

# Oracle ADF Bindings - from beginner to master !

Moje ime:

**Diana Mošnja**

Ruže Petrović 12, 52100 Pula  
tel: (052) 386 384 fax: (052) 386 407  
e-mail: [info@istrattech.hr](mailto:info@istrattech.hr)

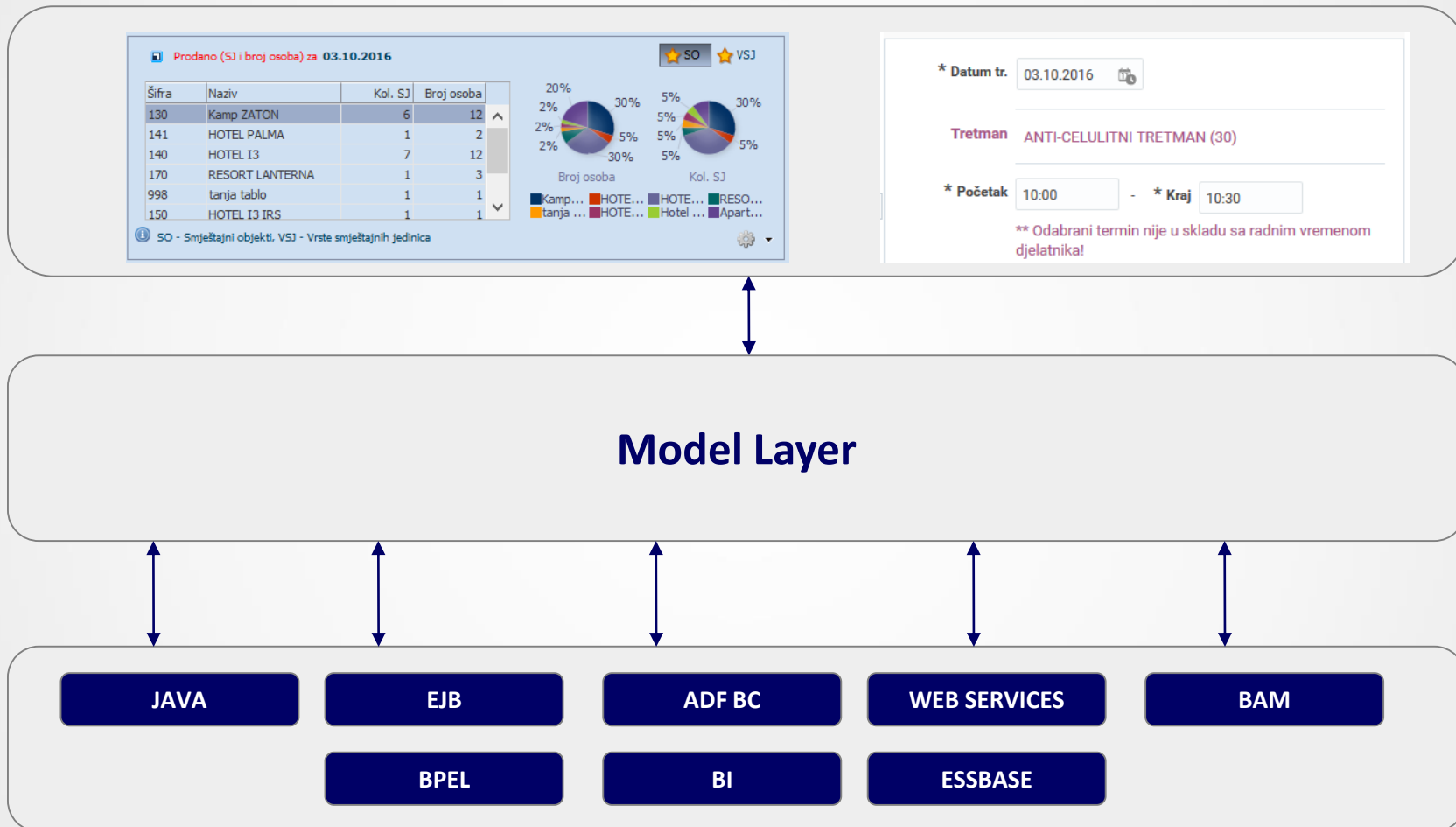


## \* Osnove ADF Bindings-a

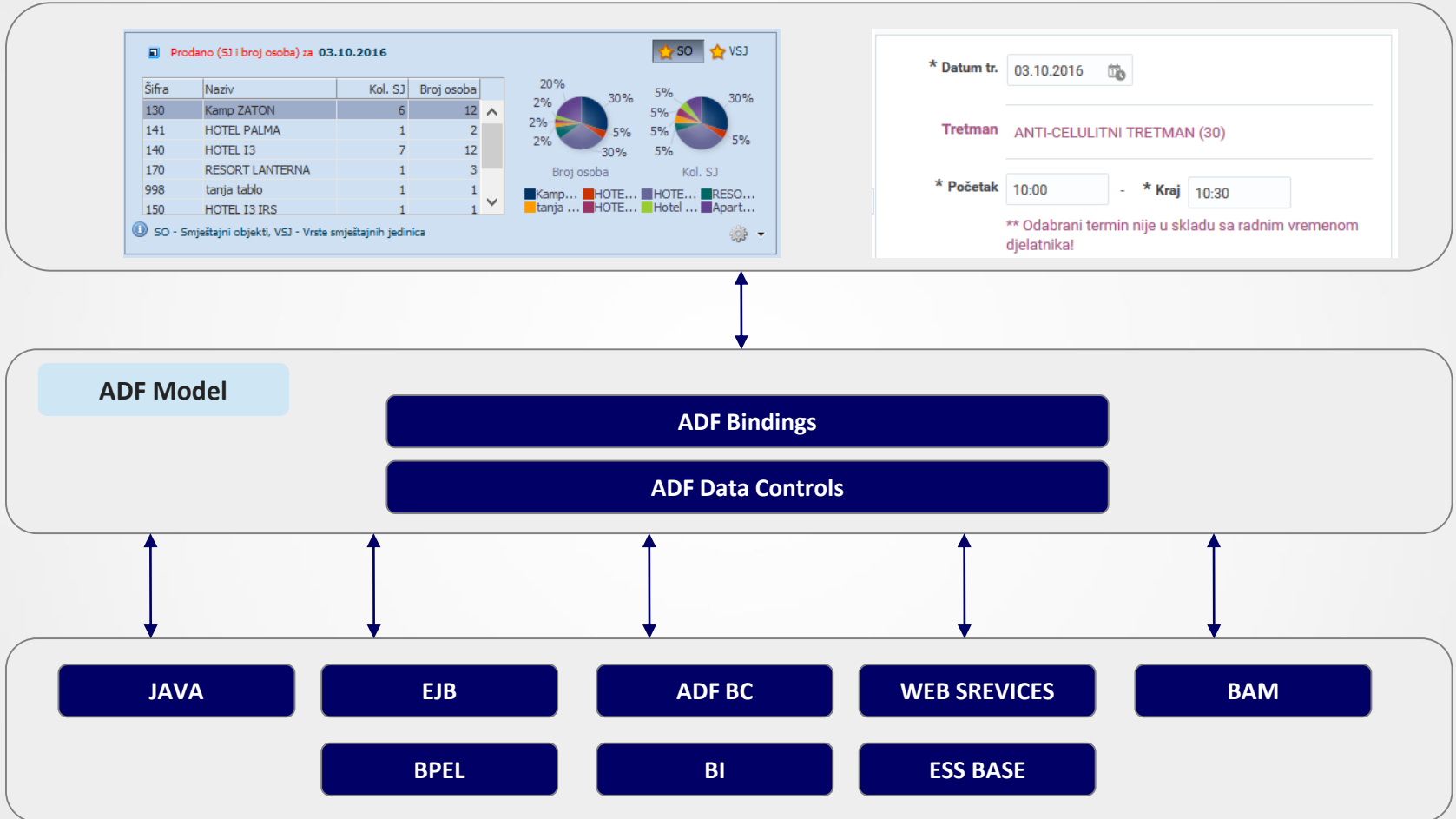
Za uspješan rad sa Oracle ADF alatom, potrebno je potpuno razumijevanje svih slojeva MVC arhitekture. ADF Model sloj primjenjuje koncepte koji omogućuju razdvajanje korisničkog sučelja od implementacije poslovnih pravila.

