

JavaFX MVC

Sadržaj

Osnove MVC arhitekture

Controller klasa

Povezivanje *Controller* klase i FXML datoteke

Anotacija @FXML

Povezivanje Java metoda s elementima
grafičkog sučelja

JavaFX zbirke

Primjer korištenja ObservableList zbirke

Povezivanje stupaca tablice s varijablama u
objektima

Povezivanje akcija s izbornikom

Osnove MVC arhitekture

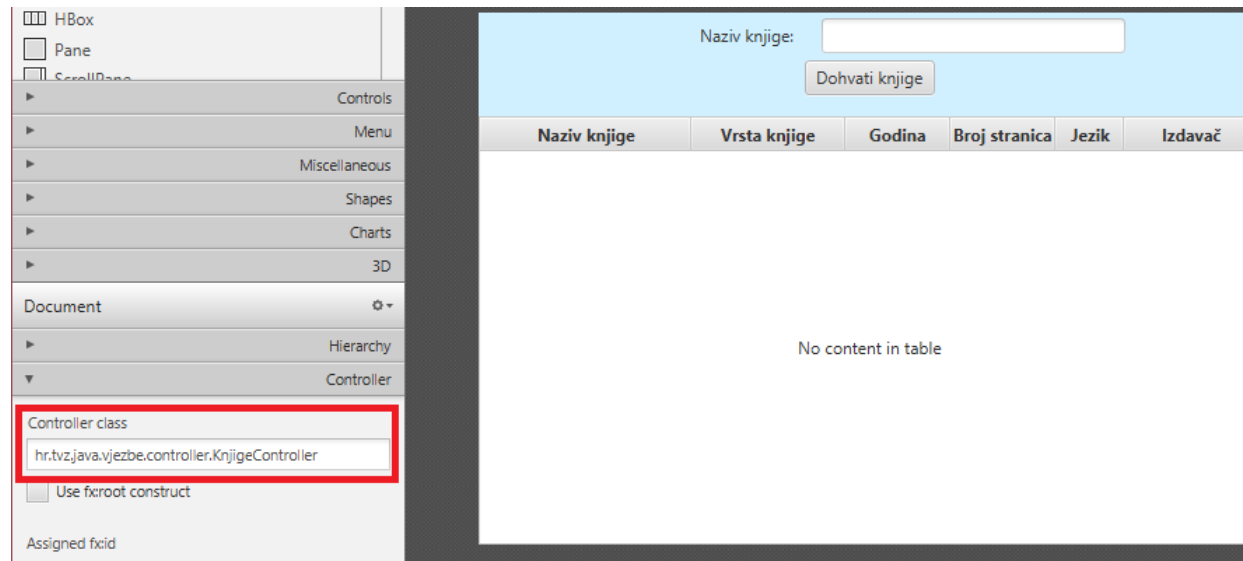
- JavaFX se temelji na MVC (engl. *Model View Controller*) arhitekturi
- Odvajanje slojeva aplikacije koji predstavljaju podatke (*Model*), grafičko sučelje (*View*) i poslovnu logiku aplikacije (*Controller*)
- Model podataka se obično sastoji od klasa koje predstavljaju domenske podatke
- Grafičko sučelje se temelji na FXML datoteci koja definira izgled grafičkog sučelja ili klasama pomoću kojih se dizajnira grafičko sučelje kroz naredbe u Java kodu
- Poslovna logika se implementira unutar klasa koje povezuju elemente grafičkog sučelja (npr. gumba) s metodama koje se moraju izvoditi tijekom interakcije korisnika s tim elementima (npr. pritisak gumba)
- Za ostvarivanje veze između grafičkog sučelja i poslovne logike, elementima grafičkog sučelja je potrebno dodijeliti jedinstvene identifikatore

Controller klasa

- Omogućava povezivanje elemenata grafičkog sučelja s Java programskim kodom koji predstavlja poslovnu logiku
- Sadrži varijable koje po tipu i nazivu odgovaraju elementima grafičkog sučelja, kako bi se mogle koristiti njihove značajke i atributi
- Naziv *Controller* klase mora biti povezan s grafičkim sučeljem korištenjem FXML konfiguracije
- Preporuča se da svaki ekran osim pripadajuće FXML datoteke ima i svoju *Controller* klasu
- Svaka *Controller* klasa koja inicijalizira izgled ekrana koji prikazuje treba imati svoju „initialize” metodu označenu s „@FXML” anotacijom koja sadrži svu logiku potrebnu za inicijalizaciju osnovnih postavki elemenata na grafičkom sučelju

