Mtime
前端开发规范文档

版本记录

修改日期	修改人	修改版本	修改内容摘要
2009-03-03	Yuanjian	1.0	初稿

目　录

1 总体说明

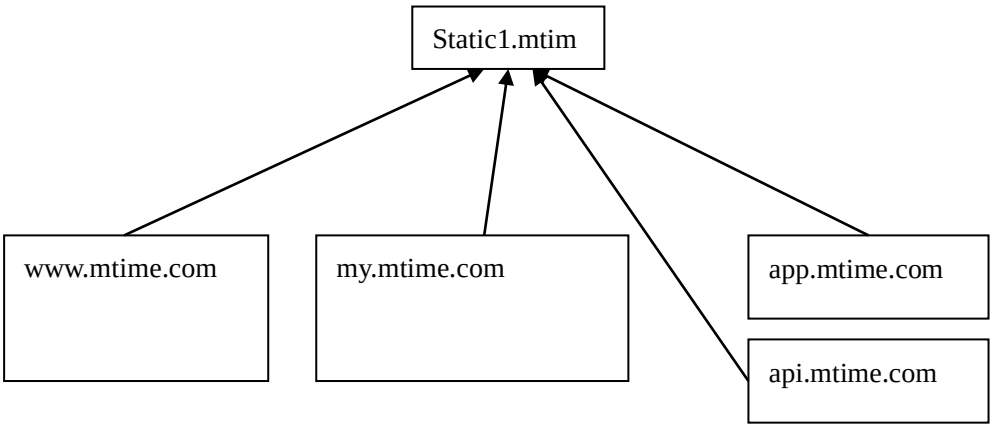
主要的 js 和 css 都在前台目录下

1.1 代码路径

前台：$/codev3.0/MtimeMovieCommunity.root/MtimeMovieCommunity

1.2 部署

前台发布后会将所有的 js、cs 以及用到的图片打包压缩后放到 static1.mtime.cn 站点下以便于各个子站点使用，同时在 static1.mtime.cn 下各个子站点也有自己的目录。



例如，在这个页面：http://my.mtime.com/app/card/会加载如下的 js：
http://i.mtime.cn/20090227111758/js/ManagerSystem08.js
http://i.mtime.cn/my/20090226121751/js/appManager.js
http://i.mtime.cn/app/card/20090303112641/my/mypocket.js
其中：
http://i.mtime.cn/20090227111758 下是前台的资源目录
http://i.mtime.cn/my/20090226121751 下是用户中心的资源目录
http://i.mtime.cn/app/card/20090303112641 下是卡片游戏的资源目录

1.3 基础

1.3.1 公共脚本

- 公共脚本在页面的底部加载

systemall.js：统一加载，它由以下文件组成

```
<File Name="/js/systemall.js">
        <Include Name="/js/prototype.js"/>
        <Include Name="/js/global.3rdExt.js"/>
        <Include Name="/js/global.PrototypeExt.js"/>
        <Include Name="/js/global.NativeExt.js"/>
        <Include Name="/js/global.Mtime.js"/>
        <Include Name="/js/global.Utility.js"/>
        <Include Name="/js/mtime/validator.js"/>
        <Include Name="/js/08/Window.js"/>
        <Include Name="/js/08/Controls/Textbox.js"/>
        <Include Name="/js/08/MenuBase.js"/>
        <Include Name="/js/08/tabPanel.js"/>
        <Include Name="/js/08/accordionPanel.js"/>
        <Include Name="/js/08/navigationBar.js"/>
        <Include Name="/js/08/dropDownList.js"/>
        <Include Name="/js/08/autoCompleteBase.js"/>
        <Include Name="/js/08/search.js"/>
        <Include Name="/js/08/BaseWindow08.js"/>
        <Include Name="/js/08/Dialog.js"/>
        <Include Name="/js/08/box.js"/>
       <Include Name="/js/08/TooltipWindow.js"/>
       <Include Name="/js/track.js"/>
</File>
```

