```
package br.com.optimedia.autor.view
{
    import br.com.optimedia.assets.CommandConstants;
    import br.com.optimedia.assets.NotificationConstants;
    import br.com.optimedia.assets.StripHtmlTags;
    import br.com.optimedia.autor.model.SubjectManagerProxy;
    import br.com.optimedia.autor.view.components.PublishPresentationPopUp;

    import flash.events.MouseEvent;

    import mx.controls.Alert;

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

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

        private var proxy:SubjectManagerProxy;

        public function PublishPresentationPopUpMediator(viewComponent:Object=null)
        {
            super(NAME+viewComponent.uid, viewComponent);
        }

        override public function onRegister():void
        {
            trace(NAME+".onRegister()");
            view.saveBtn.addEventListener(MouseEvent.CLICK, saveHandler);
            view.cancelBtn.addEventListener(MouseEvent.CLICK, closeHandler);
            proxy = facade.retrieveProxy( SubjectManagerProxy.NAME ) as SubjectManagerProxy;
            proxy.getSections();
        }

        override public function onRemove():void {
            view.removePopup();
        }

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

        override public function listNotificationInterests():Array
        {
            return [NotificationConstants.GET_SECTIONS_RESULT,
                    NotificationConstants.PUBLISH_PRESENTATION_OK,
                    NotificationConstants.UNPUBLISH_PRESENTATION_OK];
        }

        override public function handleNotification(note:INotification):void
        {
            switch (note.getName())
            {
                case NotificationConstants.GET_SECTIONS_RESULT:
                    var array:Array = note.getBody() as Array;
                    var newArray:Array = [];
                    for each( var item:Object in array ) {
                        var obj:Object = new Object();
                        obj.section_id = item.section_id;
                        obj.title = StripHtmlTags.strip( item.title );
                        //var aux:String = StripHtmlTags.strip( item.title );
                        newArray.push( obj );
                    }
                    view.sectionsArray = newArray;
                    break;
                case NotificationConstants.PUBLISH_PRESENTATION_OK:
                    Alert.show("Tema publicado com sucesso", "OK");
                    closeHandler(null);
                    break;
                case NotificationConstants.UNPUBLISH_PRESENTATION_OK:
                    Alert.show("Tema despublicado com sucesso", "OK");
                    closeHandler(null);
                    break;
                default:
                    break;
            }
        }

        private function closeHandler(event:MouseEvent):void {
            sendNotification( CommandConstants.PUBLISH_PRESENTATION_DISPOSE, view );
        }

        private function saveHandler(event:MouseEvent):void {
            if( view.currentState == "unpublish" ) {
                proxy.unpublishPresentation( view.presentationVO.presentation_id,
view.presentationVO.section_id );
            }
            else {
                proxy.publishPresentation( view.presentationVO,
view.sectionComboBox.selectedItem.section_id );
            }
        }
    }
}
```