

# Zen Mojoのご紹介

インターフェースシステムズジャパン株式会社



# アジェンダ

Zenのおさらい

Zen Mojoの2つの用途

- デスクトップ

- モバイル

アーキテクチャー

- シングルページアーキテクチャ

- プラグイン

コード例



# Zenのおさらい

約10年前にリリース

製品として完全にサポート保証

マルチページアーキテクチャー

サーバー側コンポーネントモデル

- 全ての必要なHTMLとJavaScriptコードは、サーバーからクライアントに送られる
- オンラインの接続と十分な帯域が必要

インターネットアプリケーションにはそれで良かった

昨今ユーザーの期待は

- 低い帯域でも良い応答
- オフラインモード

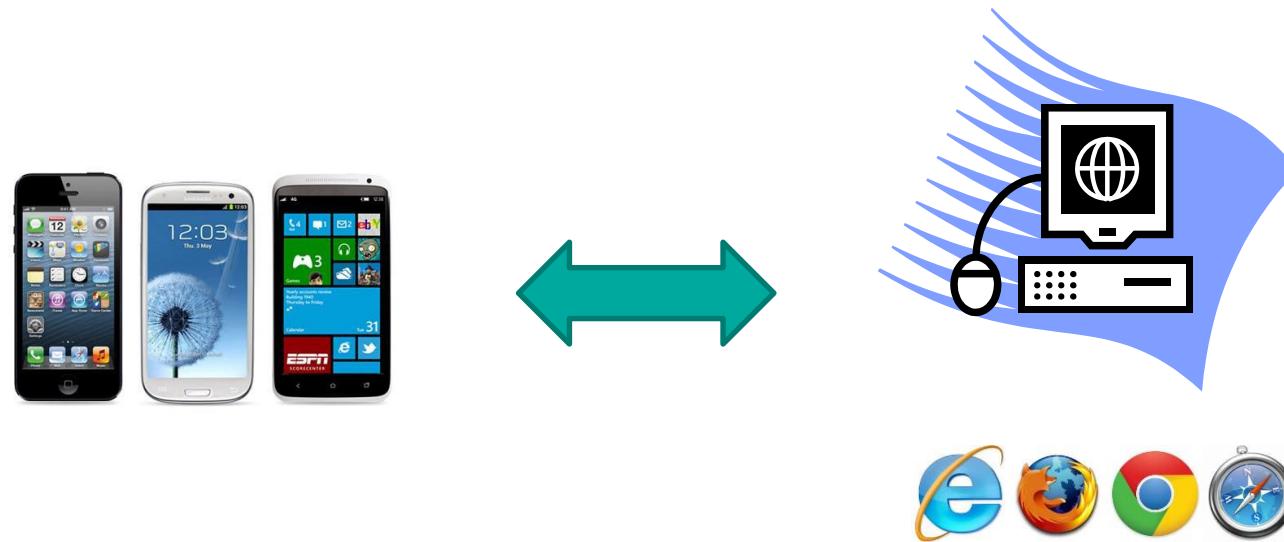


# Zen Mojo

## Zen Mojo

異なるプラットフォーム用の1つのフレームワークであり、1つの開発パラダイム

- デスクトップアプリケーション
- モバイルアプリケーション



# デスクトップ用Zen Mojo

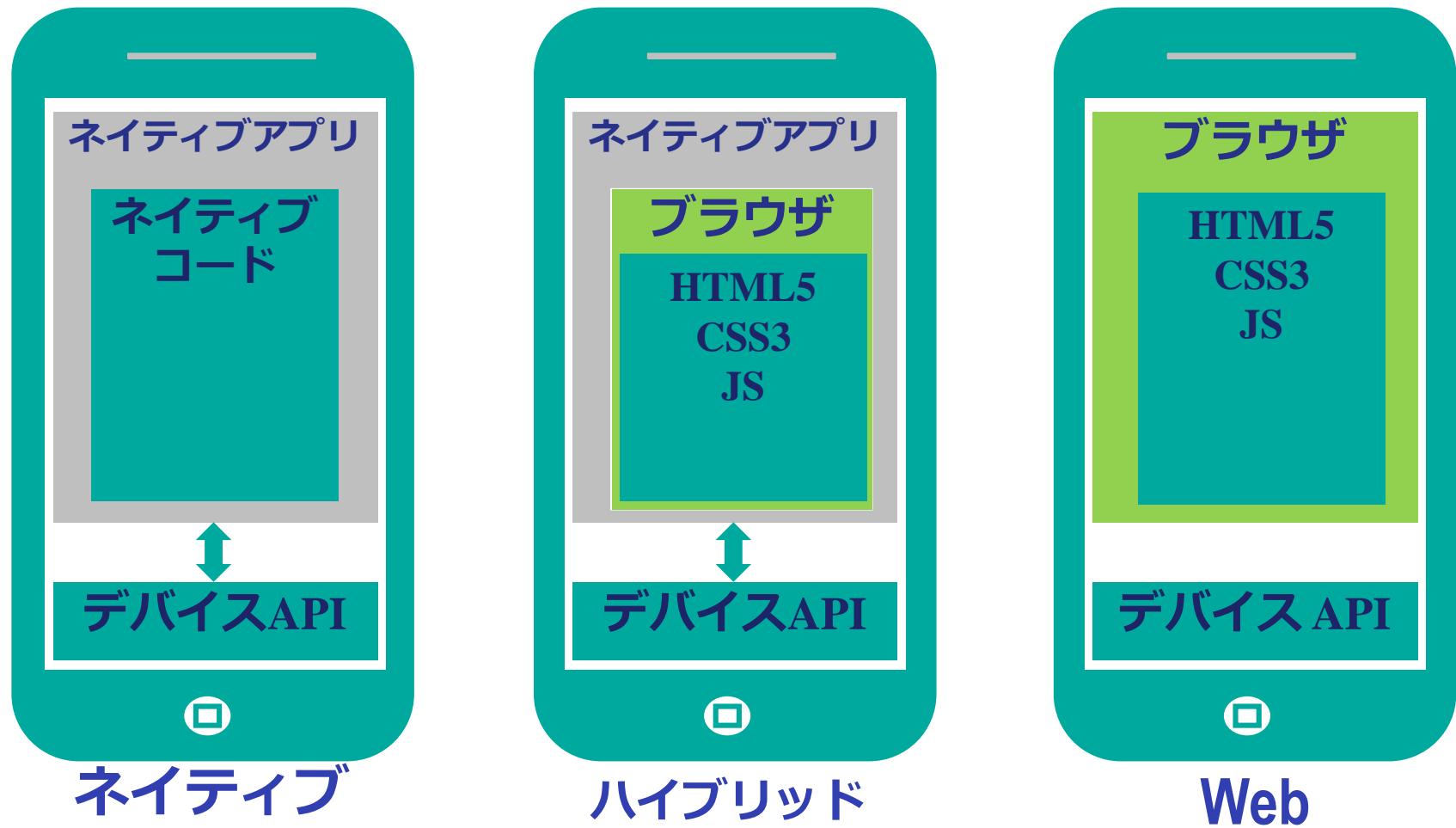
Dojo Demo

Demo By Type

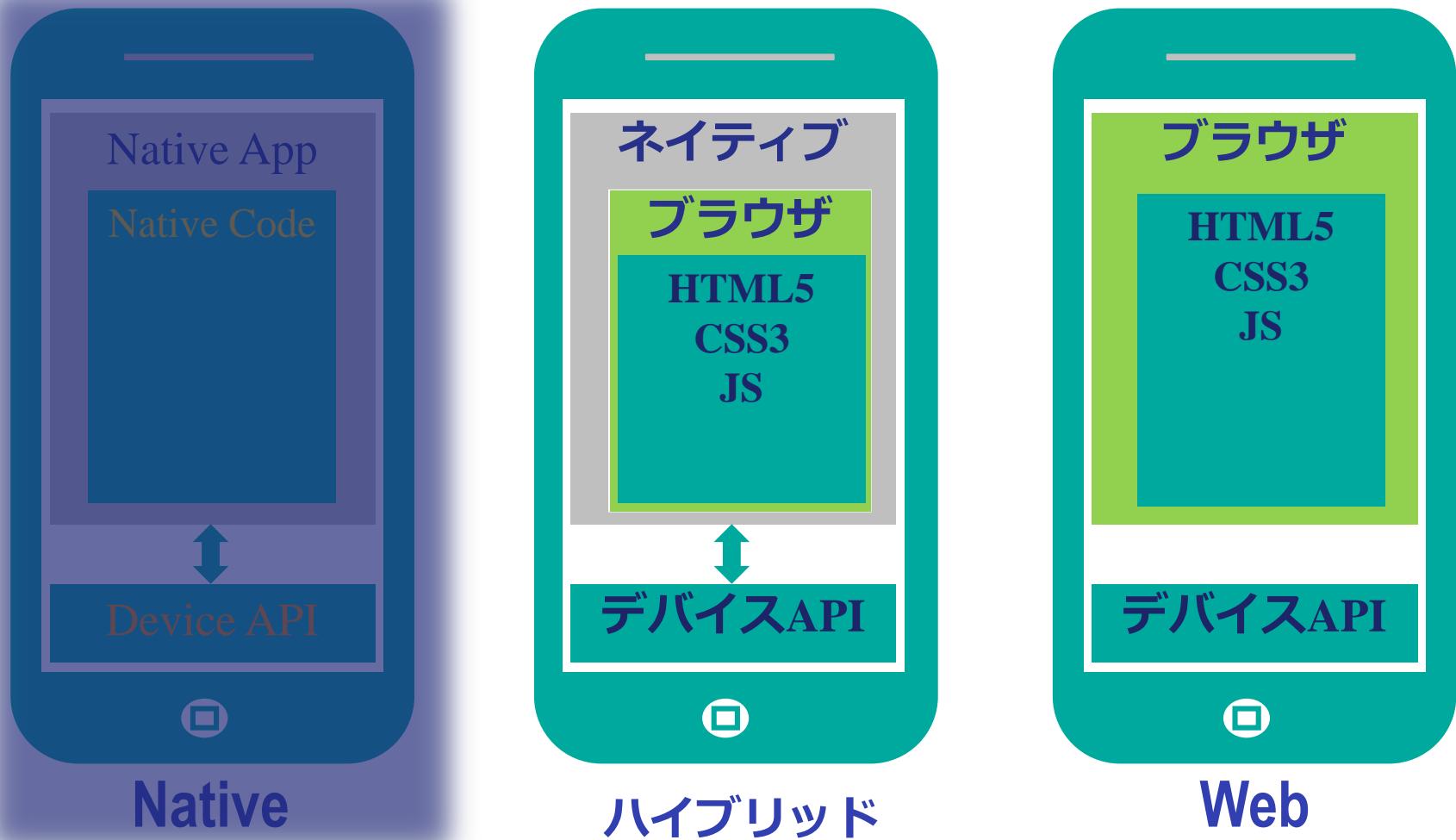
- +
- Layout Elements
- +
- Grid Elements
- +
- Charts
  - Areas
  - Clustered Bars
  - Bars
  - Bubbles
  - Bubbles (+Grid Plot)
  - Candlesticks
  - Column
  - Column with Animate
  - Column with Highlight
  - Column with Shadow
  - Column with Shake
  - Column with Mouse Zoom
  - Column with Touch Zoom (Broken?)
  - Lines
  - Lines with Magnify
  - Pie Chart
  - Pie Chart with Title
  - Pie Chart with Tooltip
  - Pie Chart with Tooltip (custom)