systemallobsolete.js：一些老页面在使用，会慢慢的去掉，它由以下文件组成

```
<File Name="/js/systemallobsolete.js">
    <Include Name="/js/prototype.js"/>
    <Include Name="/js/global.3rdExt.js"/>
    <Include Name="/js/global.PrototypeExt.js"/>
    <Include Name="/js/global.NativeExt.js"/>
    <Include Name="/js/global.Mtime.js"/>
    <Include Name="/js/global.Utility.js"/>
    <Include Name="/js/mtime/validator.js"/>
    <Include Name="/js/08/Window.js"/>
    <Include Name="/js/mtime/box.js"/>
    <Include Name="/js/mtime/tooltip.js"/>
    <Include Name="/js/mtime/popupwindow.js"/>
```

</File>

注：如果要增加打包文件，需修改文件：

$/codev3.0/Tools/MtimeClientCompress.root/MtimeClientCompress/MtimeClientCompress/Resources/Configs.xml

1.3.2公共样式表

- 样式表在页面的顶部加载

public_08.css：统一加载，它由以下文件组成

一个页面不应超过 2 个 css 文件

1.3.3动态加载脚本

- 引用整站的 js

```
$loadJs( jsPaths, callback );
```

- 引用子站的 js：在子站点里使用，加载子站点自有的 js

```
$loadSubJs( jsPaths, callback );
```

例如：

```
$loadJs( "/js/08/userAvatar.js" );
$loadJs( "/js/08/userAvatar.js", function() {
    alert( “userAvatar 已经加载” );
} );
$loadJs( ["/js/08/userAvatar1.js","/js/08/userAvatar2.js"], function() {
    lert( “userAvatar1，2 已经加载” );
} );
```

在用户中心的站点下：

```
$loadSubJs( "/js/movieManager.js );
```

注：loadJs 和 loadSubJs 的区别仅是对应的 jsserver 的地址不一样

1.3.4公共脚本中的 API

- 基于 prototype.js 1.5.1
- 页面跳转

```
$redirect( url )
```

- 延迟执行某方法

```
$defer( functionObj/*方法*/, obj/*绑定对象*/, args/*参数*/ )
FunctionExt.defer( functionObj/*方法*/, timeout/*延迟时间*/, obj/*绑定对象*/, args/*参数*/ )
```

例如：

```
var someClass = {
    doWork: function( parm ) {
        alert( parm + “done” );
```

```
    },
    deferWork: function() {
        //延迟 10s 执行
        FunctionExt.defer( this.doWork, 10 * 1000, this, [parm] );
    }
}
```

- 弹出窗口

确认框：$confirm(contentHTML, okCallback, cancelCallback, closeCallback)
提示框：$alert(contentHTML, okCallback, closeCallback, closeTimeout)

2 开发准则

2.1 清晰的了解自己开发的页面上的所有元素

建议使用 HttpWatch（\\192.168.1.29\Share\Tools\HttpWatch.Professional_v6.0.14-DOA.rar）或 Fiddler2（http://fiddler2.com/dl/Fiddler2Setup.exe）一类的工具

2.1.1 检查 css 文件

一般情况下不应该超过两个 css，一个是 public08，一个是该页面单独的 css

2.1.2 检查 js 文件

一般情况下有两个 js 是一定为加载的：
公共库：systemall.js（提前加载）
管理区：managersystem08.js
另外还有一些:顶部搜索关键词使用到的数据：SearchKeywordData.m
*对于 my.mtime.com，还有 mcApplication.js 和 staticManager.js
如果你的页面还引入了别的 js 和 css，请确保一定是必需且是最小化的，同时我们的 js 和 css 都带有版本号，不要直接使用 www 下的 js 和 css

