

Forum:[Hack Forums](#)

Thread: [java.lang.reflect.InvocationTargetException](#)

black_lord - June 8, 2017, 1:22 pm

Hi guys, after a 12h of struggling and searches trough Google, I decided to ask you for help. Just can't figure it out. I've created JavaFX application with two stages. Every stage has its own controller and .fxml. What I'm trying to do is call method in Controller1 which contains new Thread code through its object on Controller2, and I end up with java.lang.reflect.InvocationTargetException.

Here is the code:

Controller1:

Code public class Controller1 implements Initializable {

```
public Label lblState;
```

```
private Service<Void> backgroundThread;
```

```
public void doWork() {
```

```
    //Creates a new Thread
```

```
    backgroundThread = new Service<Void>() {
```

```
        @Override
```

```
        protected Task<Void> createTask() {
```

```
            return new Task<Void>() {
```

```
                @Override
```

```
                protected Void call() throws Exception {
```

```
                    for (int i = 0; i<100000; i++){
```

```
                        updateMessage("i: "+i);
```

```
                    }
```

```
                    return null;
```

```
                }
```

```
            };
```

```
        }
```

```
    };
```

```
    backgroundThread.setOnSucceeded(new EventHandler<WorkerStateEvent>() {
```

```
        @Override
```

```
        public void handle(WorkerStateEvent event) {
```

```
            System.out.println("Done!");
```

```
            lblState.textProperty().unbind();
```

```
        }
```

```
    });
```

```
    lblState.textProperty().bind(backgroundThread.messageProperty());
```

```
    backgroundThread.restart();
```

```
}
```

```
}
```

Controller2:

Code public class Controller2 {

```
public void btnStartClick(){
```

```
    Controller1 controller1 = new Controller1();
```

```

        controller1.doWork();
    }
}

```

If I call doWork() from any method in Controller1, code works fine. How to call this method from Controller2 properly?

Thanks.

CAFEEFAC - June 8, 2017, 2:22 pm

Could you post your entire source please? All classes, and the FXML files as well. Also, the stack trace.

black_lord - June 8, 2017, 5:47 pm

I wanted to paste a download link and make it easier but I need 50 posts for that. ?

So, here are the classes:

Controll package contains Controller1 and Controller2 class.

Controller1:

Code package controll;

```

import javafx.concurrent.Service;
import javafx.concurrent.Task;
import javafx.concurrent.WorkerStateEvent;
import javafx.event.EventHandler;
import javafx.scene.control.Label;
import model.Main;
import model.Model2;
import javafx.event.ActionEvent;

public class Controller1 {

    public Label lblState;
    //Background Thread
    private Service<Void> backgroundThread;

    public void btnStartClick(ActionEvent actionEvent) {
        new Model2();
    }

    public void doWork(){
        //Creates a new Thread
        backgroundThread = new Service<Void>() {
            @Override
            protected Task<Void> createTask() {
                return new Task<Void>() {
                    @Override
                    protected Void call() throws Exception {
                        for (int i = 0; i<10000000; i++){
                            updateMessage("i: "+i);
                        }
                        return null;
                    }
                };
            }
        };
    }
}

```

```

};
backgroundThread.setOnSucceeded(new EventHandler<WorkerStateEvent>() {
    @Override
    public void handle(WorkerStateEvent event) {
        System.out.println("Done!");
        lblState.textProperty().unbind();
    }
});

lblState.textProperty().bind(backgroundThread.messageProperty());
backgroundThread.restart();
}

public void btnStartFromNewThreadClick(ActionEvent actionEvent) {
    doWork(); //Works fine when called from here
}

public void btnCloseClick(ActionEvent actionEvent) {
    Main.myPrimaryStage.close();
}
}

```

Controller2:

Code package controll;

```

import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

```

```

public class Controller2 {

    public TextField txtPortNumber;
    public Button btnStart;

    public void btnStartClick(){
        Controller1 con1 = new Controller1();
        con1.doWork(); //Throws InvocationTargetException when called from here
        Stage stage = (Stage) btnStart.getScene().getWindow();
        stage.close();
    }
}

```

Package model contains Main class and Model2 class.