**ADF Bindings sastavni su dio ADF-a Modela.**

## \* Osnove ADF Bindings-a



## \* Osnove ADF Bindings-a

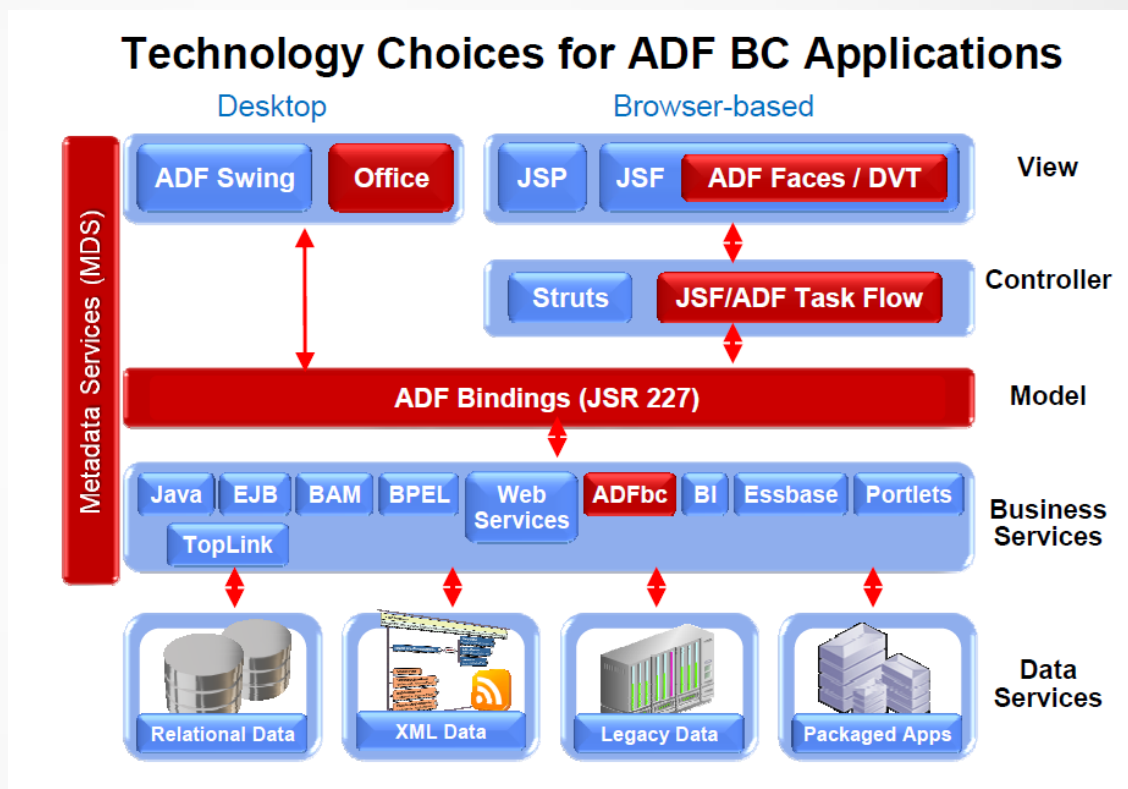


## \* Osnove ADF Bindings-a

Abstrakcija modela podataka preko ADF Modela omogućuje deklarativni pristup razvoju web aplikacije.

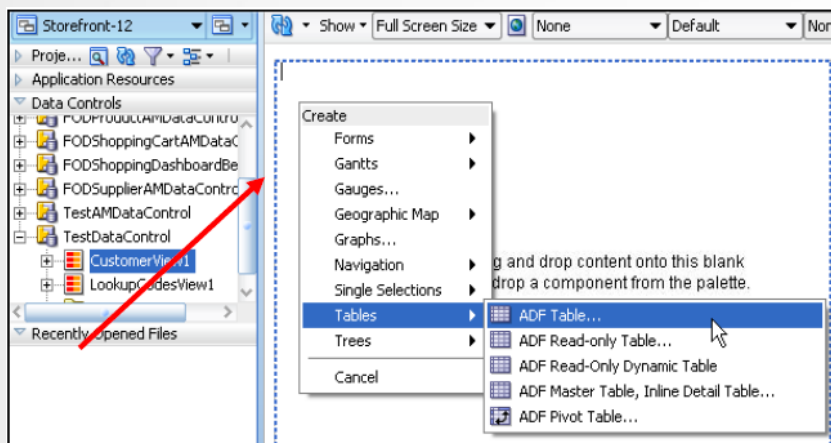
Uobičajene zadatke poslovne aplikacije moguće je riješiti "**out-of-the-box**" pristupom.

## \* Osnove ADF Bindings-a



## \* Temeljni izrazi – ADF Bindings

### Data control



Data control je java objekt u ADF Model sloju koji djeluje kao dodatni sloj za abstrakciju poslovnih pravila. Umjesto čitanja objekata sa podacima i slanja podataka direktno u objekte sa podacima, aplikacija šalje sve upite preko data controle.