# モバイルアプリケーションを開発する方法



# Zen Mojo: Webとハイブリッド



# モバイル用Zen Mojo



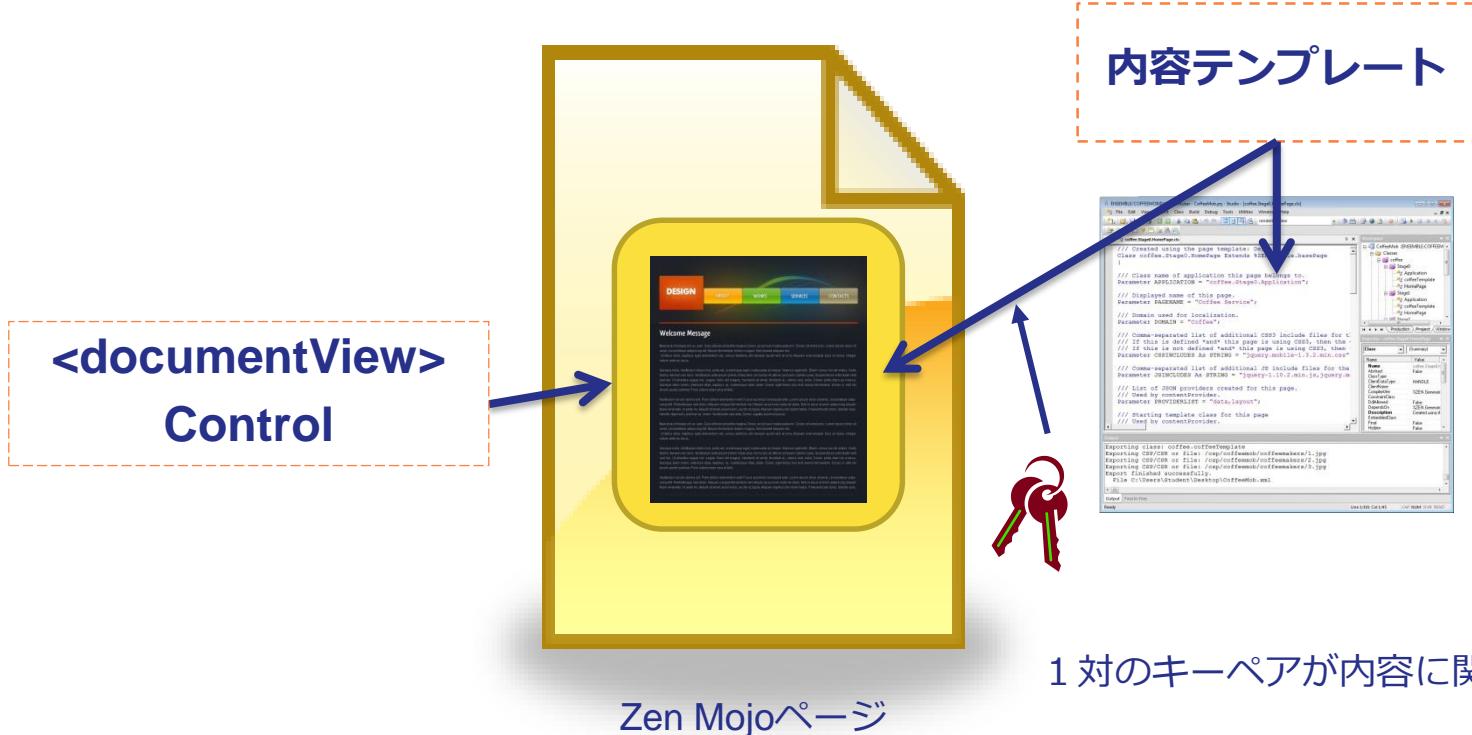
# Zen Mojoアプリを開発中

Zen Mojoでの開発は、Zenとは根本的に違っている

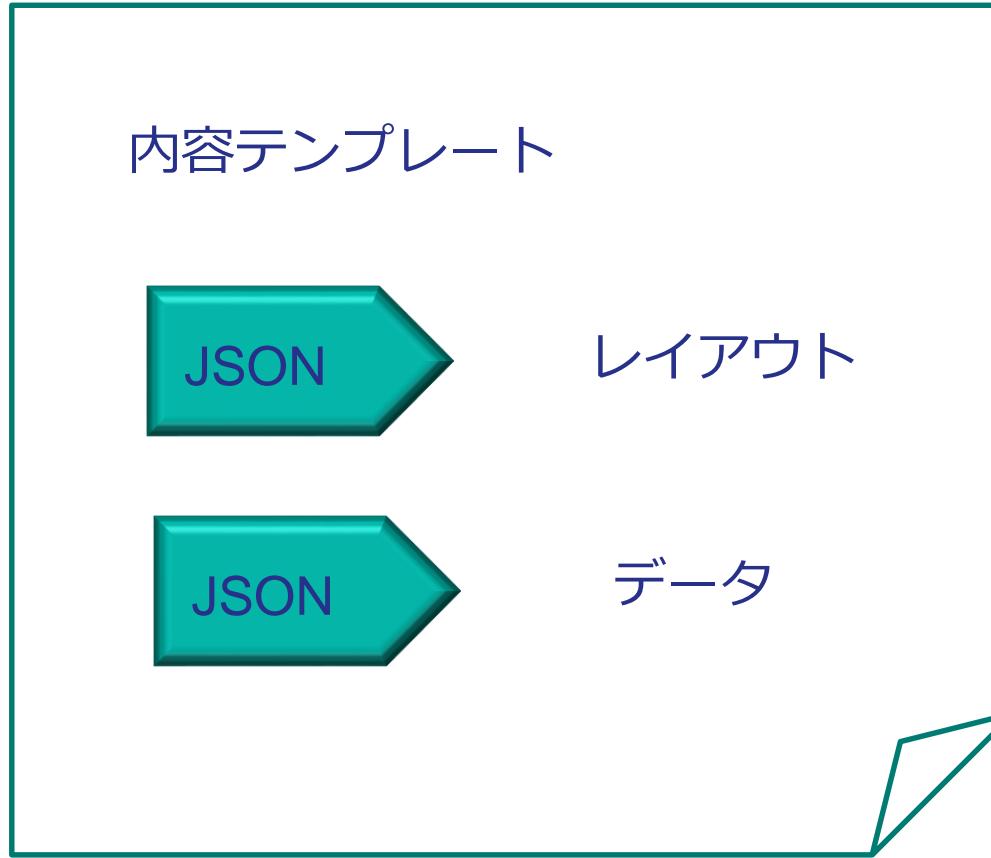
- シングルページアーキテクチャ
- クライアント側での描画
- より進化したキャッシング機構
- 効率の良いクライアント／サーバー通信(JSON)
- 完全データ駆動 (レイアウトとデータの内容)



