ExtJS Responsive Layout

Come gestire desktop e mobile

LUCA MINUTI

SENCHA MVP

email

LUCA.MINUTI@GMAIL.COM

GITHUB

HTTPS://GITHUB.COM/LMINUTI/

DelphiDay italian conference

ONLINE EDITION

wintech italia

GITHUB PROJECTS

GRAPHQL FOR DELPHI
https://github.com/lminuti/graphql

DELPHI-OPENSSL
https://github.com/lminuti/delphi-openssl

WIRL
https://github.com/delphi-blocks/WiRL

DELPHI-WKHTMLTOX
https://github.com/lminuti/delphi-wkhtmltox

DelphiDay italian conference

ONLINE EDITION

wintech italia

AGENDA

- ➔ Cos’è responsive design
- ➔ Principali layout di ExtJS
- ➔ Responsive config
- ➔ Gestione dei profili
- ➔ Media query
- ➔ Integrazione con altre librerie

RESPONSIVE DESIGN

➔ Lo sviluppo del sito/app si adatta al dispositivo:

- Dimensione
- Orientamento
- Piattaforma

DelphiDay italian conference

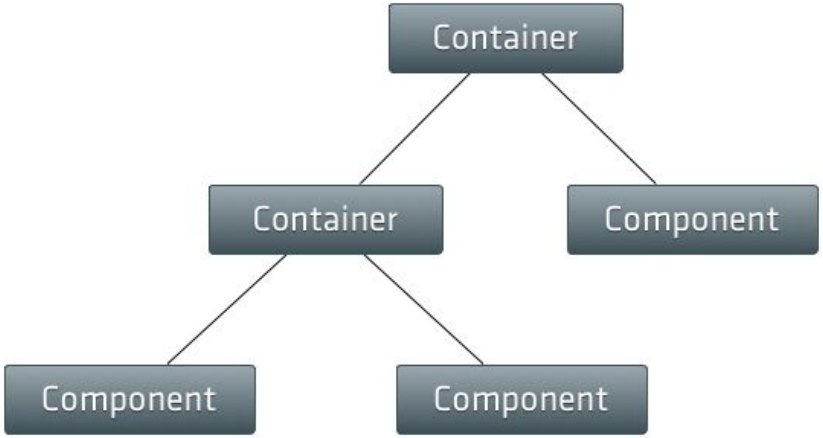
TOOLKIT

- Classic
  - Sviluppo su desktop e browser legacy
  - Al momento è la versione più completa
- Modern
  - Nato dal merge di ExtJS con Sencha Touch
  - Mobile (ma funziona con i browser moderni)
  - Framework ridotto ma migliori prestazioni
- Universal
  - Supporta entrambe le modalità (doppie view)

EXTJS LAYOUT

CONTAINER

- ➔ Sono i componenti che possono contenerne altri (panel, tabpanel, toolbar, ...)
- ➔ Hanno una proprietà *items*
- ➔ e una proprietà *layout*



LAYOUT

- Il layout di un container determina come gli elementi contenuti devono disporsi all'interno del container stesso, in particolare la loro dimensione e posizione
- Ogni container ha diversi parametri di configurazione che ne possono influenzare il comportamento
- Nel toolkit "classic" i layout sono implementati in JavaScript mentre con "modern" vengono usati i CSS

LAYOUT

- Tre modi per indicare la dimensione dei componenti:
  - Imposta: tramite le proprietà `width` e `height`
  - In base al parent: flex o percentuale
  - Automatica: calcolata da ExtJS in base al contenuto
- Anche se può sembrare più semplice in genere è meglio evitare di importare manualmente height e width

DIMENSIONI

- `Width` e `Height` permettono di impostare le dimensioni
- Possono essere numeri o valori supportati dai CSS (300, ‘300px’, ‘300%’, ‘auto’, …)
- auto o vuoto significa “natural size”: la dimensione si adegua al contenuto
- Ci sono anche `minWidth`, `maxWidth`, `minHeight` e `maxHeight` che sono analoghe

LAYOUT: AUTO

- In assenza di layout viene usato il layout “auto”
- Vengono applicate le regole base del DOM
- Non viene imposto nessun ridimensionamento o posizionamento particolare
- In generale è sempre meglio specificare espressamente un layout