Main:

Code package model;

```

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

```

```

import java.security.PublicKey;

```

```

public class Main extends Application {

```

```

public static Stage myPrimaryStage;
public static void main(String[] args) {
    launch(args);
}

@Override
public void start(Stage primaryStage) throws Exception{
    Parent root = FXMLLoader.load(getClass().getResource("../view/Controller1.fxml"));
    myPrimaryStage = primaryStage;
    myPrimaryStage.setTitle("MVC");
    myPrimaryStage.setScene(new Scene(root));
    myPrimaryStage.show();
}
}
}
Model2:
Code package model;

import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Modality;
import javafx.stage.Stage;

public class Model2
{
    Stage stage;

    public Model2(){
        try {
            Parent root = FXMLLoader.load(getClass().getResource("../view/Controller2.fxml"));
            stage = new Stage();
            stage.setTitle("Run new Thread");
            stage.initModality(Modality.APPLICATION_MODAL);
            stage.setResizable(false);
            stage.setScene(new Scene(root,224,67));
            stage.show();
        }catch (Exception e){
            //System.err.println(e);
        }
    }
}
}

```

Package view contains Controller1.fxml and Controller2.fxml. I can't paste whole .fxml file because of site firewall.

Controller1.fxml contains:

```

Code <MenuBar BorderPane.alignment="CENTER">
    <menus>
        <Menu mnemonicParsing="false" text="File">
            <items>
                <MenuItem fx:id="btnStart" mnemonicParsing="false" onAction="#btnStartClick"
text="Start from new Thread" />
                <MenuItem fx:id="btnStop" mnemonicParsing="false"
onAction="#btnStartFromNewThreadClick" text="Start from same Thread" />
            </items>
        </Menu>
    </menus>
</MenuBar>

```

```

        <MenuItem fx:id="btnClose" mnemonicParsing="false" onAction="#btnCloseClick"
text="Close" />
    </items>
</Menu>
<Menu mnemonicParsing="false" text="Help">
    <items>
        <MenuItem mnemonicParsing="false" text="About" />
    </items>
</Menu>
</menus>
</MenuBar>and

```

Code <Label fx:id="lblState" text="State">

Controller2.fxml contains:

Code <Button fx:id="btnStart" onAction="#btnStartClick" text="Start!" />

The exception occurs when I call con1.doWork(); inside Controller2 class. If I call doWork(); inside Controller1 class everything works fine.

Thanks for your time and help. ?

CAFEEFAC - June 8, 2017, 7:19 pm

I think I might know what the problem is.

The code works when you call it from within the Controller1 class because you are operating within the Controller1 instance itself. However, when you click on "Start!" on Model2 which calls the btnStartClick method, you are creating a new instance of the Controller1 class without using the FXMLoader class. Because of this, there won't be any values injected into the fields of your new Controller1 instance. If you were to have access to a FXMLoader instance, you could call its "getController" method to fetch the controller instance so you can call methods on it.

Here's a suggestion for what you could try to do:

Instead of specifying the controller inside the Controller2 FXML file, create an FXMLoader instance and set the controller via the "setController" method of the FXMLoader class. Inside the method call, pass in a new instance of the Controller2 class, and inside the parameter, put in the FXMLoader that was used in the main class.

Here's how to do it step-by-step:

1. Inside your Main class, add a field which is of type FXMLoader.
2. Inside the Main class' start method, instantiate the FXMLoader instance and use that to load the FXML file.
3. Set the controller afterwards with the FXMLoader#setController method, pass in a new instance of Controller1, and pass the FXMLoader itself as a parameter inside. For example:

```
Code loader.setController(new Controller1(loader));
```

4. Now that you have passed in an object of type FXMLoader, you need to change the constructor of the Controller1 class to take in that type of object as a parameter.
5. Inside the Controller1#btnStartClick method, add the reference to the FXMLoader into the parameter of the Model2 class.
6. Modify the constructor of the Model2 class to take in an object of type FXMLoader.
7. Inside the constructor of the Model2 class, set the controller of the Controller2 FXML file with the setController method, exactly like how we did it in the Main class. Example:

Code loaderTwo.setController(new Controller2(loader));

8. Modify the constructor of the Controller2 class to take in the object of type FXMLLoader.

9. Finally, use the FXMLLoader object passed as a parameter to fetch the Controller1 instance by calling the FXMLLoader#getController method.

