %	
target/jasmine/src/model/accomodation/selectedAccomodationsModel.js	7	4	57 %	
target/jasmine/src/model/dossier/confirmationDossierModel.js	7	4	57 %	
target/jasmine/src/model/dossier/creationOptionModel.js	7	4	57 %	
target/jasmine/src/model/proposal/validationProposalModel.js	7	4	57 %	

# Saga pour la couverture

target/jasmine/src/view/homeView.js

9

4

44%

Missed lines: [15-27](#)

```
1  | 1 define([ 'underscore', 'backbone', 'render!home', 'require', 'router' ],
2  |   function(
3  |     | 1 var singleton = undefined,
4  |         Backbone, template, require, router ) {
5  |         HomeView = Backbone.View.extend({
6  |             el : "#container",
7  |
8  |             initialize : function() {
9  |             },
10 |
11 |             events: {
12 |                 "click #toStartBooking": "toStartBooking"
13 |             },
14 |
15 |             render : function() {
16 |                 this.$el.html(template());
17 |             },
18 |
19 |             toStartBooking : function() {
20 |                 require([ './startBookingView' ],
21 |                 function(startBookingView) {
22 |                     startBookingView.showMe();
23 |                 });
24 |             }, {
25 |
26 |             showMe: function() {
27 |                 router.navigate("", {title: "Cobalt Démo"});
28 |                 singleton.render();
29 |             }
30 |         });
31 |     });
32 | }
```

# Couverture dans l'UDD

## ▪ saga-maven-plugin

- <http://timurstrekalov.github.com/saga/>
- S'interface facilement avec Jasmine (mais pas que)

**maven**

## ▪ Jenkins

- Publier le rapport html dans le job



**Jenkins**

## ▪ Sonar

- Importer le rapport .dat

**sonar**



[www.parisjug.org](http://www.parisjug.org)

***À vous de jouer !***

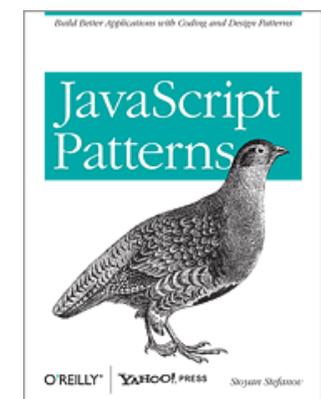
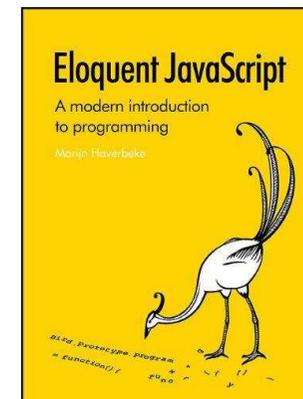
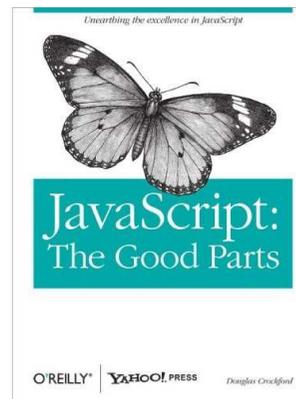


# Take-away

- **Structurer votre code JavaScript grâce à Backbone.js**
- **Intégrer des devs JavaScript dans un environnement Java**
- **S'intégrer avec la culture de développement**
  - Capitaliser sur l'UDD
  - Permettre aux développeurs de se former
- **S'intégrer avec le logiciel existant**
  - Minimiser l'adhérence entre le legacy et les nouveaux développements

# Plus loin avec JavaScript

- **Coder ensemble !**
  - Un pionnier
  - Binômage, dojo
- **Apprendre JavaScript**
  - JavaScript : The Good Parts
  - Eloquent JavaScript
  - JavaScript Patterns



# Plus loin sur le MVC client

## ▪ Tutoriaux

- TodoMVC : <http://addyosmani.github.com/todomvc/>
- Wine Cellar :  
<http://www.adobe.com/devnet/html5/articles/backbone-e-cellar-pt1.html>
- Notre exemple :  
<https://github.com/fredZen/java-backbone-demo>

## ▪ Les docs des librairies sont bien faites

- Backbone.js, jQuery, jQuery-UI, Underscore.js, Require.js, Lo-Dash, Twitter Bootstrap, ...

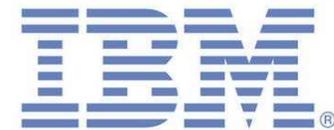


[www.parisjug.org](http://www.parisjug.org)

# *Questions / Réponses*



# Sponsors



Pensée novatrice  
& Vision technologique



# *Merci de votre attention!*



Sponsors

**Platinum** [www.parisjug.org](http://www.parisjug.org)

Xebia zenika OBJET DIRECT NOVEDIA Sfeir  
OXiane F&S Connect valtech IBM  
Sponsors **Gold**  
in fine Objectif So@t ippen

# Licence



Paternité-Pas d'Utilisation Commerciale-Partage des Conditions Initiales à l'Identique  
2.0 France

- <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>