Povezivanje *Controller* klase i FXML datoteke

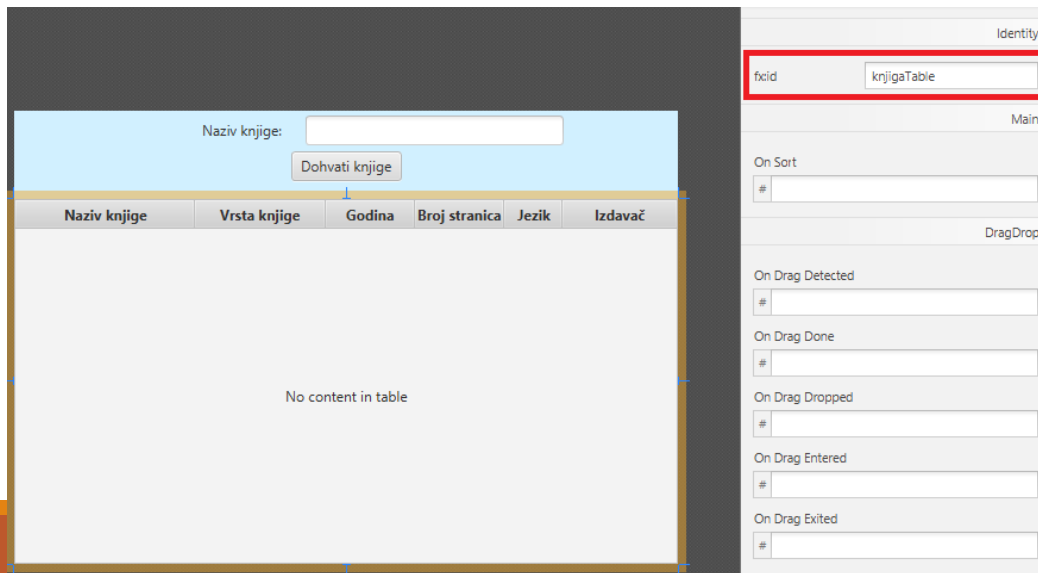
- Glavni grafički element ekrana moguće je povezati s pripadajućom *Controller* klasom korištenjem opcije u „Scene Builderu” ili unutar FXML datoteke:



```
<BorderPane minWidth="-1.0" prefHeight="380.0" prefWidth="558.0"
xmlns:fx="http://javafx.com/fxml/1" xmlns="http://javafx.com/javafx/2.2"
fx:controller="hr.tvz.java.vjezbe.controller.KnjigeController">
```

Anotacija @FXML

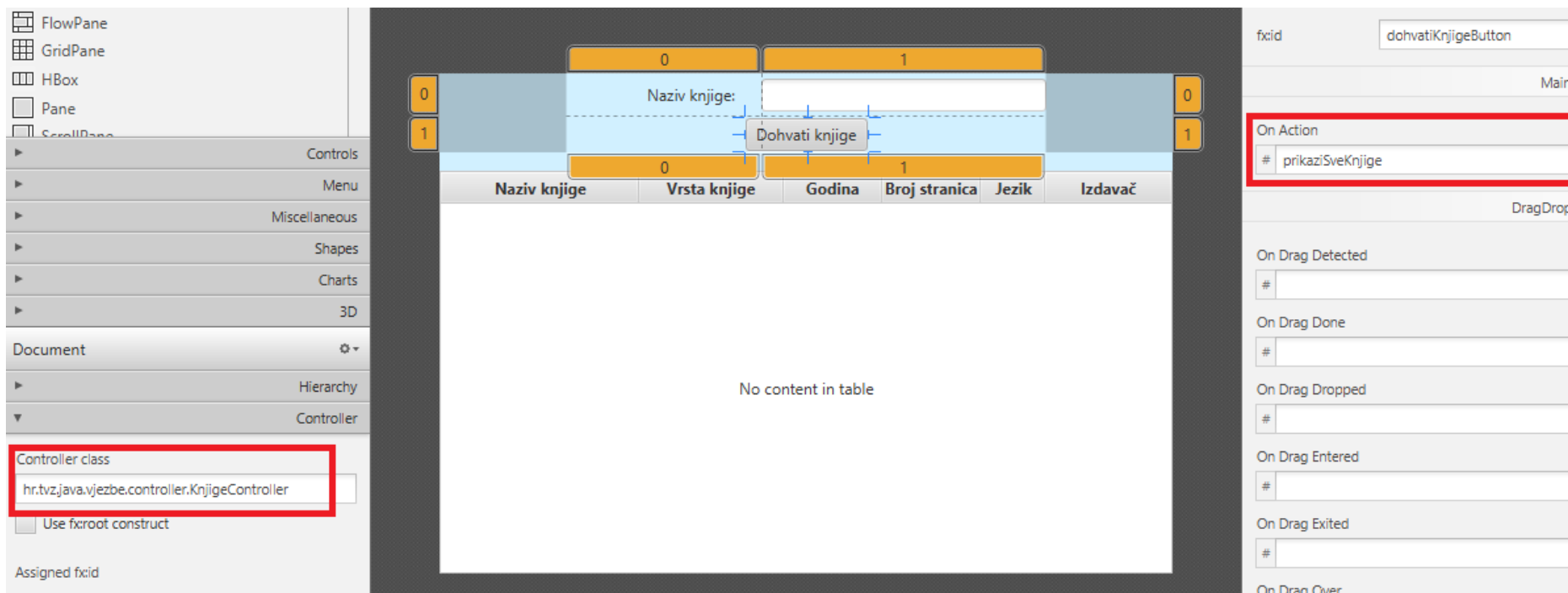
- Služi za označavanje elemenata u programskom jeziku Java (*Controller* klasi) koji se koriste unutar FXML datoteke
- Može se koristiti kod varijabli koje predstavljaju grafičke elemente ili metoda koje se pozivaju prilikom interakcije s nekim od grafičkih elemenata
- Na primjer, ako se elementu „TableView” unutar FXML datoteke postavi za „fx:id” identifikator „knjigaTable”, taj element je moguće povezati s varijablom unutar Java klase korištenjem anotacije „@FXML” (nazivi se moraju podudarati):



```
public class KnjigeController {  
  
    @FXML  
    private TableView<Knjiga> knjigaTable;  
  
}
```