2.1.3 检查其它子请求

A、不允许出现 404
B、图片大于 30K 让 UI 检查

2.2 不准侵入 HTML，记得资源的清理和释放，避免内存泄露

- HTML 与 javascript 完全分离
- 资源一定要释放

例如：

```
//首页
var HomeClient = Mtime.Page.HomeClient;
HomeClient = Class.create();

Object.extend( HomeClient.prototype, {
    name: "HomeClient",

    //初始化
    initialize: function(){
        this.initializeServerData();
        this.initializeDOM();
        this.initializeEvent();
        this.initializeControl();
        this.load();
    },

    initializeField: function() {
    },

    //初始化服务器端数据
    initializeServerData: function() {
        this.url = homeVariables.url;
    },

    //初始化DOM元素
    initializeDOM: function() {
        //
        this.emptyFeedRegion = $( "emptyFeedRegion" );
    },

    destroyDOM: function() {
        this.emptyFeedRegion = null;
    },

    //初始化DOM事件
    initializeEvent: function() {
        Event.observe( window, "unload", this.close.bind( this ) );
    },
```

```
//销毁DOM事件
destroyEvent: function() {
},

//初始化控件
initializeControl: function() {
    this.notification = new Notification();
    //猜你喜欢电影
    this.attentionMovie = new AttentionMovie( this.url );
    //没有动态
    if ( this.emptyFeedRegion ) {
        this.emptyFeed = new EmptyFeed();
    }
    //每日任务
    this.task = new Task();
    //谁访问过我
    $loadJs( "/js/08/userAvatar.js", function() {
        UsersAvatars.register( ["feedListRegion", "visitedListRegion"] );
    } );
    //我的订阅
    this.subscription = new Subscription();
},

destroyControl: function() {
     this.notification.close();
     //
     if ( this.attentionMovie ) {
         this.attentionMovie.close();
     }
     //
     if ( this.emptyFeed ) {
         this.emptyFeed.close();
     }
    //每日任务
    this.task.close();
    //
    this.subscription.close();
},

load: function() {
     //第一次初始化
     if ( userVariables.userData.MessageFlag === 1 ) {
           $loadSubJs( "/home/help.js", function() {
```

```
                    new HelpClient();
                } );
            }
        },


        //资源清理
        close: function() {
            this.destroyControl();
            this.destroyEvent();
            this.destroyDOM();
        }
} );
```

2.3 尽量少的操作 DOM

2.3.1缓存已经访问过的元素

例如：

```
this.emptyFeedRegion = $( "emptyFeedRegion" );
```

之后对"emptyFeedRegion"的操作都通过 this.emptyFeedRegion

2.3.2"离线"更新节点

例如：

```
for ( var i = 0, count = 10; i < count; ++i ) {
    var node = document.createElement( “div” );
    document.body.appendChild( node );
}
```

应该重构为：

```
var nodes = [];
for (var i = 0, count = 10; i < count; ++i) {
    nodes.push("<div>1</div>");
}
var node = document.createElement( "div" );
node.innerHTML = nodes.join("");
document.body.appendChild(node);
```

2.4 使用直接量

- 数组

var foo = new Array();

=>

var foo = [];

- 对象

var foo = new Object();

=>

var foo = {};

- 正则

var reg=new RegExp();

