

```

package br.com.optimedia.player.view
{
    import br.com.optimedia.assets.NotificationConstants;
    import br.com.optimedia.assets.vo.QuestionVO;
    import br.com.optimedia.assets.vo.SlideVO;
    import br.com.optimedia.player.PlayerFacade;
    import br.com.optimedia.player.model.PlayerSlideManagerProxy;
    import br.com.optimedia.player.view.components.MediaPopUp;
    import br.com.optimedia.player.view.components.QuestionView;

    import flash.events.MouseEvent;

    import mx.collections.ArrayCollection;
    import mx.controls.Alert;
    import mx.managers.BrowserManager;
    import mx.managers.IBrowserManager;
    import mx.rpc.events.ResultEvent;

    import org.puremvc.as3.multicore.interfaces.INotification;
    import org.puremvc.as3.multicore.patterns.mediator.Mediator;

    public class PlayerModuleMediator extends Mediator
    {
        public static const NAME:String = 'PlayerModuleMediator';

        private var proxy:PlayerSlideManagerProxy;

        public function PlayerModuleMediator(viewComponent:Object=null)
        {
            super(NAME, viewComponent);
        }

        override public function onRegister():void
        {
            trace(NAME+".onRegister()");

            view.urlBtn.addEventListener(MouseEvent.CLICK, showURL);
            view.addEventListener(PlayerModule.GET_QUESTION_EVENT, getQuestion);

            proxy = facade.retrieveProxy( PlayerSlideManagerProxy.NAME ) as
            PlayerSlideManagerProxy;

            if( view.mode == PlayerModule.PLAYER_MODE ) {
                if( PlayerFacade(facade).allowNavigationLog ) {
                    getLastViewedSlide();
                    view.prevBtn.addEventListener(MouseEvent.CLICK, registerNavigation);
                    view.nextBtn.addEventListener(MouseEvent.CLICK, registerNavigation);
                }
                else {
                    proxy.getPresentation( view.presentationID );
                }
            }
            else if( view.mode == PlayerModule.PREVIEW_MODE ) {
                view.urlBtn.visible = false;
            }
        }
    }
}

```

```

    }

}

override public function onRemove():void {
    trace(NAME+".onRemove'()");
}

public function get view():PlayerModule
{
    return viewComponent as PlayerModule;
}

override public function listNotificationInterests():Array
{
    return [NotificationConstants.GET_PRESENTATION_FOR_PLAYER,
            NotificationConstants.GET_QUESTION_RESULT,
            NotificationConstants.GET_LAST_VIEWED_SLIDE_RESULT];
}

override public function handleNotification(note:INotification):void
{
    switch (note.getName())
    {
        case NotificationConstants.GET_PRESENTATION_FOR_PLAYER:
            setSlidesArray( new ArrayCollection( note.getBody() as Array ) );
            break;
        case NotificationConstants.GET_QUESTION_RESULT:
            var questionView:QuestionView = mediaPopUp.getChildByName(QuestionView.NAME)
as QuestionView;
            var questionVO:QuestionVO = note.getBody() as QuestionVO;
            questionView.questionVO=questionVO;
            //view.p( new ArrayCollection( note.getBody() as Array ) );
            break;
        case NotificationConstants.GET_LAST_VIEWED_SLIDE_RESULT:
            view.slideID = note.getBody() as int;
            proxy.getPresentation( view.presentationID );
            break;
        default:
            break;
    }
}

private function setSlidesArray(slidesArray:ArrayCollection):void {
    view.slidesArray = slidesArray;
    if( view.slidesArray.length <= 0 ) {
        Alert.show("Não há slides nesta apresentação");
        view.enabled = false;
    }
    else {
        if( view.slideID == 0 ) {
            view.setSlide( view.slidesArray[0] as SlideVO );
        }
    }
}

```

```
        else {
            for each( var slide:SlideVO in view.slidesArray ) {
                if( slide.slide_id == view.slideID ) {
                    view.setSlide( slide );
                }
            }
        }
        view.enabled = true;
    }
}

private function showURL(event:MouseEvent):void {
    var end:IBrowserManager = BrowserManager.getInstance();
    end.init();
    var url:String;
    url = end.url.split("#")[0]+"&s="+view.slideID;
    Alert.show(url, "Link para este slide:");
}

private var mediaPopUp:MediaPopUp;

private function getQuestion(event:ResultEvent):void {
    mediaPopUp=event.result as MediaPopUp;
    proxy.getQuestion(mediaPopUp.mediaVO.media_id as int);
}

private function registerNavigation(e:MouseEvent):void {
    proxy.registerNavigation(PlayerFacade(facade).userID, view.presentationID,
view.slideID);
}

private function getLastViewedSlide():void {
    proxy.getLastViewedSlide(PlayerFacade(facade).userID, view.presentationID);
}
}
```