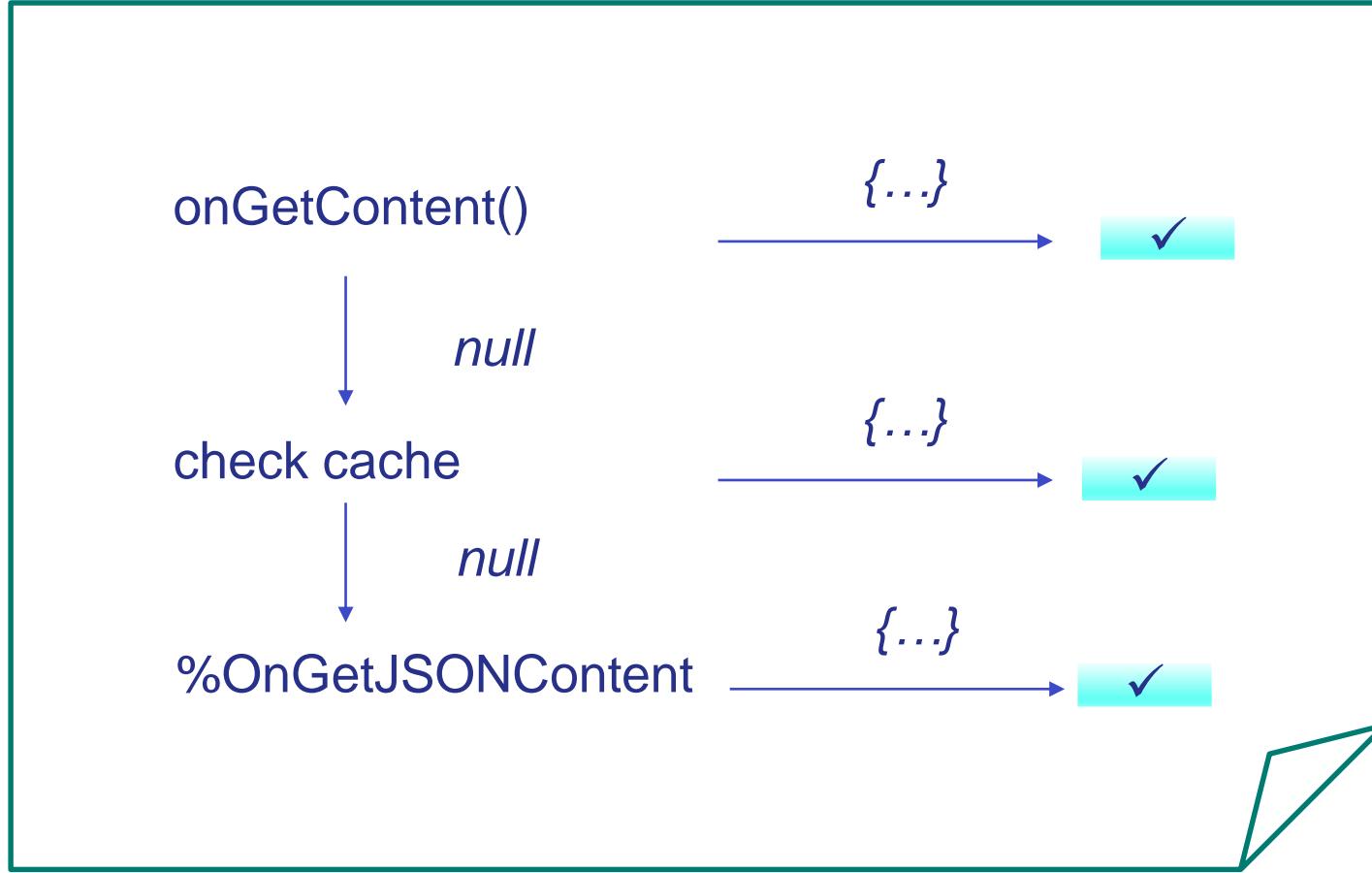
# Zen Mojoアプリを開発中



# Zen Mojoアプリを開発中



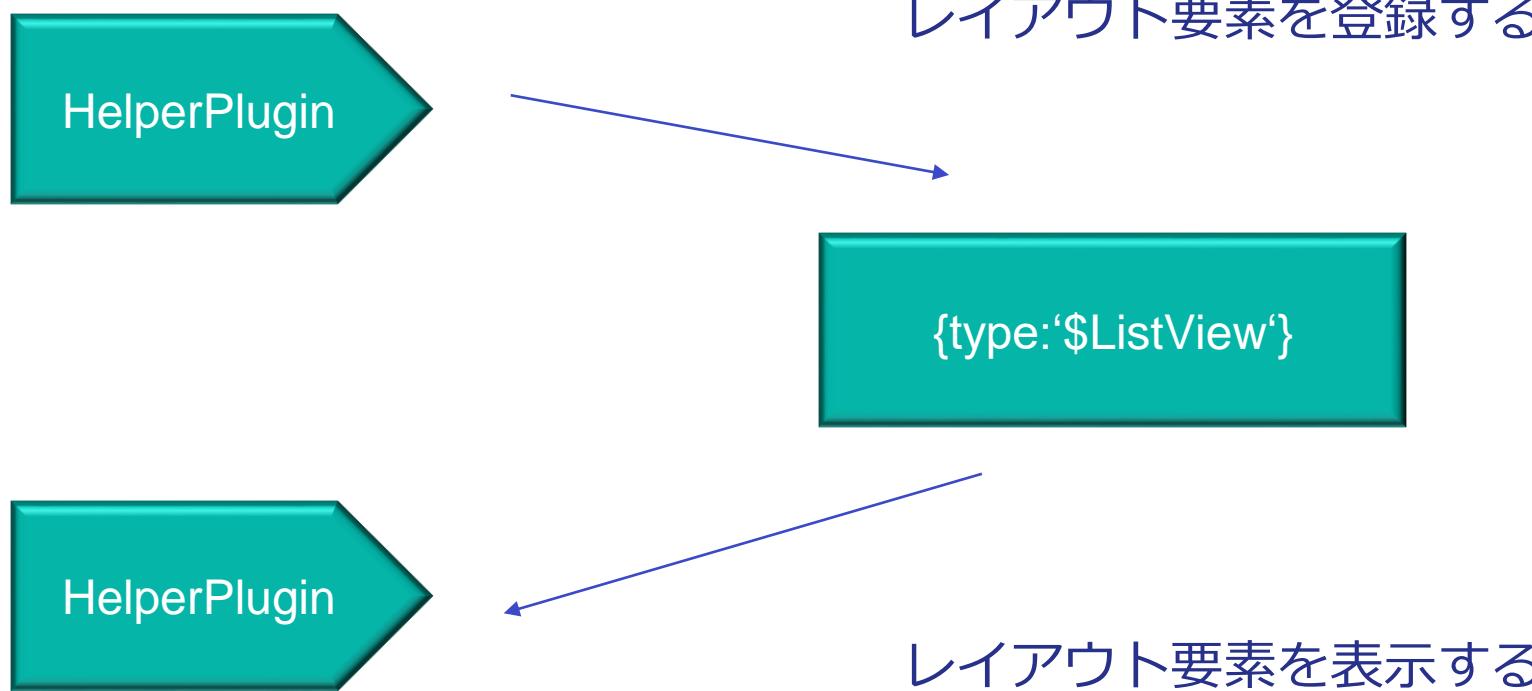
# Zen Mojoアプリを開発中



# Zen Mojoアプリを開発中



# Zen Mojoアプリを開発中



# Zen Mojoアプリを開発中

```
<documentView>  
  ページマネージャー  
  ヘルパー・プラグイン  
</documentView>
```

全てのイベントは、documentView  
コンポーネントによって起こされ、  
その内容テンプレートに委譲される

## イベントハンドラー

- onchange
- onselect
- onevent



# コードの例



# Zen Mojoアプリを開発中

HomePageの定義

```
/// Created using the page template: Default
Class coffeeDesktop.HomePage Extends %ZEN.Mojo.basePage

/// List of JSON providers created for this page.
/// Used by contentProvider.
Parameter PROVIDERLIST = "data,layout";

/// Starting template class for this page
/// Used by contentProvider.
/// This is overriden by application settings, of provided.
Parameter TEMPLATECLASS = "coffeeDesktop.coffeeDesktopTemplate";

/// Comma-separated list of additional JS include files for the page.
Parameter JSINCLUDES As STRING = "dojo-release-1-9-1/dojo/dojo.js";

/// Comma-separated list of additional CSS include files for the page.
/// If this page is using CSS3 then the CSS3INCLUDES parameter is used (if defined).
Parameter CSSINCLUDES As STRING = "dojo-release-1-9-1/dijit/themes/claro/claro.css,
                                    dojo-release-1-9-1/gridx/resources/claro/Gridx.css";
```