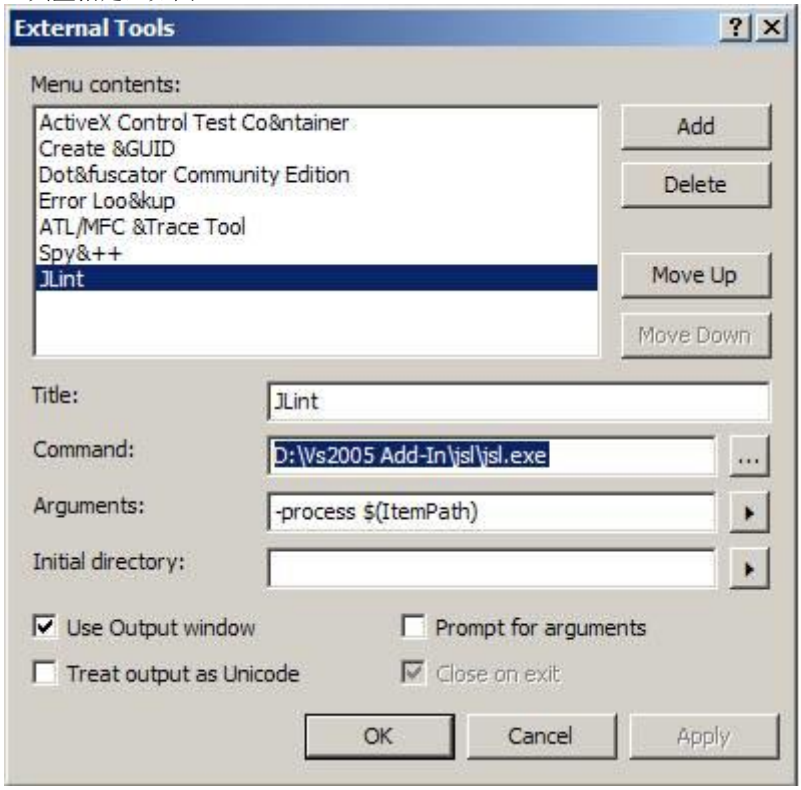
=>

var reg = /../;

2.5 采用 Jlint 检查脚本文件

不能有错误，尽量避免警告

Jlint 安装方法：
下载地址：\\192.168.1.29\Share\Tools\Vs2005 Add-In.rar,解压到 D 盘，然后在 VisualStudio 的扩展工具里指定，如图：



Command：D:\Vs2005 Add-In\jsl\jsl.exe
Args：-process $(ItemPath)

使用方法：打开某一 js 文件，从“tools”下面选择 Jlint 即可，这时会从输出窗口里看到结果

3 控件

3.1 弹出窗口

3.1.1 提示框

方法说明：\Web\Community\Js\08\Dialog.js

```
function $alert(
contentHTML/*HTML内容*/,
okCallback/*确定按钮的回调方法*/,
closeCallback/*窗口关闭时的回调方法*/)
```

例子：\Web\Community\Test\TestRss.html

```
$alert( "请先<a href=\"#\" class=\"ml6 mr6\">登录</a>后再进行此操作",  function()
{ alert( "ok" );}, function() { alert( "close" );});
```

3.1.2 确认框

方法说明：\Web\Community\Js\08\Dialog.js

```
function $confirm(
/*HTML内容*/
contentHTML,
/*确定按钮的回调方法*/
okCallback,
/*取消按钮的回调方法*/
```

```
cancelCallback,
/*窗口关闭时的回调方法*/
closeCallback)
```

例子：\Web\Community\Test\TestRss.html

```
$confirm( "请先<a href=\"#\" class=\"ml6 mr6\">登录</a>", function() { alert( "ok" );},
function() { alert( "cancel" );}, function() { alert( "close" );} );
```

3.1.3 自定义弹出窗口

基本分为 3 部分：窗口头部（带关闭按钮）区域，实际 HTML 区域和按钮区域，头部的处理在\Web\Community\Js\08\BaseWindow08.js，一般弹出窗口从 Dialog 派生就可以了，特别复杂的可以从 BaseWindow08.js 派生。

方法说明：\Web\Community\Js\08\Dialog.js

```
new Dialog( {
    //窗口的ClassName，一般是用来定义窗口的宽
    windowClassName: "",
    //窗口的Style，一般是用来定义窗口的宽
    windowStyle: "",
    //窗口的实际HTML
    content: "",
    //是否直接打开窗口
    isDirectOpen: true,
    //按钮区域的样式
    buttonRegionClass: "",
    //按钮数组
```

```
            /*
            buttonOption{
                title:标题,
                buttonStyle:button的class,
                onmouseover:当鼠标移动到Button上面执行的方法(主要对于CancelButton),
                onmouseout:当鼠标移动到Button外面执行的方法(主要对于CancelButton),,
                callback:点击后的回调方法
            }
        buttons: null
//当点击窗口的X时的回调
        onClickCloseCallback: null,
        //当窗口准备好时的回调
        readyCallback: null,
        //当窗口关闭时的回调
        closeCallback: null
    } );
```