Prilikom kreiranja aplikacijskog modula, JDeveloper automatski kreira data control.



## \* Temeljni izrazi – ADF Bindings

### Page Data Binding Definition

Za svaku stranicu ili fragment koji sadrže ADF Bindings, JDeveloper formira odgovarajući **Page Definition datoteku**.

U runtimeu binding objekti, koji su definirani u Page Definition datoteci, instancirani su u binding containeru.



## \* Temeljni izrazi – ADF Bindings

### Page Data Binding Definition

**Rezervacija tretmana**

Klijent    Tretman    **Termin**    Napomena    Pregled

\* Datum tr. 03.10.2016

Tretman ANTI-CELULITNI TRETMAN (30)

\* Početak 10:00 - \* Kraj 10:30

\*\* Odabrani termin nije u skladu sa radnim vremenom djelatnika!

\* Terapeut DEJAN PRINCE

Fiksna rezervacija

\* Prostor SMOOTH ROOM

U redu  Odustani

Set - viewcontrollerBundle |

Rezervacija tretmana context

Train

Datum tr. #{...DatTretmana.inputValue}

Tretman #{...NazivTretmana.inputValue} (#{...TrajanjeTretmana.inputValue})

\* Početak #{...SatPocChr.inputValue} Kraj #{...SatZavChr.inputValue}

Terapeut #{...ZmpdjeId.inputValue}

Dodatni terapeut #{...ZmpdjeDodId.inputValue}

Fiksna rezervacija

Prostor #{...ZmpproslId.inputValue}

## \* Temeljni izrazi – ADF Bindings

### Page Data Binding Definition

Page Definition File: [pages/train/rezervacijeKadPageDef.xml](#)

Bindings and Executables    Contextual Events    Parameters

Bindings

- DatTretmana
- Zmpdjeld
- ZmpprosId
- SatPocChr
- SatZavChr
- Commit
- TretmanjeTretmana

Executables

- variables
- RezervacijeTretmanaVOIterator
- RezervacijeSchedulerVOIterator

Data Control

- WellnessAMDataControl
- AktivniDjelatniciPregled
- AostimaniKaseNewSin
- AsortimanNestandardni
- AsortimanTretmanaLC
- BojeGrupaTretmana1

```

<af:inputDate value="#{bindings.DatTretmana.inputValue}" label="#{bindings.DatTretmana.hints.label}"
  required="#{bindings.DatTretmana.hints.mandatory}"
  columns="#{bindings.DatTretmana.hints.displayWidth}"
  shortDesc="#{bindings.DatTretmana.hints.tooltip}" id="id1" autoSubmit="true">
  <f:validator binding="#{bindings.DatTretmana.validator}"/>
  <af:convertDateTime pattern="#{bindings.DatTretmana.format}" secondaryPattern="ddMMMyyyy"/>
</af:inputDate>
  
```

Data Controls

- RezervacijeTretmanaCheck
- RezervacijeTretmanaRefreshVo
- RezervacijeTretmanaVO
  - BoravakGosta
  - BrTelefonaDjelatnika
  - Cijena
  - DatTretmana
  - Datum**
  - DodatnaInfo
  - DodatnaInfo2
  - DodatnaInfo3

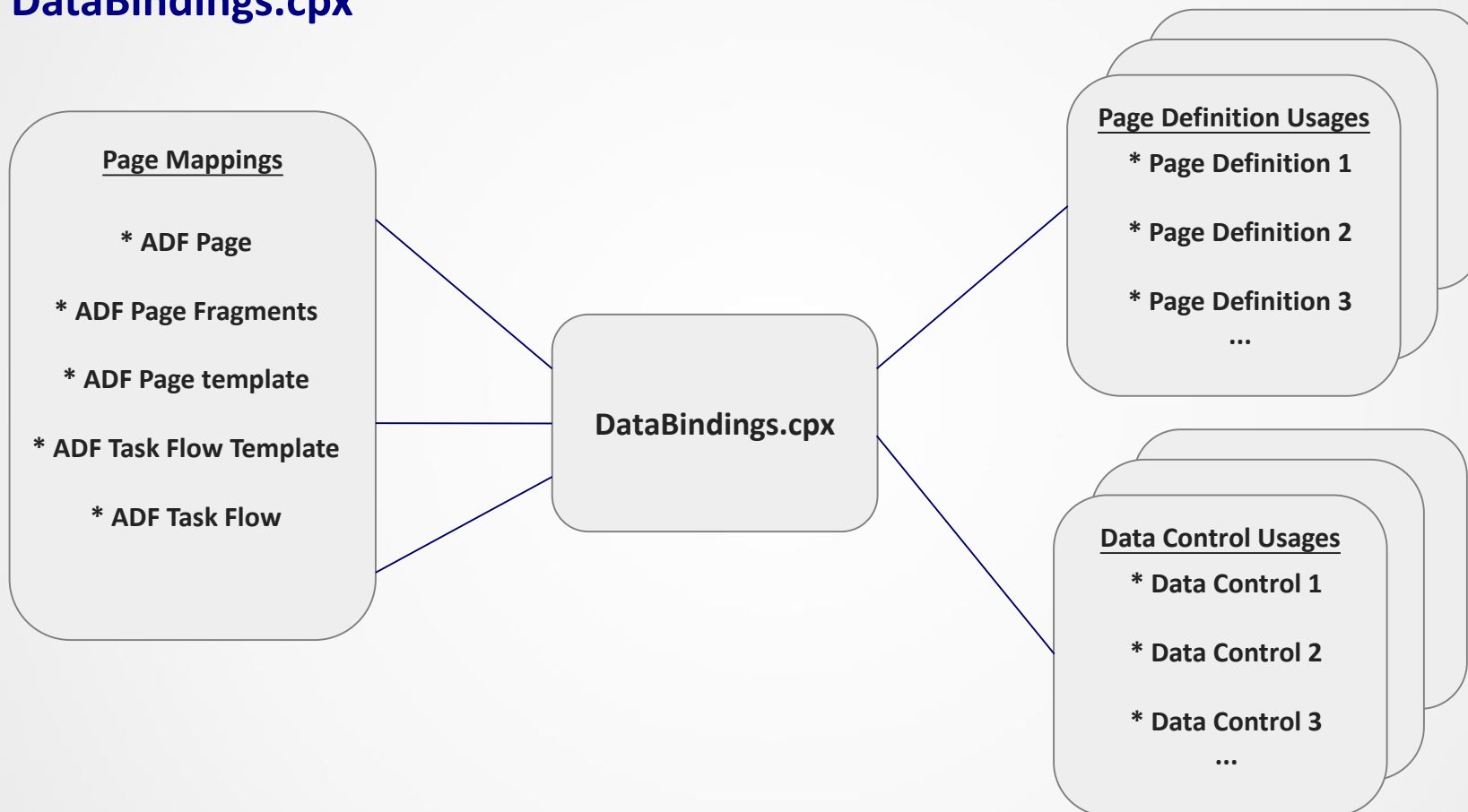
## Oracle ADF Bindings

### DataBindings.cpx

DataBindings.cpx je XML datoteka koja definira Binding Context za aplikaciju. Prilikom prvog bindanja na stranici, fragmentu ili taskflowu, JDeveloper automatski kreira datoteku DataBindings.cpx