LAYOUT: FIT

➔ Un unico componente che occupa tutto lo spazio disponibile

LAYOUT: BOX

- Esistono due tipo: hbox e vbox
- I componenti si dispongono uno affianco all'altro orizzontalmente o verticalmente
- Layout config: `align`, `pack`, `reverse`, `wrap`
- I singoli componente possono usare la proprietà flex

LAYOUT: FORM

- Accetta solo form field come figli
- Simile al vbox ma allinea automaticamente le label
- Le proprietà sono:
  - `labelWidth`: la dimensione di tutte la label (‘auto’ le calcola in base alla più grande)
  - `itemSpacing`: spazio tra un field e l’altro

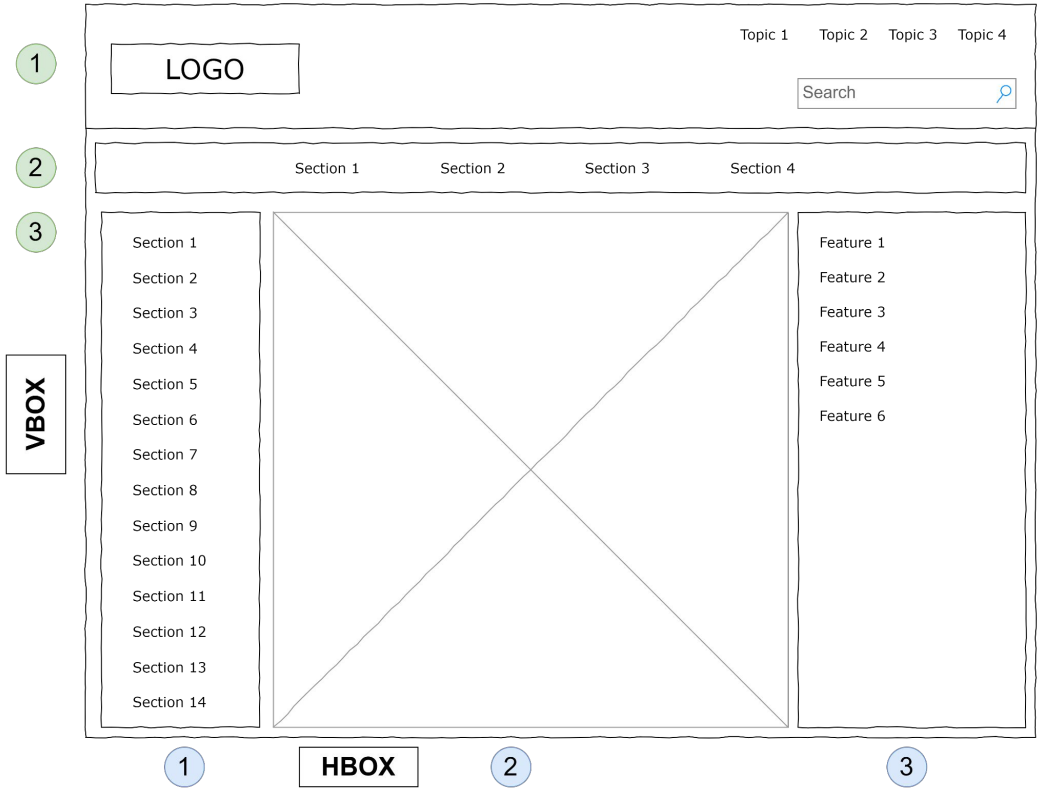
LAYOUT: CARD

- Impila tutti i componenti uno sopra l'altro
- È visibile un solo componente alla volta
- `setActiveItem` può essere usato per definire quale elemento visualizzare