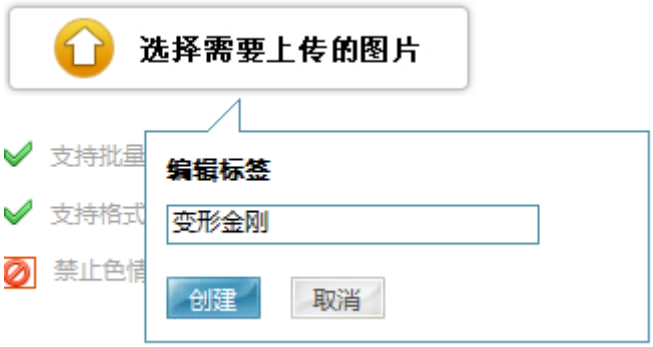
例子：

```
        new Dialog( {
            //设定Window的宽
            windowStyle: "width: 340px;",
            //设定Window的实际内容
            content: Builder.node( "div", { className: "pl12 pr12" }, [
                Builder.node( "h4", { className: "lh25" }, [
                    "一句话点评：",
                    Builder.node( "em", { className: "c_666" }, [
                        "(不少于5个汉字)"
                    ] )
                ] ),
                Builder.node( "textarea", { id: "commentContent", className: "bor_a5", style:
"width: 310px; height: 170px;" } ),
                Builder.node( "div", { className: "line_dot mt12 mb12" }, [
                ] )
            ] ),
            //设定Window的按钮区域的实际内容
            buttonRegionClass: "pl12 pr12 tl mb12",
            //设定Window的按钮数组
            buttons: [{
                title: "确定",
                buttonStyle: "btn_square_hover mr15",
                callback: function( dialog ) {
                    $alert( this.commentContent.value );
                    dialog.close();
                }.bind( this )
            },{
```

```
            title: "取消",
            buttonStyle: "btn_square",
            callback: function( dialog ) {
                $alert( "取消" );
                dialog.close();
            }.bind( this )
        }],
        //当窗口Ready时的回调
        readyCallback: function( dialog ) {
              this.commentContent = dialog.getEl( "commentContent" );
        }.bind( this ),
        //当窗口关闭回调
        closeCallback: function () {
            this.commentContent = null;
        }.bind( this )
    } );
```

一个更复杂的例子见：\Web\Community\Js\08\selectLocation.js中ChoiceLocationDialog对dialog的使用

3.1.4 跟随控件弹出窗口



基本类似于对 Dialog 的调用，如：

```
//弹出对话框
new DialogBox( {
        content: Builder.node( "div", [
            Builder.node( "h3", [
                "编辑标签"
            ] ),
            Builder.node( "p", { className: "mt5" }, [
                Builder.node( "input", { type: "text", className: "input_title", style:
"width:180px;", value: "变形金刚" } )
            ] )
        ] ),
        element: this.browseButton, //触发点击的DOM元素
        buttons: [ {
          title: "创建",
            buttonStyle: "btn_blue2 mr15",
            callback: function() {
            }.bind( this ) }, {
          //
            title: "取消",
            buttonStyle: "btn_gray2",
            onmouseover: "this.className='btn_blue2'",
            onmouseout: "this.className='btn_gray2'",
            callback: function( dialog ) {
                if ( this.cancelCallback ) {
                    this.cancelCallback( dialog );
                }
                dialog.close();
            }.bind( this ) } ],
        readyCallback: function( dialog ) {}, //窗口创建后的回调事件，便于客户端处理
        closeCallback: function( dialog ) {} //窗口关闭后的回调事件，用于销毁事件等
```

```
} );
```

3.1.5 成功、失败提示


```
new SuccessBox( {
    boxRegion: "resultRegion",
    messages: ["成功！", "成功！<a href=\"http://www.mtime.com\">mtime.com</a>"]
} );
```


```
new FailureBox( {
    boxRegion: "resultRegion",
    messages: ["失败！", "失败！<a href=\"http://www.mtime.com\">mtime.com</a>"]
} );
```