## \* Temeljni izrazi – ADF Bindings

### DataBindings.cpx








## \* Temeljni izrazi – ADF Bindings


### Page Mappings

path	usageId
<a href="#">/LoginPage.jsf</a>	hr_iii_wellness_view_LoginPagePageDef
<a href="#">/pages/dashboard.jsf</a>	hr_iii_wellness_view_dashboardPageDef
<a href="#">/templates/basicTemplate.jsf</a>	hr_iii_wellness_view_basicTemplatePageDef
<a href="#">/pages/wellness_mp.jsf</a>	hr_iii_wellness_view_wellness_mpPageDef
<a href="#">/pages/fragments/wellness_mp.jsff</a>	hr_iii_wellness_view_wellness_mpPageDef1
<a href="#">/pages/fragments/browseGrupeDjelatnika.jsff</a>	hr_iii_wellness_view_browseDjelatniciPageDef

### Page Definition Usages

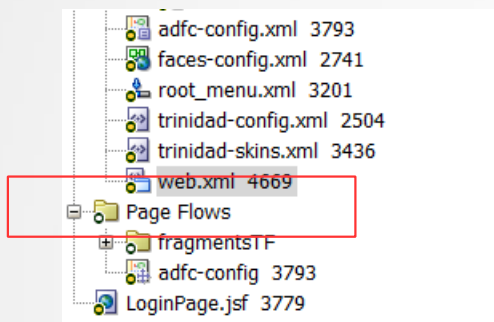
id	path
 hr_iii_wellness_view_LoginPagePageDef	<a href="#">hr.iii.wellness.view.pageDefs.LoginPagePageDef</a>
 hr_iii_wellness_view_dashboardPageDef	<a href="#">pages.dashboardPageDef</a>
 hr_iii_wellness_view_basicTemplatePageDef	<a href="#">templates.basicTemplatePageDef</a>
 hr_iii_wellness_view_wellness_mpPageDef	<a href="#">pages.wellness_mpPageDef</a>
 hr_iii_wellness_view_wellness_mpPageDef1	<a href="#">pages.fragments.wellness_mpPageDef</a>

### Data Control Usages

id
 WellnessAMDataControl

## \* Temeljni izrazi – ADF Bindings

### Web.xml



U descriptor datoteci **web.xml** definiran je adfBindings filter i moguće ga je kustomizirati.

ADF web aplikacije koriste ADF Binding filter za predprocesiranje HTTP requesta.

```

<filter>
  <filter-name>adfBindings</filter-name>
  <!--
  <filter-class>oracle.adf.model.servlet.ADFBindingFilter</filter-class>
-->
  <filter-class>hr.iii.wellness.login.DynamicJDBCBindingFilter</filter-class>
</filter>

```

oracle.adf.model.servlet.ADFBindingFilter

## \* Temeljni izrazi – ADF Bindings

### Tipovi bindingsa

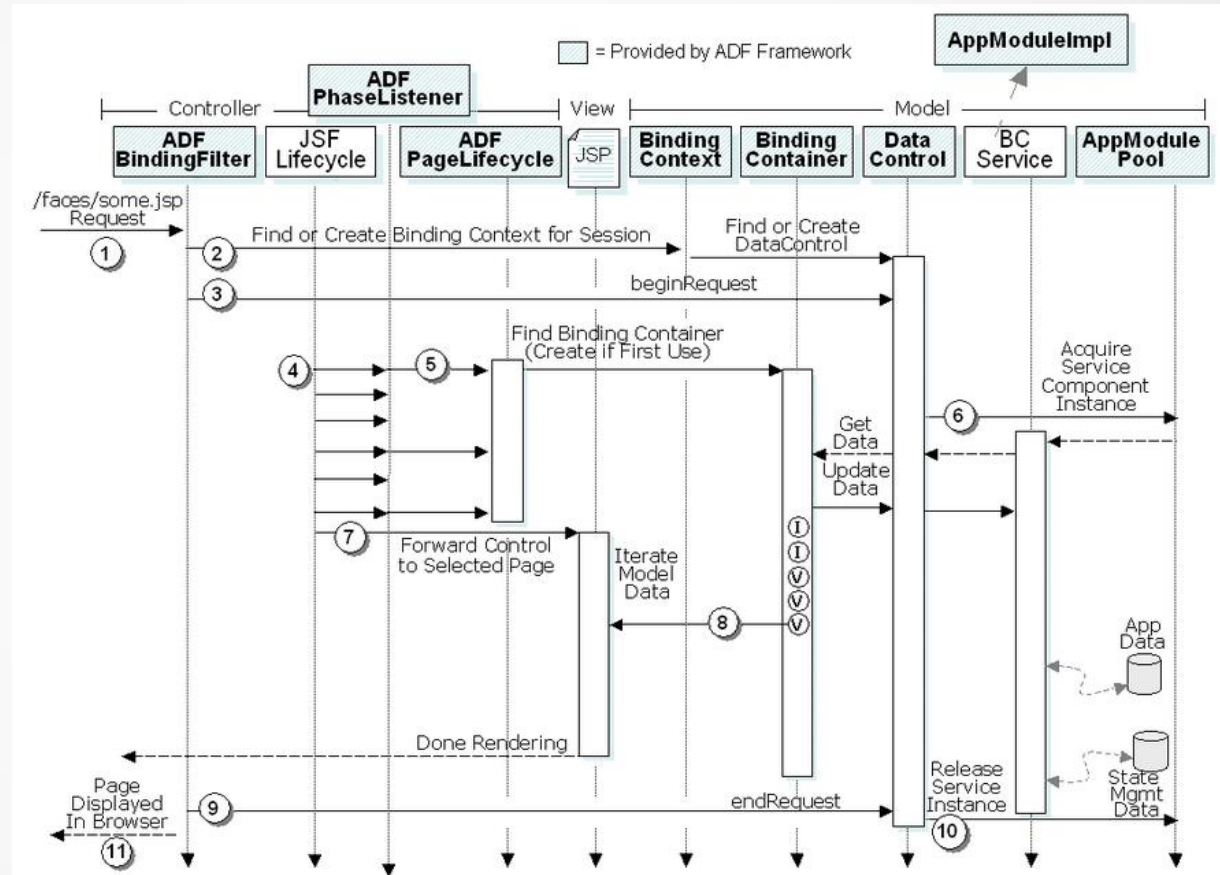
<u>Iterator binding</u>	
<ul style="list-style-type: none"><li>• accessor iterator</li></ul>	U master-detail relaciji, iterira kroz detail objekt definiran accessorom.
<ul style="list-style-type: none"><li>• iterator</li></ul>	Iterira kroz kolekciju.
<ul style="list-style-type: none"><li>• method iterator</li></ul>	Iterira kroz rezultat metode (row set).
<u>Value bindings</u>	Za svaku UI komponentu koju bindamo (attribute binding, list binding).
<u>Action binding</u>	Omogućuje dostup do metode ili operacije definirane na poslovnom objektu.



