

```
package br.com.optimedia.autor.view
{
    import br.com.optimedia.assets.NotificationConstants;
    import br.com.optimedia.autor.AutorFacade;
    import br.com.optimedia.autor.model.SubjectManagerProxy;

    import mx.controls.Alert;
    import mx.rpc.events.FaultEvent;
    import mx.rpc.events.ResultEvent;
    import mx.rpc.http.HTTPService;

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

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

        private var subjectManagerProxy:SubjectManagerProxy;

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

        override public function onRegister():void
        {
            trace(NAME+'.onRegister()');
            view.visible = false;
            subjectManagerProxy = facade.retrieveProxy( SubjectManagerProxy.NAME ) as
SubjectManagerProxy;
            whoAmI();
            //view.addEventListener( Autor.DO_UNLOCK_EVENT, doUnlock );
        }

        override public function onRemove():void
        {

        }

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

        override public function listNotificationInterests():Array
        {
            return [NotificationConstants.BEGIN_PRESENTATION_EDIT,
                    NotificationConstants.BACK_TO SUBJECT_MANAGER];
        }

        override public function handleNotification(note:INotification):void
        {
            switch (note.getName())
            {
```

```
        case NotificationConstants.BEGIN_PRESENTATION_EDIT:
            view.viewStack.selectedIndex++;
            break;
        case NotificationConstants.BACK_TO SUBJECT_MANAGER:
            view.viewStack.selectedIndex--;
            break;
        default:
            break;
    }
}

private function whoAmI():void {
    var service:HTTPService = new HTTPService();
    service.url = 'http://174.122.24.74/sinase.moodle/autor/whoami.php';
    service.resultFormat = "e4x";
    service.showBusyCursor = true;
    service.addEventListener(ResultEvent.RESULT, resultHandler);
    service.addEventListener(FaultEventFAULT, faultHandler);
    service.send();
}
private function resultHandler(event:ResultEvent):void {

    var roleID:int = event.result.roleID;
    var userID:int = event.result.userID;
    //HABILITE ESSAS DUAS VARIÁVEIS PARA TESTAR LOCALMENTE
    //userID = 10;
    //roleID = 1;
    //SE NÃO ESTIVER LOGADO NO MOODLE
    if( roleID == 0 ) {
        Alert.show("É necessário logar-se no Moodle antes.", "Erro");
    }
    //SE FOR ADMIN
    else if( roleID == 1 ) {
        view.visible = true;
        AutorFacade(facade).userRole = AutorFacade.IS_ADMIN;
        AutorFacade(facade).userID = userID;
        view.addModuleManager();
        view.addSlideEditor();
    }
    //SE FOR AUTOR
    else if( roleID == 2 ) {
        view.visible = true;
        AutorFacade(facade).userRole = AutorFacade.IS_AUTHOR;
        AutorFacade(facade).userID = userID;
        view.addModuleManager();
        view.addSlideEditor();
    }
    //SE FOR CONTEUDISTA (EDITOR)
    else if( roleID == 8 ) {
        view.visible = true;
        AutorFacade(facade).userRole = AutorFacade.IS_EDITOR;
        AutorFacade(facade).userID = userID;
        view.addModuleManager();
        view.addSlideEditor();
    }
}
```

```
        }
        //SE FOR OBSERVADOR
        else if( roleID == 9 ) {
            view.visible = true;
            AutorFacade(facade).userRole = AutorFacade.IS_OBSERVER;
            AutorFacade(facade).userID = userID;
            view.addModuleManager();
            view.addSlideEditor();
        }
        //SE NÃO TIVER PREMISSÃO DE ADMIN, AUTOR OU CONTEUDISTA
        else {
            Alert.show( "Você não tem permissão para editar", "Atenção" );
        }
    }
private function faultHandler(event:FaultEvent):void {
    trace(event);
}

/* private function doUnlock(event:Event):void {
    if( view.viewStack.selectedChild is PresentationEditor ) {
        subjectManagerProxy.unlockPresentation( PresentationEditor(
view.viewStack.selectedChild).presentationVO.presentation_id );
    }
} */
}
```