Povezivanje Java metoda s elementima grafičkog sučelja

- Kako bi se npr. gumb na grafičkom sučelju povezao s odgovarajućom metodom unutar *Controllera*, nakon povezivanja ekrana s *Controller* klasom potrebno je definirati „OnAction” parametar koji mora sadržavati naziv metode koja se mora pozvati:



Povezivanje Java metoda s elementima grafičkog sučelja

- Definirani *Controller* mora sadržavati metodu koja sadrži logiku za dohvaćanje knjiga i postavljanje podataka u odgovarajuće objekte

```
public class KnjigeController {  
    ...  
    public void prikaziSveKnjige() {  
        ...  
    }  
}
```

- Ako je potrebno prikazivati zbirku podataka na grafičkom sučelju, JavaFX podržava „Observable” tipove zbirke koje omogućavaju izravno praćenje promjena na grafičkom sučelju

JavaFX zbirke

- Za potrebe rada s podacima na grafičkom sučelju JavaFX koristi sljedeće zbirke:
 - **ObservableList**: predstavlja listu koja omogućava praćenje promjena na ekranu korištenjem *Listenera*
 - **ObservableMap**: predstavlja mapu koja omogućava praćenje promjena na ekranu korištenjem *Listenera*
 - **ListChangeListener** i **MapChangeListener**: predstavljaju implementacije *Listener* klasa koje služe za primanje notifikacija koje generira objekt klase „ObservableList” i „ObservableMap”
- Osim toga postoje još i sljedeće korisne klase:
 - **FXCollections**: sadrži istovjetne statičke metode onima u klasi „java.util.Collections”
 - **ListChangeListener.Change**: instance koje predstavljaju promjene nad ObservableList zbirkom
 - **MapChangeListener.Change**: instance koje predstavljaju promjene nad ObservableMap zbirkom

Primjer korištenja ObservableList zbirke

- Objekt klase **ObservableList** moguće je kreirati iz „java.util.List” zbirke na sljedeći način:

```
ObservableList<Knjiga> listaKnjiga = FXCollections  
.observableArrayList(filtriraneKnjige);
```

- Ako je tu zbirku potrebno povezati s „TableView” komponentom, to je moguće obaviti na sljedeći način:

```
knjigaTable.setItems(listaKnjiga);
```

Povezivanje stupaca tablice s varijablama u objektima

- JavaFX komponenta „TableView” za tablično prikazivanje podataka sastoji se od elemenata „TableColumn” koji služe za prikazivanje podataka u pojedinim stupcima



Povezivanje stupaca tablice s varijablama u objektima

- Za povezivanje određenog stupca tablice s nazivom varijable (npr. „naziv”) u domenskoj klasi potrebno je unutar „initialize” metode *Controller* klase koristiti metodu „setCellValueFactory” i klasu „PropertyValueFactory”:

```
@FXML
```

```
private TableColumn<Knjiga, String> nazivKnjigeColumn;
```

```
...
```

```
@FXML
```

```
public void initialize() {
```

```
    nazivKnjigeColumn
```

```
        .setCellValueFactory(new PropertyValueFactory<Knjiga, String>("naziv"));
```