其中：resultRegion 为在页面对应位置的 div 的 id

3.2 文本框 Textbox

使用：

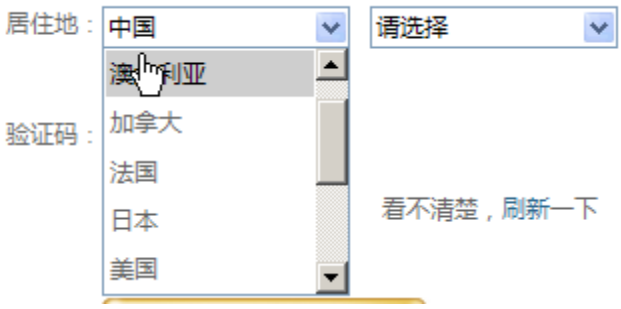
初始化：

```
var titleTextbox = new Textbox( {
    text: "input的ID或实例",
```

```
        watermarkText: "请输入关键字"
    } );

    方法：
    //获取值
    titleTextbox.getValue();
    //关闭
    titleTextbox.close();
```

3.3 模拟下拉框



1、页面

```
<div id="countrySelectRegion" class="i_select fl mr6" style="width: 100px; _margin-left:3px;">
<span id="countrySelectDefaultOption" class="c_000">请选择</span>
<a id="countrySelectButton" onclick="return false;" href="#" title="选择地区" class="sel">选择地区</a>
<div id="countrySelectListRegion" class="openlist" style="display:none;width: 122px;"></div>
</div>
```

2、Js

```
        var select = new DropDownList( {
            //选择区域
            selectRegion: this.selectTypeRegion,
            //默认选项
             selectDefaultOption: this.selectTypeDefaultOption,
             //选择按钮
             selectButton: this.selectTypeButton,
             //列表区域
             listRegion: this.selectTypeListRegion,
             //列表数据
             selectDatas: this.selectTypeDatas,
             //列表数据中Id列名
             idFieldName: "id",
             //列表数据中title列名
             titleFieldName: "text",
             //初始显示的列表项个数
```

```
            showCount: 6,
            //列表项发生改变后回调
            onChangeCallback: this.onChangeTypeCallback.bind( this )
        } );
```

3、例子：

```
        //国家选择下拉框
        this.countryDropDownList = new DropDownList( {
            //选择区域
            selectRegion: this.countrySelectRegion,
            //默认选项
            selectDefaultOption: this.countrySelectDefaultOption,
            //选择按钮
            selectButton: this.countrySelectButton,
            //列表区域
            listRegion: this.countrySelectListRegion,
            //列表项模板
            itemTemplate: "<a href=\"#\" onclick=\"return false;\"
value=\"#{Id}\">#{NameCn}</a>",
            //列表数据中Id列名
            idFieldName: "Id",
            //列表数据中title列名
            titleFieldName: "NameCn",
            //初始显示的列表项个数
            showCount: 6,
            //列表项发生改变后回调
            onChangeCountryCallback: this.onChangeCountryCallback.bind( this )
        } );
```

参见： \Web\Community\Js\08\selectLocation.js

3.4 菜单切换

3.4.1 Tab 切换

使用：

```
    new TabPanel( {
        //菜单区域
        menuRegion: "feedMenu",
        //内容区域
```

```
        contentRegion: "feedMenuContent",
        //初始化菜单索引
        initializeMenuIndex: 1,
        //菜单TagName
        menuTagName: "li"
    } );
```

见：\Web\Community\Test\TestRss.html

3.4.2 accordionPanel


```
new AccordionPanel( {
        //菜单区域
        menuRegion: "topTenAccordion",
        //初始化菜单索引
        initializeMenuIndex: 0,
        //菜单TagName
```

```
        menuTagName: "div"
    } );
```

见：\Web\Community\Test\TestRss.html

3.4.3 自定义菜单

基本分为 2 部分：菜单区域和菜单内容区域，菜单区域主要起到控制作用，处理鼠标悬停及点击，具体的内容显示的方式在派生类中实现。基类：Web\Community\Js\08\menuBase.js 以下几个例子都来自 MenuBase 的派生类

