

Play! with Sencha

@vijaykiran ವಿಜಯ್ ಕಿರಣ್



LUNATECH

lunatech-research.com

Overview

- Background
- REST First approach
- Front end technology choices
- Sencha
- Play! REST Backend Implementation
- What's Next ..

Play! with Sencha



Background

- Typical Play! Web Application
 - Server-Driven templates.
 - Ajax using JQuery.

Background

- First Step towards Mobile Web
 - JQuery Mobile
 - Ajax using JQuery.

Play! with Sencha



REST First

- REST API
- Service Oriented
- Decoupling of Data and Presentation.
- Uniform interface

Play! with Sencha



REST First

GET `/ {controller}` `{controller}.index`

GET `/ {controller} / {id}` `{controller}.show`

POST `/ {controller}` `{controller}.create`

PUT `/ {controller} / {id}` `{controller}.update`

DELETE `/ {controller} / {id}` `{controller}.delete`

Play! with Sencha



Frontend

- JQuery UI/JQuery Mobile
- SproutCore/Cappuccino
- Dojo
- Sencha Touch/ExtJS
- Native Apps

Play! with Sencha



Frontend - Mobile

- HTML/JS
- Sencha Touch
- JQuery Mobile

Play! with Sencha



Why Sencha Touch

- Class System
- Support for MVC Applications
- Dependency Management
 - Async Loading, Build support
- Bundle as “Native App”

Play! with Sencha



Sencha (Touch) App

- Model (Domain Objects)
 - REST, JSON-P, Direct, LocalStorage
- Store
 - Encapsulates Model Objects
 - Sorting, Filtering etc.

Play! with Sencha



Sencha Touch App

```
Ext.define('TalkRate.model.Talk', {
  extend: 'Ext.data.Model',
  fields: [
    {name: "id", type: "int"},
    {name: "title", type: "string"},
    {name: "speaker", type: "string"}
  ],
  proxy: {
    type: 'rest',
    url: '/talks',
    reader: {
      type: 'json',
      root: 'items'
    }
  }
});
```

Play! with Sencha



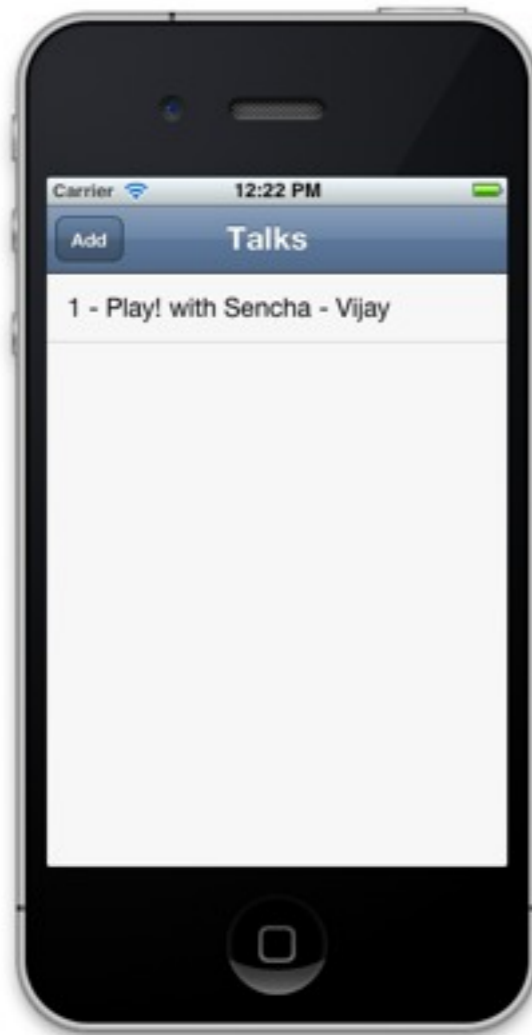
Sencha Touch App

- View/Components
 - List, Navigation View, Toolbar etc.
 - Can be themed

Play! with Sencha



Sencha Touch App



```
Ext.define('TalkRate.view.Talks', {
  extend: 'Ext.NavigationView',
  requires: ["TalkRate.view.TalksList"],
  xtype: 'talks',
  config: {
    fullscreen: false,
    navigationBar: {
      items: [
        {
          xtype: 'button',
          id: 'addTalk',
          text: 'Add'
        }
      ]
    },
    items: [
      {
        xtype: 'talkslist',
        title: 'Talks',
        store: 'TalkStore'
      }
    ]
  }
});
```

Play! with Sencha



Sencha Touch App

- Controllers
 - Glue between model & views
 - "control"s component actions.

Play! with Sencha



Sencha Touch App

```
Ext.define('TalkRate.controller.Home', {
  views:['Talks'],
  stores:['TalkStore'],
  init:function () {
    this.getTalksView().create();
    this.control({
      'button#addTalk':{ 'tap':this.showTalkForm }
    });
  },
  showTalkForm:function () {
    ....
  }
});
```

Play! with Sencha



REST back end with Play!

- Routes Make REST Easy
- Request Parsing
 - request.body parsing with Binder

Play! with Sencha



routes

GET `/{controller}` `{controller}.index`

GET `/{controller}/{id}` `{controller}.show`

POST `/{controller}` `{controller}.create`

PUT `/{controller}/{id}` `{controller}.update`

DELETE `/{controller}/{id}` `{controller}.delete`

Play! with Sencha



REST back end with Play!

```
//GET {controller}  
public static void index() {  
    renderJSON(new SenchaResponseWrapper(true, Talk.findAll()));  
}
```

```
//POST {controller}  
public static void create(JsonObject body) throws IOException  
{  
    Talk talk = new Gson().fromJson(body, Talk.class);  
    final Talk result = JPA.em().merge(talk);  
    renderJSON(new SenchaResponseWrapper(true, result));  
}
```

Play! with Sencha



REST back end with Play!

@Global

```
public class JsonBinder implements TypeBinder<JsonObject> {  
    @Override  
    public Object bind(String name,  
                      Annotation[] annotations,  
                      String value,  
                      Class actualClass,  
                      Type genericType)  
        throws Exception {  
        return new JsonParser().parse(value);  
    }  
}
```

Play! with Sencha



What's Next?

- Auto-generate Model JS
- Generic REST Controllers
- Auto-binding to JPA Models
- OAuth Provider for Play!

Play! with Sencha



Demo/Code

<https://github.com/lunatech-labs/play-sencha-touch>

@vijaykiran

Play! with Sencha



End



Play! with Sencha