# Zen Mojoアプリを開発中

単純なXData 内容ブロック

```
✉<mojo:documentView id="mainView"
    initialDocumentKey="customer"
    initialLayoutKey="customer-list"
    ongetlayout = "return zenPage.getContent('layout',key,criteria);"
    ongetdata = "return zenPage.getContent('data',key,criteria);"
    <mojo:dojo-1.9.1-PageManager onPageShow="zenPage.onPageShow(key);">
        <mojo:dojo-1.9.1-DijitHelper/>
        <mojo:dojoGridX-1.3.0-Helper/>
        <mojo:HTML5Helper/>
        <mojo:mojoDefaultHelper/>
    </mojo:dojo-1.9.1-PageManager>
</mojo:documentView>

</mojo:documentView>
```

templateクラスのonGetContent()に  
ディスパッチする



# Zen Mojoアプリを開発中

template内にコールバックを実装中

```
/// Client-side content provider method.  
/// <var>which</var> is the name of the content to return.  
/// <var>key</var> is the key value associated with the request.  
/// <var>criteria</var> is the search criteria associated with the request.  
ClientMethod  
    which, switch (which) {  
        key,         case 'layout':  
        criteria:   return this.getLayout(key, criteria);  
    }  
    switch (which) {  
        case 'data':  
            return null;  
        case '':  
            return null;  
    }  
    return null;  
}
```

サーバーの%OnGetJSONContent()  
にディスパッチする



# Zen Mojoアプリを開発中

クライアントにレイアウトを生成中

```
content = {  
    children: [  
        {type: '$LayoutContainer',  