If this works, it's up to you to improve this to make it even better, or to find your own solution to fetching the Controller1 instance. This is obviously not an elegant solution, but it was meant to be shown as an example of how you need a reference to the FXMLLoader class to be able to get the Controller1 instance. If I'm wrong about all of this, please reply back. Actually, please let me know if the solution worked or not. :)

black_lord - June 9, 2017, 3:43 am

Well, this could work but I'm having problem with setting controller. I've deleted `fx:controller="controll.Controller1"` from `Controller1.fxml` and setting it through `loader.setController(new Controller1(loader));` as you suggested. It seems it doesn't work because I'm getting "No controller specified for top level element". If I live `fx:controller="controll.Controller1"` then I get `javafx.fxml.LoadException: Controller value already specified`.

I think that `fx:controller` should be updated automatically after calling `loader.setController(new Controller1(loader));` but it doesn't.

Any suggestions?

CAFEEFAC - June 9, 2017, 6:24 am

(Jun 9, 2017 03:43 AM)black_lord Wrote: [\[CITE\]](#) Well, this could work but I'm having problem with setting controller. I've deleted `fx:controller="controll.Controller1"` from `Controller1.fxml` and setting it through `loader.setController(new Controller1(loader));` as you suggested. It seems it doesn't work because I'm getting "No controller specified for top level element". If I live `fx:controller="controll.Controller1"` then I get `javafx.fxml.LoadException: Controller value already specified`.

I think that `fx:controller` should be updated automatically after calling `loader.setController(new Controller1(loader));` but it doesn't.

Any suggestions?

You need to specify a controller in your top level element. I'm sorry I didn't mention this in my solution. First of all, find your top level element. This might be the MenuBar in your case unless there are other elements before that.

Then, add an attribute to the top element which says:

Code `fx:controller="view.Controller1"`

Edit this so that it has the right path.

You will have to specify a controller for your Controller2 class to your top level element in the Controller2 FXML.

black_lord - June 9, 2017, 11:10 am

My Controller1 class is inside controller package so I can't set Code `fx:controller="view.Controller1"` I can only set Code `fx:controller="controll.Controller1"` and then I get `javafx.fxml.LoadException: Controller value already specified`

I'm probably missing something or Code `loader.setController(new Controller1(loader));` doesn't work as expected.

Here are my classes:

Main:

Code public class Main extends Application {

```
    public static Stage myPrimaryStage;
    public FXMLLoader loader;
```

```
    public static void main(String[] args) {
        launch(args);
    }
```

@Override

```
    public void start(Stage primaryStage) throws Exception{
        loader = new FXMLLoader(getClass().getResource("../view/Controller1.fxml"));
        loader.setController(new Controller1(loader));
        Parent root = loader.load();
        myPrimaryStage = primaryStage;
        myPrimaryStage.setTitle("MVC");
        myPrimaryStage.setScene(new Scene(root));
        myPrimaryStage.show();
    }
```

}

Model2:

Code public class Model2

{

```
    Stage stage;
```

```
    public Model2 (FXMLLoader loaderObject){
        loaderObject.setController(new Controller2(loaderObject));
    }
```

```
    public Model2(){
        try {
            Parent root = FXMLLoader.load(getClass().getResource("../view/Controller2.fxml"));
            stage = new Stage();
            stage.setTitle("Run new Thread");
            stage.initModality(Modality.APPLICATION_MODAL);
            stage.setResizable(false);
            stage.setScene(new Scene(root,224,67));
            stage.show();
        }catch (Exception e){
            //System.err.println(e);
        }
    }
```

}

Controller1:

Code public class Controller1 {

```
    public Label lblState;
    //Background Thread
    private Service<Void> backgroundThread;
    FXMLLoader loader;
```

```

public Controller1 (FXMLLoader loaderObject){
    this.loader=loaderObject;
}

public void btnStartClick(ActionEvent actionEvent) {
    new Model2(loader);
}

public void doWork(){
    //Creates a new Thread
    backgroundThread = new Service<Void>() {
        @Override
        protected Task<Void> createTask() {
            return new Task<Void>() {
                @Override
                protected Void call() throws Exception {
                    for (int i = 0; i<100000000; i++){
                        updateMessage("i: "+i);
                    }
                    return null;
                }
            };
        }
    };
    backgroundThread.setOnSucceeded(new EventHandler<WorkerStateEvent>() {
        @Override
        public void handle(WorkerStateEvent event) {
            System.out.println("Done!");
            lblState.textProperty().unbind();
        }
    });

    lblState.textProperty().bind(backgroundThread.messageProperty());
    backgroundThread.restart();
}

public void btnStartFromNewThreadClick(ActionEvent actionEvent) {
    doWork(); //Works fine when called from here
}

public void btnCloseClick(ActionEvent actionEvent) {
    Main.myPrimaryStage.close();
}
}
Controller2:
Code public class Controller2 {

    private FXMLLoader loader;
    public Controller2(FXMLLoader loaderObject){
        this.loader = loaderObject;
    }
    public TextField txtPortNumber;

```

```

public Button btnStart;

public void btnStartClick(){
    Controller1 obj = loader.getController();
    obj.doWork();
    Stage stage = (Stage) btnStart.getScene().getWindow();
    stage.close();
}
}
Controller1.fxml:
Code BorderPane fx:id="borderpane" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
fx:controller="controll.Controller1"
Controller2.fxml:
Code AnchorPane fx:controller="controll.Controller2"
Some more ideas? ?

```

CAFEEFAC - June 9, 2017, 11:51 am

First of all, I have no idea why I wrote view.Controller1. I'll blame it on my lack of sleep. ;)

As for the controller already being specified, this probably has something to do with the fact that you're setting a new controller to the loaderObject inside the constructor of the Model2 class. Or is this an error that occurs before you have even gotten the chance to instantiate the object of the Model2 class?

Also, you have provided two constructors in the Model2 class, so the other one won't be called. Reduce this to one constructor, and just modify it to take a parameter of type FXMLLoader.

Afterwards, create a field of type FXMLLoader and set the one passed in the constructor to the field. After that, create a separate FXMLLoader instance which you will use for loading the Controller2 view, and do everything like you did in the Main class. The only thing I want you to use the loaderObject for is to pass it into the constructor of the Controller2 object which you will set as a controller.

Here is an example of what I want the Model2 class to look like:

```

Code public class Model2
{
    Stage stage;
    public FXMLLoader loaderObject;

    public Model2(FXMLLoader loaderObject){
        this.loaderObject = loaderObject;

        try {
            FXMLLoader c2Loader = new
FXMLLoader(getClass().getResource("../view/Controller2.fxml"));
            c2Loader.setController(new Controller2(loaderObject)); // See what I did here?
            Parent root = c2Loader.load();
            stage = new Stage();
            stage.setTitle("Run new Thread");
            stage.initModality(Modality.APPLICATION_MODAL);
            stage.setResizable(false);

```

```

        stage.setScene(new Scene(root,224,67));
        stage.show();
    }catch (Exception e){
        //System.err.println(e);
    }
}
}
}

```

black_lord - June 9, 2017, 3:31 pm

Quote:First of all, I have no idea why I wrote view.Controller1. I'll blame it on my lack of sleep. ;) Hahahaha, don't even sweat men. At first, I thought "there's no logic but let me try anyway" and then comes a red line cross a screen. That was a desperate attempt. ?

Quote:Or is this an error that occurs before you have even gotten the chance to instantiate the object of the Model2 class?

I think it happens before. I've commented next lines of code out and still getting same exception.

```

Code //loaderObject.setController(new Controller2(loaderObject));
//FXMLLoader m2loader = FXMLLoader.load(getClass().getResource("../view/Controller2.fxml"));
//m2loader.setController(new Controller2(loaderObject));
//Parent root = m2loader.load();
//stage.setScene(new Scene(root,224,67));

```

Quote:Also, you have provided two constructors in the Model2 class, so the other one won't be called. Reduce this to one constructor, and just modify it to take a parameter of type FXMLLoader. I've corrected this issue, and I'm not sure how I overlooked this. Maybe I need some sleep as well ?