LAYOUT: DOCKING

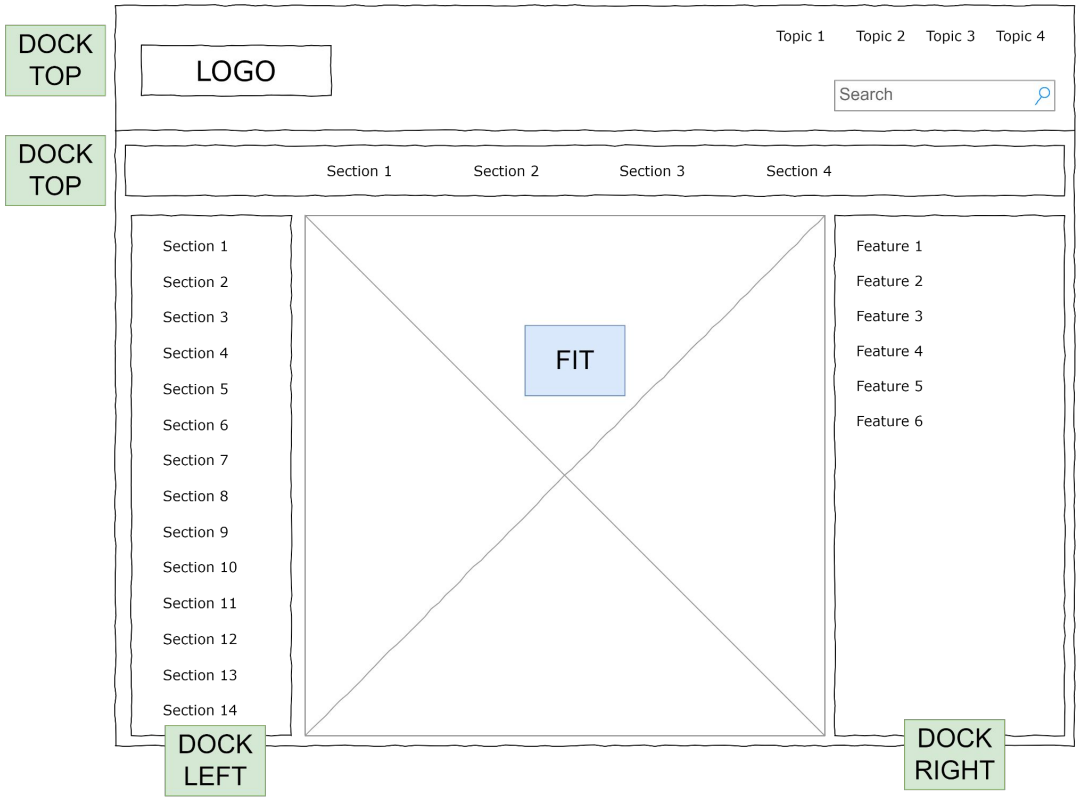
- Qualunque componente può avere la proprietà `docked` (`top`, `right`, `bottom`, `left`)
- Vengono agganciati al bordo indicato
- Non rientrano nella gestione del layout
- Permettono di semplificare molto lo sviluppo di un'interfaccia

LAYOUT: DOCKING



DelphiDay italian conference

LAYOUT: DOCKING



DelphiDay italian conference

demo time

RESPONSIVE CONFIG

RESPONSIVE CONFIG

- Permettono una modifica dinamica della proprietà (read/write)
- Si basano su:
  - `landscape, portrait, tall, wide, width, height, platform`
- Per attivarle require di `Ext.Responsive`
- Oppure mixin dove non è supportato:
  - `Ext.mixin.Responsive`

Responsive config

```
responsiveConfig: {
    landscape: {
        region: 'west'
    },
    portrait: {
        region: 'north'
    }
}
```

```
responsiveConfig: {
    'desktop || width > 800': {
        region: 'west'
    },

    '!(desktop || width > 800)': {
        region: 'north'
    }
}
```

demo time

PLATFORM CONFIG

PLATFORM CONFIG

- ➔ Simile a `responsiveConfig`
- ➔ Si può usare su tutte le view
- ➔ Viene valutata all'apertura dell'applicazione
- ➔ Associa una configurazione al valore della proprietà `Ext.platformTags`:
- ➔ android, chrome, classic, desktop, edge, firefox, ie8, ios, ipad, iphone, modern, phone...

Platform config

```
Ext.define('App.view.main.Main', {

    extend: 'Ext.tab.Panel',

    title: 'Main',

    platformConfig: {
        desktop: {
            tabBarPosition: 'left'
        },
        phone: {
            tabBarPosition: 'top'
        }

    }
});
```

demo time

PROFILE

DelphiDay
italian conference

PROFILE

- Più versioni della stessa view
  - Per dispositivo (phone, tablet, PC tramite): `Ext.os`
  - Per risoluzione: `Ext.Viewport.getSize()`
  - Orientamento: `Ext.Viewport.getOrientation()`
  - Support: `Ext.supports` (svg, canvas, Geolocation,...)
  - L'ora del giorno !?!?
- Configurazione:
  - Dichiarazione dei profili su app.js
  - Definizione dei singoli profili

Profili

```
// Application.js

Ext.define('App.Application', {
    extend: 'Ext.app.Application',

    profiles: [
        'Desktop',
        'Mobile'
    ]
});
```