## \* ADF Bindings u ADF Lifecycle

Prilikom novog zahtjeva za prikazom stranice, aplikacija izvodi ADF Faces lifecycle koji je proširena verzija standardnog JSF lifecycle. Prošireni JSF lifecycle upravlja spremanjem vrijednosti sa stranice, validacijom, navigacijom, prikazom komponenti na stranici i spremanjem odnosno povratom prošlog stanja.

## \* ADF Bindings u ADF Lifecycle



## \* ADF Bindings u ADF Lifecycle

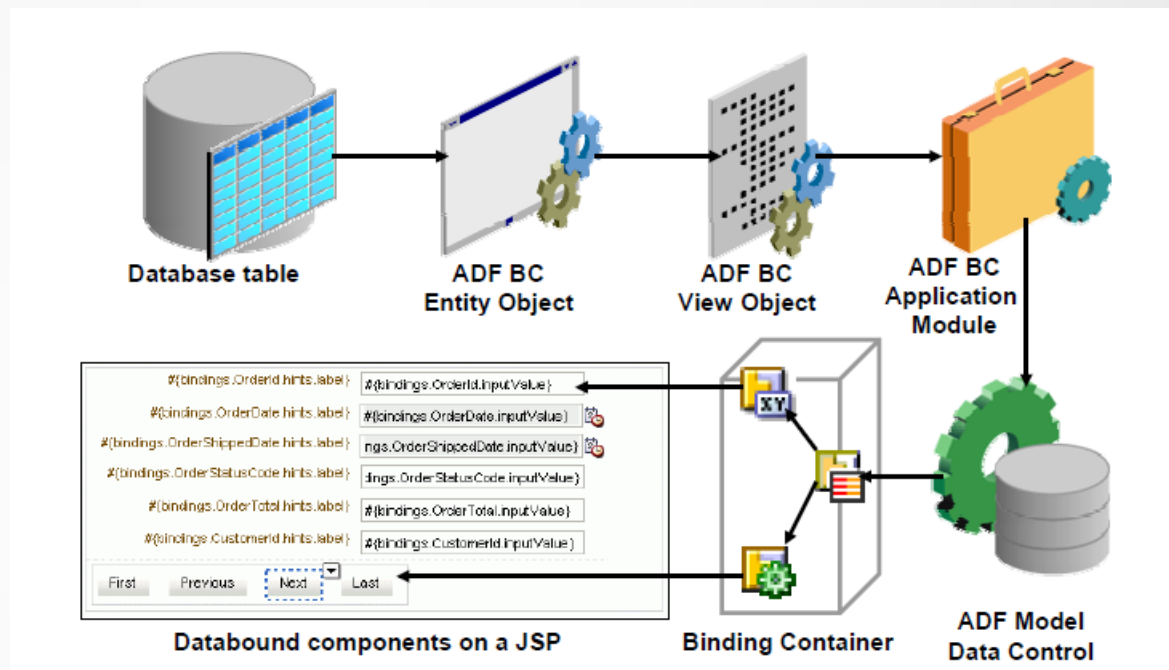
1	Sa klijenta na server upućen je web zahtjev ( <b>web request</b> ) za prikaz stranice faces/some.jsp.
2	<b>ADF BindingFilter</b> objekt traži <b>ADF Binding context</b> u HTTP sesiji i ako još ne postoji, inicijalizira ga po prvi puta. Također kreira instance svih potrebnih <b>data controla</b> .
3	ADF BindingFilter izvršava <b>beginRequest()</b> metodu na dana controli koja sudjeluje u requestu.
4	S obzirom da je ADF Binding context inicijaliziran kontrola se vraća faces servletu koji je zadužen za procesiranje JSF Lifesycle faza.
5	ADF PhaseListener objekt kreira ADF PageLifecycle objekt kako bi upravljao svakim zahtjevom i delegirao prikladne before ili after metode. Ukoliko <b>Binding Container</b> stranice nije korišten ranije, u ovom se djelu uspješno instancira.
6	Po prvi puta, u toku zahtjeva, referencira se <b>aplikacijski modul</b> . Instanca aplikacijskog modula dobija se iz am pool-a.
7	JSF lifecycle objekt prosljeđuje kontrolu stranici koju treba renderirati.
8	UI komponente na stranici dohvaćaju <b>value bindings</b> i <b>iterator bindings</b> u binding containeru stranice i renderira se formatirani prikaz koji se pojavljuje u pregledniku.
9	ADF BindingFilter izvršava <b>endRequest()</b> metodu na svakoj dana controli.
10	Aplikacijski modul koristi endRequest notifikaciju kako bi <b>otпустиo instancu aplikacijskog modula</b> natrag u am pool.
11	<b>Korisnik vidi zatraženu stranicu u pregledniku.</b>

## \* Java kod uporabom ADF modela

Veliki dio onoga što moramo postići u ADF-u prilikom izrade nove aplikacije moguće je napraviti deklarativno.

No, postoje isto tako mnogi zahtjevi kada određene funkcije moramo pozvati ručno iz java koda.