This is my class now:

Code public class Model2

```

{
    Stage stage;
    FXMLLoader loader;

    public Model2(FXMLLoader loaderObject){
        this.loader = loaderObject;
        loaderObject.setController(new Controller2(loaderObject));
        try {
            FXMLLoader m2loader =
FXMLLoader.load(getClass().getResource("../view/Controller2.fxml"));
            m2loader.setController(new Controller2(loaderObject));
            Parent root = m2loader.load();
            stage = new Stage();
            stage.setTitle("Run new Thread");
            stage.initModality(Modality.APPLICATION_MODAL);
            stage.setResizable(false);
            stage.setScene(new Scene(root,224,67));
            stage.show();
        }catch (Exception e){
            //System.err.println(e);
        }
    }
}
}

```

End this is exception I'm getting:

Code Exception in Application start method

```

java.lang.reflect.InvocationTargetException
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:498)
  at com.sun.javafx.application.LauncherImpl.launchApplicationWithArgs(LauncherImpl.java:389)
  at com.sun.javafx.application.LauncherImpl.launchApplication(LauncherImpl.java:328)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:498)
  at sun.launcher.LauncherHelper$FXHelper.main(LauncherHelper.java:767)
Caused by: java.lang.RuntimeException: Exception in Application start method
  at com.sun.javafx.application.LauncherImpl.launchApplication1(LauncherImpl.java:917)
  at
com.sun.javafx.application.LauncherImpl.lambda$launchApplication$155(LauncherImpl.java:182)
  at java.lang.Thread.run(Thread.java:745)
Caused by: javafx.fxml.LoadException: Controller value already specified.
/C:/Users/X/Desktop/MVCProject/out/production/MVCProject/view/Controller1.fxml:15

  at javafx.fxml.FXMLLoader.constructLoadException(FXMLLoader.java:2597)
  at javafx.fxml.FXMLLoader.access$100(FXMLLoader.java:103)
  at javafx.fxml.FXMLLoader$ValueElement.processAttribute(FXMLLoader.java:914)
  at javafx.fxml.FXMLLoader$InstanceDeclarationElement.processAttribute(FXMLLoader.java:971)
  at javafx.fxml.FXMLLoader$Element.processStartElement(FXMLLoader.java:220)
  at javafx.fxml.FXMLLoader$ValueElement.processStartElement(FXMLLoader.java:744)
  at javafx.fxml.FXMLLoader.processStartElement(FXMLLoader.java:2707)
  at javafx.fxml.FXMLLoader.loadImpl(FXMLLoader.java:2527)
  at javafx.fxml.FXMLLoader.loadImpl(FXMLLoader.java:2441)
  at javafx.fxml.FXMLLoader.load(FXMLLoader.java:2409)
  at model.Main.start(Main.java:21)
  at
com.sun.javafx.application.LauncherImpl.lambda$launchApplication1$162(LauncherImpl.java:863)
  at com.sun.javafx.application.PlatformImpl.lambda$runAndWait$175(PlatformImpl.java:326)
  at com.sun.javafx.application.PlatformImpl.lambda$null$173(PlatformImpl.java:295)
  at java.security.AccessController.doPrivileged(Native Method)
  at com.sun.javafx.application.PlatformImpl.lambda$runLater$174(PlatformImpl.java:294)
  at com.sun.glass.ui.InvokeLaterDispatcher$Future.run(InvokeLaterDispatcher.java:95)
  at com.sun.glass.ui.win.WinApplication._runLoop(Native Method)
  at com.sun.glass.ui.win.WinApplication.lambda$null$148(WinApplication.java:191)
  ... 1 more
Exception running application model.Main
Code Caused by: javafx.fxml.LoadException: Controller value already specified.
/C:/Users/X/Desktop/MVCProject/out/production/MVCProject/view/Controller1.fxml:15
Line 15 in Controller1.fxml:
Code <BorderPane fx:id="borderpane" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0"
prefWidth="600.0" fx:controller="controll.Controller1">

```

I found this on Google but didn't have time to analyze this approach. I have exams in days to come, and first starts today. I've decided to share it with you.

Quote: If I understand correctly, you want to define the controller instance in Java code, instead of

letting the FXMLLoader instantiate it.

You can do this by calling `setController` on the FXMLLoader before you call `load()`. Note that this means you must create an FXMLLoader instance and invoke the instance `load()` method, and must not call the static `load(URL)` method:

```
Code MyController controller = new MyController();
FXMLLoader loader = new FXMLLoader(getClass().getResource("path/to/fxml"));
loader.setController(controller);
Parent root = loader.load();
```

If you use this technique, you must not use the `fx:controller` attribute in the root element of the FXML file.