         key: 'layoutContainer-1',  
         design: 'headline',  
         children: [  
             {type: '$ContentPane', region: 'top', title: 'Headline', key:  
              'headline-1',  
              content: 'Customer List', cssClass: 'headline-content'},  
             {type: '$div', $content: 'Customer List', cssClass: 'customer-list'},  
             {type: '$MenuBar', children: [  
                 {type: '$MenuBarItem', label: 'Customer List',  
                  key: 'menu-show',  
                  value: 'customer-list'}],  
                  children: [  
                      {type: '$ContentPane', region: 'bottom', title: 'Customer List',  
                       key: 'customer-list-1',  
                       content: 'Customer List', cssClass: 'customer-list-content'}]  
             }]  
        }]  
    }]
```

子オブジェクトの配列

Typeプロパティにクライアントコンポーネントのタイプを記述する



# Zen Mojoアプリを開発中

サーバー側に内容を生成中

```
set pObject = ##class(%ZEN.proxyObject).%New()  
i "customerListColumns": [ {  
    "field": "id",  
    "name": "Identity"  
}, {  
    "field": "name",  
    "name": "Customer"  
}  
]  
]
```

```
set column = ##class(%ZEN.proxyObject).%New()  
set column.field = "name"  
set column.name = "Customer"  
do pObject.customerListColumns.Insert(column)
```

}



# Zen Mojoのご紹介

インターフェーズシステムズジャパン株式会社