1、首页精彩导航

2、首页内容导航

3、导航

使用

```
var CustomMenu = Class.create();
```

```
Object.extend( Object.extend(CustomMenu.prototype, MenuBase.prototype ), {

    //初始化菜单元素
    initializeMenu: function() {
    },

    //关闭菜单元素
    destroyMenu: function() {
    },

    //当菜单活跃时
    onMenuActiveCallback: function( menu, index ) {
    },

    //当菜单Inactive时
    onMenuInactiveCallback: function( menu, index ) {
    },

    //当鼠标在菜单上时
    onMenuOverCallback: function( menu, index ) {
    },

    //当鼠标移开菜单时
    onMenuOutCallback: function( menu, index ) {
    },

} );
```

3.5 登录控件

1、

```
    /// <summary>
    /// 登录控件类型
    /// </summary>
    public enum LoginType
    {
        /// <summary>
        /// 首页
        /// </summary>
        Index,
        /// <summary>
        /// 电影
        /// </summary>
        Movie,
        /// <summary>
        /// 影人
        /// </summary>
        Person,
        /// <summary>
        /// 视频
        /// </summary>
        Video,
        /// <summary>
        /// 图片
        /// </summary>
        Picture,
        /// <summary>
        /// 博客
        /// </summary>
        Blog,
        /// <summary>
        /// 群组
        /// </summary>
        Group,
        /// <summary>
        /// 游戏
        /// </summary>
        Game,
        /// <summary>
        /// 电影院
        /// </summary>
        Theater,
        /// <summary>
        /// 数据库
```

```
        /// </summary>
        DataBase
    }
```

2、实例化

```
loginCtrl = new LoginCtrl ( this, LoginType.Index );
```

3、输出

```
ReplaceTagString ( "<!--LoginCtrl-->", loginCtrl.ToString() );
```

如果需要修改，请修改：Controls\Controls\08\Utility\Login\Impl下的对应文件

3.6 导航控件

1、初始化：

```
navigationBar = new NavigationBar08 ( this, MainNavigaionType.Home,
SubNavigaionType.NotSet );
```

其中：MainNavigaionType.Home为主导航类型，SubNavigaionType.NotSet为二级导航类型

2、输出：

```
ReplaceTagString ( "<!--NavigationBar-->", navigationBar.ToString () );
```

3.7 头像控件

1、生成 HTML

通过：\MtimeMovieCommunity\Controls\Controls\08\Utility\UserAvatar.cs 的方法：Render 来生成。（目前只有 48X48 最普通的样式）如果 HTML 有较大的改动，请先和产品和我确认后再添加。

2、初始化

- 初始化一个范围内的用户头像，需要传递外围div区域的ID

```
UsersAvatars.register( "gameLeavingMessageRegion" );
```

- 初始化单一的用户头像

```
new UserAvatar( {
    userId: userId,
    //头像区域
    avatarRegion: avatarRegion,
    //显示菜单按钮
    showMenuButton: showMenuButton,
    //菜单区域
    menuRegion: menuRegion
} )
```

3、例子：（游戏频道页）

\MtimeMovieCommunity\Web\Community\08\Game\Quiz\Index.html

```
<script type="text/javascript">
//用户头像
$loadJs( "/js/08/userAvatar.js", function() {
    UsersAvatars.register( "gameLeavingMessageRegion" );
} );
</script>
```

3.8 电影操作控件

1、指定最外层区域的 ID

```
<div id="attentionMovieRegion" class="content">
  <%= movieListHtml %>
</div>
```

2、指定每个电影区域的电影 ID

```
<li movieid="<%= movie.MovieID %>">
  <div class="table">
    <div class="t_r">
      <div class="td pr9 c_666">
        <a href="<%= url %>" target="_blank" title="<%= movie.AltTitle %>"><img
alt="<%= movie.AltTitle %>" class="img_box" src="<%=
Globals.GetSiteUrls().GetImageUrls().GetMovieTitleImageUrl( Mtime.Community.Components.Co
nstants.MovieCoverType.W75H100, movie.MovieID ) %>" /></a>
……
```