Note that this gives you the ability to instantiate the controller any way you like - e.g. if you have controller classes which have constructors that take parameters you can use them here (controllers used by the default mechanism must have a zero-arg constructor).

More advanced but related information:

A related technique, but perhaps for more specialized use cases, is to set a controller factory on the FXMLLoader. The controller factory is a function that maps a `Class<?>` (the one defined by the `fx:controller` attribute) to an object (the controller). This essentially gives you programmatic control over the way the controllers are instantiated (typically using reflection). One use of this is if you have a model class:

```
Code public class Model { /* ... */ }
and several different controller classes that take a model reference as a constructor parameter.
Typically you would want to use the same model instance and pass it to all controllers. So you could
define a controller factory as follows:
Code Model model = new Model();
Callback<Class<?>, Object> controllerFactory = type -> {
    try {
        // look for a constructor with a single parameter of type Model:
        for (Constructor<?> c : type.getConstructors()) {
            if (c.getParameterCount() == 1 && c.getParameterTypes()[0] == Model.class) {
                return c.newInstance(model);
            }
        }
        // no constructor found, just invoke no-arg constructor as in default:
        return type.newInstance();
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
};
```

Then you can set the controller factory on your FXMLLoader(s):

```
Code FXMLLoader loader = new FXMLLoader(getClass().getResource("path/to/fxml"));
loader.setControllerFactory(controllerFactory);
Parent root = loader.load();
```

CAFEEFAC - June 9, 2017, 4:26 pm

Unfortunately, debugging is a lengthy process so we might have to exchange information and ideas like we are currently doing, a number of times before we figure this thing out. Hope you're okay with that.

First of all, you must make sure that you load with the FXMLLoader after you have set the

controller, just like it says in that bit of information you shared with me.

Here is an idea for something you could try:

First of all, fix it so that it loads the FXML file after the Controller2 has been set in the Model2 class.

Afterwards, remove the Controller attribute that you put inside the Controller1.fxml and make sure that the Controller1 is set in the Main class before it is loaded. Don't use the FXMLLoader.load method, keep it an instance like the way you have it right now. So, remove the controller from the FXML Controller1 file, set the Controller in the Main class, and then load it. See if that works.

black_lord - June 12, 2017, 8:49 am

Quote:Unfortunately, debugging is a lengthy process so we might have to exchange information and ideas like we are currently doing, a number of times before we figure this thing out. Hope you're okay with that.

That's totally fine by me. I'd like to thank you for your support.

I've double check everything you wrote and did some small modifications. Now if I run doWork() method from btnStartClick event, code in new Thread finally works! It's really strange because IDE still screams that there is no controller specified for top level element and makes my onAction="#btnStartClick" red, but when I run it, it works without any errors.

So it seems like some issue with compiler.

And now the funny part. When I call

```
Code public void btnStartClick(ActionEvent actionEvent) {  
    new Model2(loader);  
}stage2 doesn't show end gives me an error javafx.fxml.LoadException: No controller specified.
```

So it seems that I already solve this error once on stage1 with your help but I don't know how.

So, here are my classes now.

Main:

```
Code public class Main extends Application {  
    public static Stage myPrimaryStage;  
    public FXMLLoader loader;  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```

@Override

```
public void start(Stage primaryStage) throws Exception{  
    loader = new FXMLLoader(getClass().getResource("../view/Controller1.fxml"));  
    loader.setController(new Controller1(loader));  
    Parent root = loader.load();  
    myPrimaryStage = primaryStage;  
    myPrimaryStage.setTitle("MVC");  
    myPrimaryStage.setScene(new Scene(root));  
    myPrimaryStage.show();  
}  
}
```

Model2:

Code public class Model2