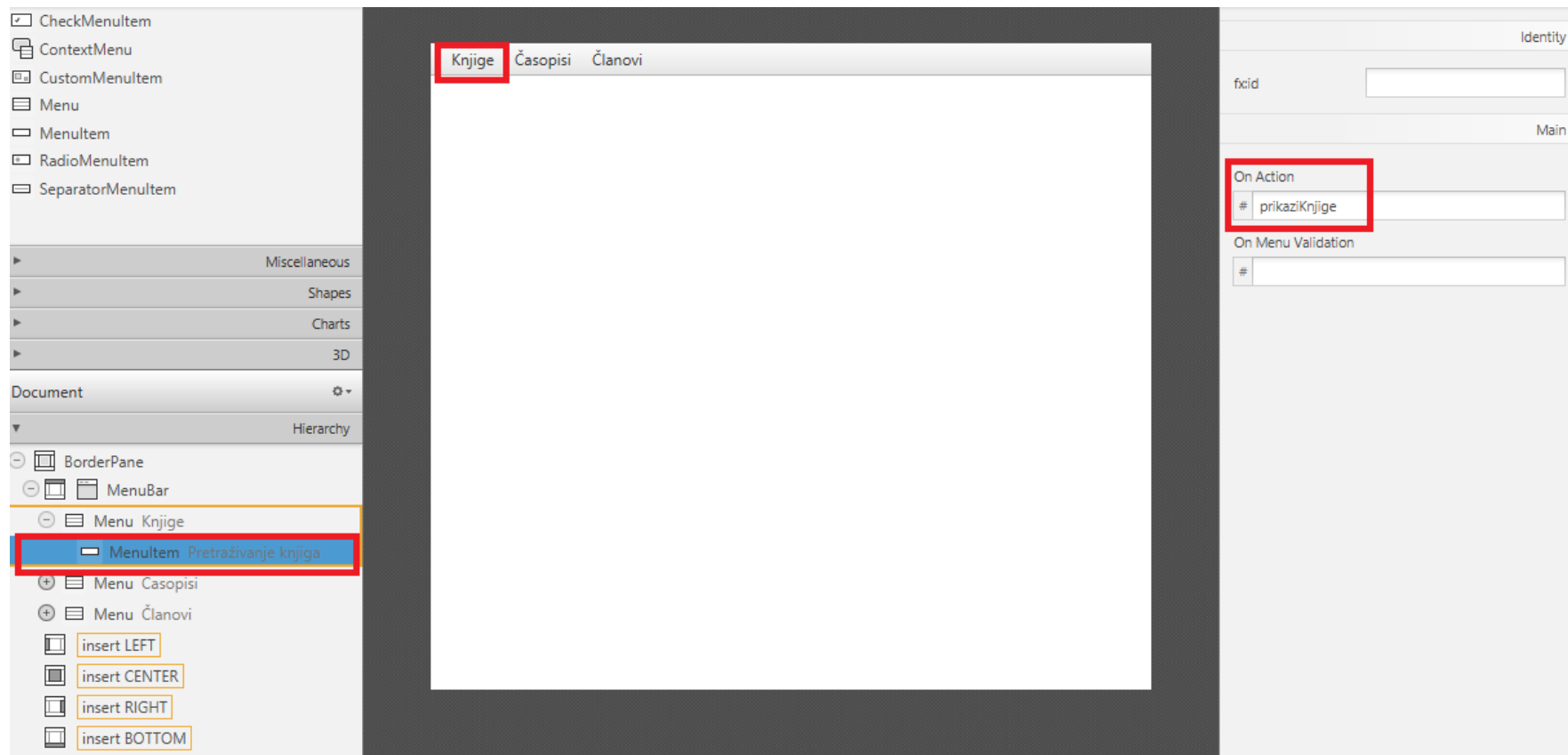
```
...
```

Povezivanje akcija s izbornikom

- Izbornik za odabir opcija na grafičkom sučelju sastoji se od glavnog „Menu” elementa s pripadajućim „MenuItem” elementima
- Svaki „MenuItem” element može se povezati s pripadajućom akcijom unutar *Controller* klase
- Preporuča se da se nakon odabira opcije iz izbornika kreira i prikaže ekran koji je definiran u zasebnoj FXML datoteci na sljedeći način:

```
public void prikaziKnjige() {  
    try {  
        BorderPane knjigaPane =  
            FXMLLoader.load(JavaFXGlavna.class.getResource("../javafx/knjige.fxml"));  
        JavaFXGlavna.setCenterPane(knjigaPane);  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Povezivanje akcija s izbornikom



Pitanja?