3、设定各个操作的 action

```
<input action="hasSeen" class="btn_gray2" type="button" value="看过" />
<input action="wantToSee" class="btn_gray2" type="button" value="想看" />
<input  action="notInterest" class="btn_gray2" type="button" value="没兴趣" />
如果有收藏：
<a action="favorite" onclick="return false;" href="#" class="fr normal btn_fav">收 藏</a>
```

4、初始化

```
//看过、想看、没兴趣
```

```
$loadJs( "/js/movieManager.js", function() {
    new MovieManagerClient( {
        container: "movieRecommendList"
    } );
} );
//收藏
$loadJs( "/js/favorite.js", function() {
    new FavoriteManagerClient( {
        container: "movieRecommendList"
    } );
} );
```

4 静态页面

4.1 文件中调用整站 Header 和 Footer 及 Manager

1、设置页面中Head的ID，<head id="head">

2、在Head中设置Css

```
<link rel="stylesheet" href="http://i.mtime.cn/static/css/public_08.css" type="text/css"
media="all" />
<link rel="stylesheet" href="http://i.mtime.cn/static/css/reg_08.css" type="text/css"
media="all" />
```

3、定义Header的位置，注意：/utility/top/目前支持common，movie，blog

```
<!--top Begin-->
<script type="text/javascript" src="/utility/top/common/"></script>
<div id="top"></div>
<!--top End-->
```

4、定义Footer的位置

```
<!--bottom Begin-->
<div id="contentEnd"></div>
<div id="bottom"></div>
<!--bottom End-->
```

5、在页面的最后增加脚本

```
<!--Script Begin-->
```

```
<script type="text/javascript">document.write( unescape( "%3Cscript
src='http://i.mtime.cn/static/js/systemall.js'
type='text/javascript'%3E%3C/script%3E" ) );</script>
 <!--
    $loadJs( "/js/08/staticManager.js", function() {
        new StaticManager( {
            //主导航类型
            mainNavigaionType: 2,
            //二级导航类型
            subNavigaionType: -1
        } );
    } );
//-->
</script>
<!--Script End-->
```

注意：

1、如果该静态文件是可控的，那么css,js的路径在上线后需要将static替换为实际版本号

2、所有的静态文件需要在最下方增加google和Webtrends的统计

```
<div style="display: none">
<script type="text/javascript" id="dcs4nkrx26o9cegqpjp6x8jvt_5i7u"
language="JavaScript" src="http://webtrends.mtime.com/mtime_dcs_tag.js"></script>
<noscript><img alt="" border="0" height="1" name="DCSIMG"
src="http://webtrends.mtime.com/dcs4nkrx26o9cegqpjp6x8jvt_5i7u/njs.gif?dcsuri=/noja
vascript&WT.js=No" width="1"></noscript> <script
src="http://www.google-analytics.com/urchin.js" type="text/javascript"></script>
<script type="text/javascript"> _uacct = "UA-257914-4"; try {
    urchinTracker();
} catch ( e ) {;}
</script>
</div>
```

参见：http://www.mtime.com/404.htm

4.2 资源引用

图、js、css 通过绝对地址 i.mtime.cn 来引用

例如：http://www.mtime.com/act/intel/mva/index.html

1、如果是引用整站资源：

如：http://www.mtime.com/css/public_08.css

请修改为：**http://i.mtime.cn/static**/css/public_08.css

(我们每次发版本的时候会把整站的资源文件拷贝一份放到 static 下面)

2、如果是自行上传到 act 下面的资源

如：images/i_bg.jpg

请修改为：**http://i.mtime.cn/act/**intel/mva/images/i_bg.jpg

(www.mtime.com/act/下面的所有资源会和 i.mtime.cn/act/同步，并且它没有缓存问题)