```
{
    Stage stage;
    FXMLLoader m2loader;

    public Model2(FXMLLoader loaderObject){
        loaderObject.setController(new Controller2(loaderObject));
        try {
            m2loader = FXMLLoader.load(getClass().getResource("../view/Controller2.fxml"));
            m2loader.setController(new Controller2(loaderObject));
            Parent root = m2loader.load();
            stage = new Stage();
            stage.setTitle("Run new Thread");
            stage.initModality(Modality.APPLICATION_MODAL);
            stage.setResizable(false);
            stage.setScene(new Scene(root,224,67));
            stage.show();
        }catch (Exception e){
            System.err.println(e);
        }
    }
}
```

Controller1:

Code public class Controller1 {

```
    public Label lblState;
```

```
    //Background Thread
```

```
    private Service<Void> backgroundThread;
```

```
    FXMLLoader loader;
```

```
    public Controller1 (FXMLLoader loaderObject){
```

```
        this.loader=loaderObject;
```

```
    }
```

```
    public void btnStartClick(ActionEvent actionEvent) {
```

```
        new Model2(loader);
```

```
    }
```

```
    public void doWork(){ ... };
```

```
        backgroundThread.setOnSucceeded(new EventHandler<WorkerStateEvent>() {...});
```

```
        lblState.textProperty().bind(backgroundThread.messageProperty());
```

```
        backgroundThread.restart();
```

```
    }
```

```
    public void btnStartFromNewThreadClick(ActionEvent actionEvent) {
```

```
        doWork(); //Works fine when called from here
```

```
    }
```

```
    public void btnCloseClick(ActionEvent actionEvent) {
```

```
        Main.myPrimaryStage.close();
```

```
    }
```

```

}
Contorller2:
Code public class Controller2 {

    private FXMLLoader loader;
    @FXML public TextField txtPortNumber;
    @FXML public Button btnStart;

    public Controller2(FXMLLoader loaderObject){
        this.loader = loaderObject;
    }

    @FXML
    public void btnStartClick(ActionEvent actionEvent){
        //Invoke new thread
        Controller1 obj = loader.getController();
        obj.doWork();
        //Close stage after invoking new thread
        Stage stage = (Stage) btnStart.getScene().getWindow();
        stage.close();
    }
}

```

fx:controller="controll.Controller1" and fx:controller="controll.Controller2" are not set in fxml files.

CAFEEFAC - June 12, 2017, 10:44 am

Take a look at your Model2 class.

One thing that's really important to remember is that when you want to set a controller without specifying it in the FXML file, you must create an instance of FXMLLoader and set the controller before you load it. But what you have done is use the FXMLLoader's static load method and this will load it before you have set any controller.

```

Code public Model2(FXMLLoader loaderObject){
    loaderObject.setController(new Controller2(loaderObject)); // This is not correct, this sets a new
    controller to the Controller1 FXML. Remove it. :)
    try {
        m2loader = FXMLLoader.load(getClass().getResource("../view/Controller2.fxml")); // This is
        where the problem lies. You're using the static load method, do you see?
        m2loader.setController(new Controller2(loaderObject));
        Parent root = m2loader.load(); // You can't do this, you already loaded it. Get it? ;)
        stage = new Stage();
        stage.setTitle("Run new Thread");
        stage.initModality(Modality.APPLICATION_MODAL);
        stage.setResizable(false);
        stage.setScene(new Scene(root,224,67));
        stage.show();
    }catch (Exception e){
        System.err.println(e);
    }
}
}

```

So, instead of doing:

Code m2loader = FXMLLoader.load(getClass().getResource("../view/Controller2.fxml"));
Do this:
Code m2loader = new FXMLLoader(getClass().getResource("../view/Controller2.fxml"));
That should work!

black_lord - June 13, 2017, 10:07 am

It works! ? That was a thing I didn't understand and I'll never forget it. Thank you very much for teaching me this concept and helping me to solve this problem. I hope I didn't cost you too much time. Have a great day ??

CAFEEFAC - June 13, 2017, 10:12 am

(Jun 13, 2017 10:07 AM)black_lord Wrote: [\[CITE\]](#) It works! ? That was a thing I didn't understand and I'll never forget it. Thank you very much for teaching me this concept and helping me to solve this problem. I hope I didn't cost you too much time. Have a great day ??

The great part of teaching others is that the more you teach someone a subject, the more ingrained it becomes in your own mind. :)

I'm glad it works. Now work on improving it. ;)

MikeNdu87 - June 14, 2017, 3:48 am

so hard to understand bt you niga got my atention i gona check this read again?