## \* Java kod uporabom ADF modela



```
public BindingContainer getBindings() {
    return BindingContext.getCurrent().getCurrentBindingsEntry();
}
```

## \* Java kod uporabom ADF modela

### Attribute Binding

```
AttributeBinding attr = (AttributeBinding) getBindings().getControlBinding("RezId");  
Integer rezId = (Integer) attr.getInputValue();
```

The screenshot displays three panels from the ADF IDE:

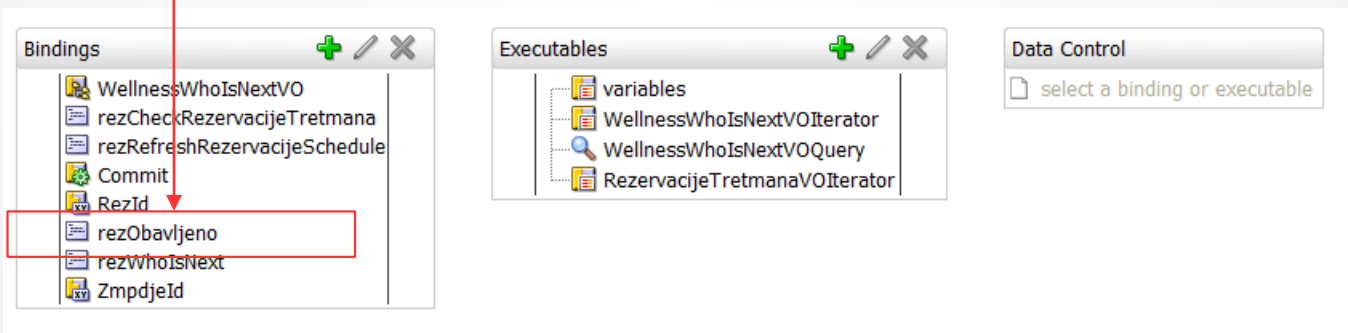
- Bindings:** A list of bindings including WellnessWhoIsNextVO, rezCheckRezervacijeTretmana, rezRefreshRezervacijeSchedule, Commit, RezId (highlighted with a red box), rezObavljeno, rezWhoIsNext, and ZmpdjeId.
- Executables:** A list of executables including variables, WellnessWhoIsNextVOIterator, WellnessWhoIsNextVOQuery, and RezervacijeTretmanaVOIterator.
- Data Control:** A panel with a search icon and the text "select a binding or executable".

A red line connects the "RezId" binding in the Bindings panel to the corresponding "RezId" string in the Java code above.

## \* Java kod uporabom ADF modela

### Operation Binding

```
OperationBinding operation = getBindings().getOperationBinding("rezObavljeno");  
if (operation != null) {  
    Map paramMap = operation.getParamsMap();  
    paramMap.put("rezId", rezId);  
    operation.execute();  
}
```



The screenshot displays three panels from the ADF IDE:

- Bindings:** A list of bindings including WellnessWhoIsNextVO, rezCheckRezervacijeTretmana, rezRefreshRezervacijeSchedule, Commit, RezId, rezObavljeno (highlighted with a red box), rezWhoIsNext, and ZmpdjeId. A red arrow points from the code above to this binding.
- Executables:** A list of executables including variables, WellnessWhoIsNextVOIterator, WellnessWhoIsNextVOQuery, and RezervacijeTretmanaVOIterator.
- Data Control:** A panel with a search icon and the text "select a binding or executable".



## \* Java kod uporabom ADF modela

### Iterator Binding

```
DCIteratorBinding iterator = (DCIteratorBinding) getBindings().get("WellnessDayCalcParamsVOIterator");  
  
if (iterator != null) {  
    Row currentRow = iterator.getCurrentRow();  
    if (currentRow != null) {  
        oracle.jbo.domain.Date l_dbDateStart = (oracle.jbo.domain.Date) currentRow.getAttribute("OdDatuma");  
    }  
}
```

The screenshot shows three panels from the ADF IDE:

- Bindings:** Lists various bindings including WellnessWhoIsNextVO, rezCheckRezervacijeTretmana, rezRefreshRezervacijeSchedule, Commit, RezId, rezObavljeno, rezWhoIsNext, and ZmpdjeId.
- Executables:** Lists executables including variables, WellnessWhoIsNextVOIterator (highlighted with a red box), WellnessWhoIsNextVOQuery, and RezervacijeTretmanaVOIterator. A red arrow points from the code above to this entry.
- Data Control:** Contains a dropdown menu with the text "select a binding or executable".

## \* Java kod uporabom ADF modela

### Table iterator

Osvježi Tko?

Šifra	Naziv	Obavezan OIB	Napomena
KOZ	Kozmetičar	✓	
MAS	Masažeri		Napomena na masažerima dddd
PED	Pediker		
TER	Medicinski terap...		00d asd
ARO	Aroma-terapeuti	✓	
SS	ss		ss
XX	xx		xx
VIZ	Vizažisti	✓	Grupa od 10 ljudi.

```
public DCIteratorBinding getIteratorfromTable() {  
    RichTable table1 = this.getTableData();  
    CollectionModel tableModel = (CollectionModel)table1.getValue();  
    JUCtrlHierBinding treeBinding = (JUCtrlHierBinding)tableModel.getWrappedData();  
    DCIteratorBinding iter = treeBinding.getDCIteratorBinding();  
    return iter;  
}
```

## \* Oracle ADF Bindings - table

Šifra	Naziv
413033	SALATA OD TJEST. S PILETINOM
413021	SALATA OD PLODOVA MORA
413034	HOBOTNICA S LIM. I ČEŠNJAKOM
404021	TRIS SA ŠKAMPIMA
404022	NJOKI-ČETIRI VRSTE SIRA
405038	GRDOBINA U BIJELOM VINU
405036	SRDELA NA ŽARU
403040	SENDVIČ-SELJAČKI
403044	SENDVICH FRESCO
403045	SENDVICH S ROAST-BEEFOM
660013	EIS CAFFEE
620017	KORNAT 0,2-VINOPLOD
621018	MALVAZIJA 0,75-MATOŠEVIĆ
621019	MALVAZIJA 0,75-PIL.ARM.KOZLOVIC
621020	MERLOT 0,2-AGROLAGUNA
621021	MERLOT 0,75 FRAJONA
621022	PLAVAC-0,75-PLOD
660012	DUPLI MACHIA
700000	ŠIBICE
700001	BENSTON MEK
700002	CROATIA
700003	DORSET
700004	FILTER 160 ME
700005	KOLUMBO
700007	MARLBORO

```
private java.util.List<ProizvodiBasic> data;
```

```
public class ProizvodiBasic {
```

```
private String sifra;
```

```
private String sifProizvodaKasa;
```

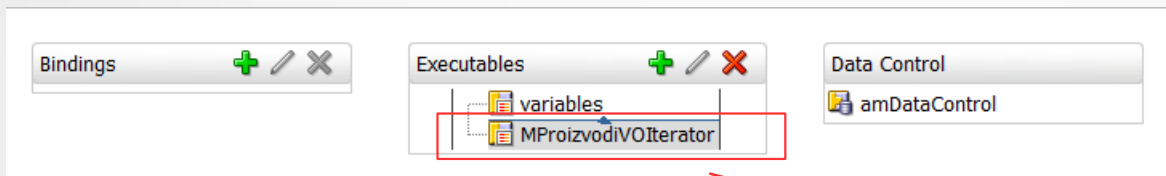
```
private String naziv;
```

```
private String jm;
```

```
private BigDecimal zadNabavnaCijena;
```