Profili

```
Ext.define('App.profile.Desktop', {
    extend: 'Ext.app.Profile',

    mainView: 'App.view.desktop.Main',
    views: {
        'users': 'App.view.desktop.user.UserView',
        'chat': 'App.view.desktop.user.ChatView'
    },
    isActive() {
        return Ext.os.is.Desktop;
    },
    launch() {
        console.log('Launch Desktop');
        Ext.getBody().addCls('desktop-profile');

    }
});
```

Profili

```
Ext.define('App.profile.Phone', {
    extend: 'Ext.app.Profile',

    mainView: 'App.view.phone.Main',
    views: {
        'users': 'App.view.phone.user.UserView',
        'chat': 'App.view.phone.user.ChatView'
    },
    isActive() {
        return !Ext.os.is.Desktop;
    },
    launch() {
        console.log('Launch Phone');
        Ext.getBody().addCls('phone-profile');

    }
});
```

CSS / MEDIA QUERY

MEDIA QUERY

- Permettono di applicare delle regole CSS in base a una serie di condizioni
  - sul dispositivo
  - il browser
  - o le impostazioni di sistema

```
@media (max-width: 12450px) { ... }
```

MEDIA QUERY

- MediaType: all, print, screen, speech
- MediaFeature: width, height, orientation, display-mode, scripting
- Operator: and, or, not

BOOTSTRAP

BOOTSTRAP GRID

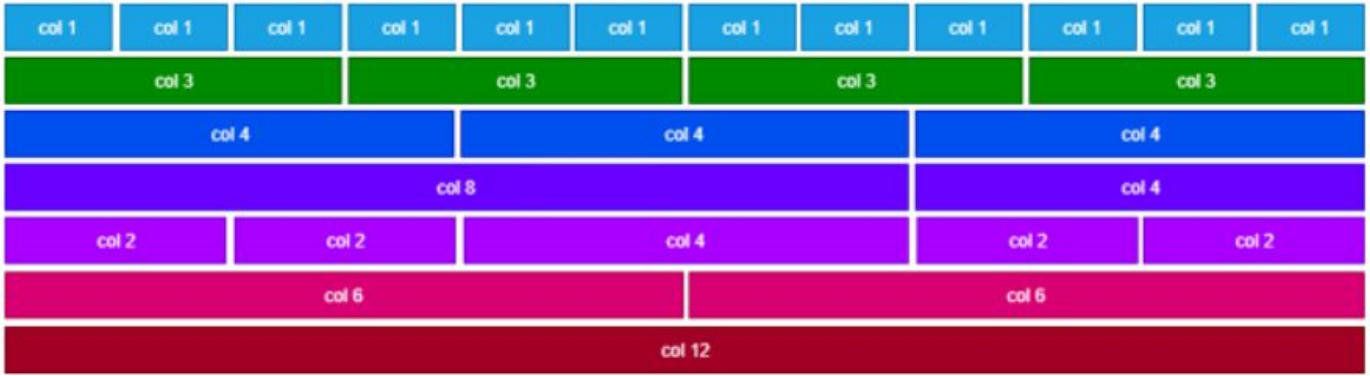
- ➔ Si basano su CSS flex layout
- ➔ Ci sono 5 responsive breakpoint

Breakpoint	Class infix	Dimensions
Extra small	*None*	<576px
Small	`sm`	≥576px
Medium	`md`	≥768px
Large	`lg`	≥992px
Extra large	`xl`	≥1200px
Extra extra large	`xxl`	≥1400px

DelphiDay italian conference

BOOTSTRAP GRID

- Gli elementi di una riga si spartiscono equamente lo spazio oppure è possibile indicare lo spazio da occupare in una griglia virtuale di 12 celle



Bootstrap

```
{
        innerCls: 'row',

        defaults: {
            cls: 'col-lg-4 col-sm-12'
        },

        items: [{
            xtype: 'textfield',
            label: 'Campo 1'
        },{
            xtype: 'textfield',
            label: 'Campo 2',
            cls: 'col-lg-8 col-sm-12'
        },{
            xtype: 'textfield',
            label: 'Campo 3'
        }]
}
```

CLASSI CSS

col-lg-4: La colonna occupa 4 slot su schermi grandi (large)
col-lg-8: La colonna occupa 8 slot su schermi grandi (large)
col-sm-12: La colonna occupa 12 slot su schermi piccoli (small)

demo time

THANK YOU

DelphiDay
italian conference