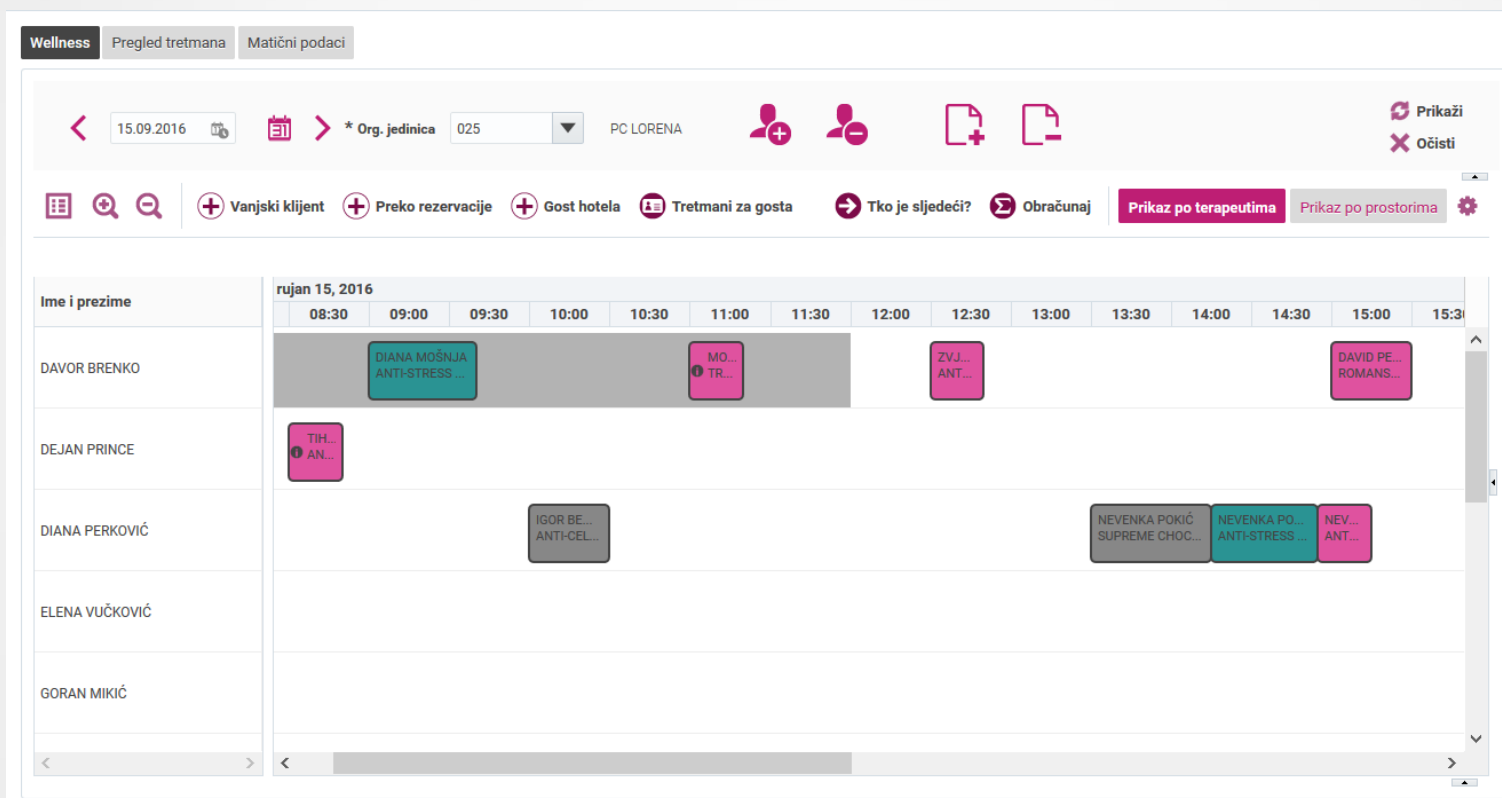
```
public ProizvodiBasic (String _sifra, String _sifProizvodaKasa, String _naziv, String _jmj) {
    this.sifra = _sifra;
    this.sifProizvodaKasa = _sifProizvodaKasa;
    this.naziv = _naziv;
    this.jm = _jmj;
}
```

## \* Oracle ADF Bindings - table

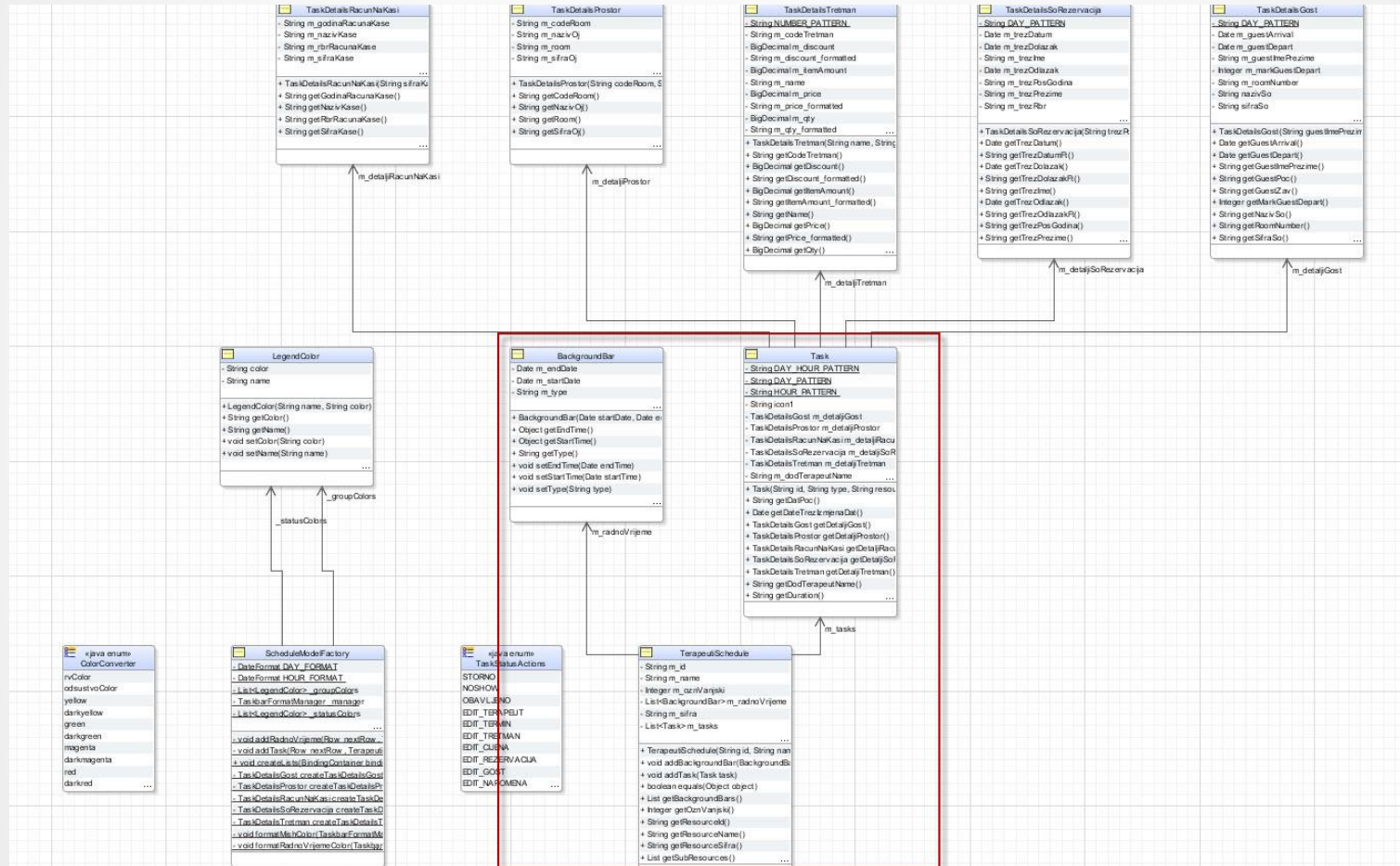


```
public void startButtonClick(ActionEvent actionEvent) {  
    // Add event code here...  
    this.data = new ArrayList<ProizvodiBasic>();  
    DCIteratorBinding iterator = (DCIteratorBinding) getBindings().get(ITERATOR_PRO);  
    ViewObject vo = iterator.getViewObject();  
    if (vo != null) {  
        RowSetIterator rsi = vo.createRowSetIterator(null);  
        if (rsi.hasNext()) {  
            while (rsi.hasNext()) {  
                Row row = rsi.next();  
                String _sifra = (String) row.getAttribute("Sifra");  
                String _sifraProizvodaKasa = (String) row.getAttribute("SifProizvodaKasa");  
                String _jmj = (String) row.getAttribute("Jm");  
                String _naziv = (String) row.getAttribute("Naziv");  
                ProizvodiBasic proizvod = new ProizvodiBasic (_sifra, _sifraProizvodaKasa, _naziv, _jmj);  
                data.add(proizvod);  
            }  
        }  
    }  
    AdfFacesContext.getCurrentInstance().addPartialTarget(getTable());  
}
```

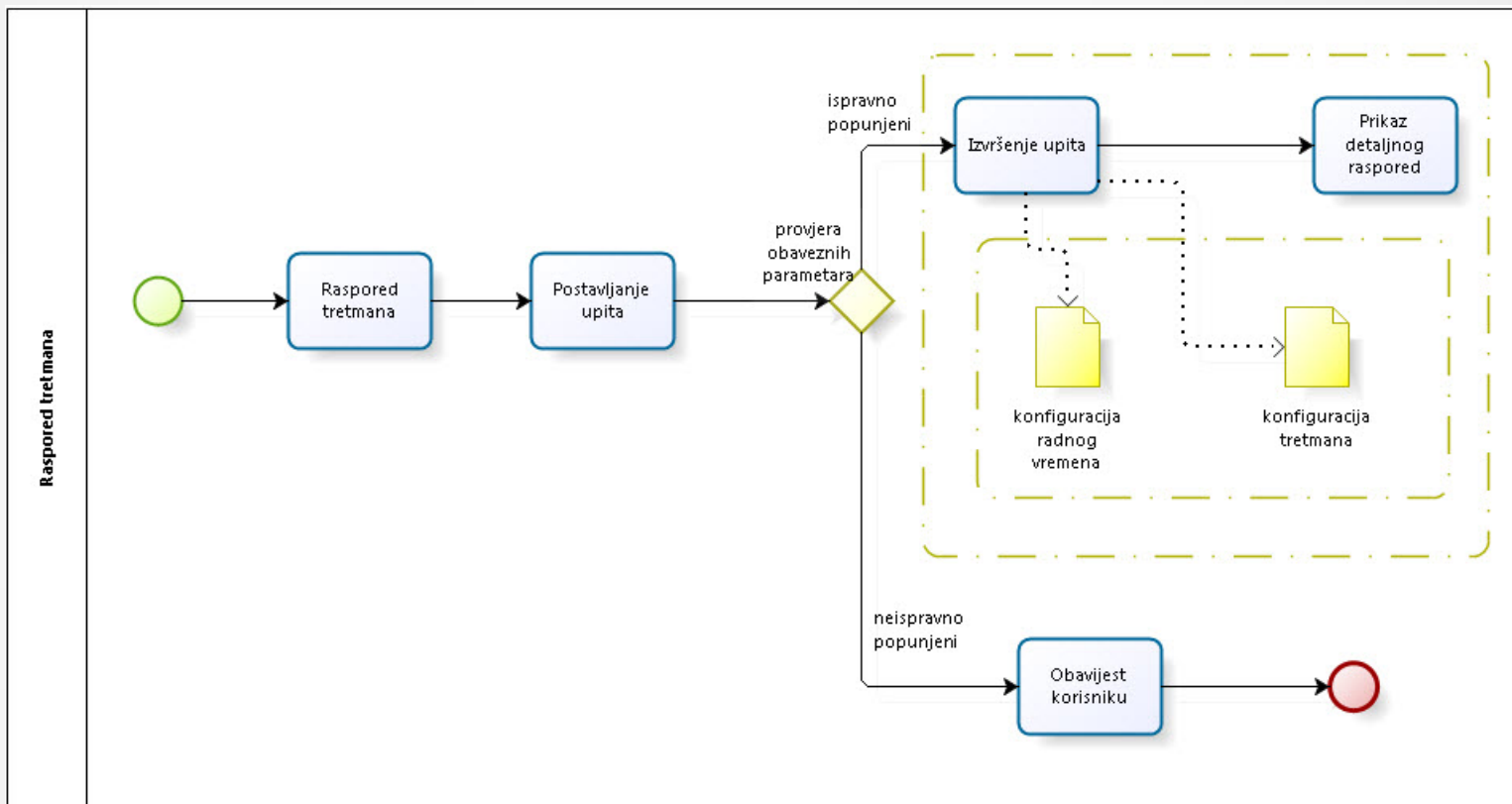
## \* Oracle ADF Bindings – Scheduling gantt



# \* Oracle ADF Bindings – Scheduling gantt

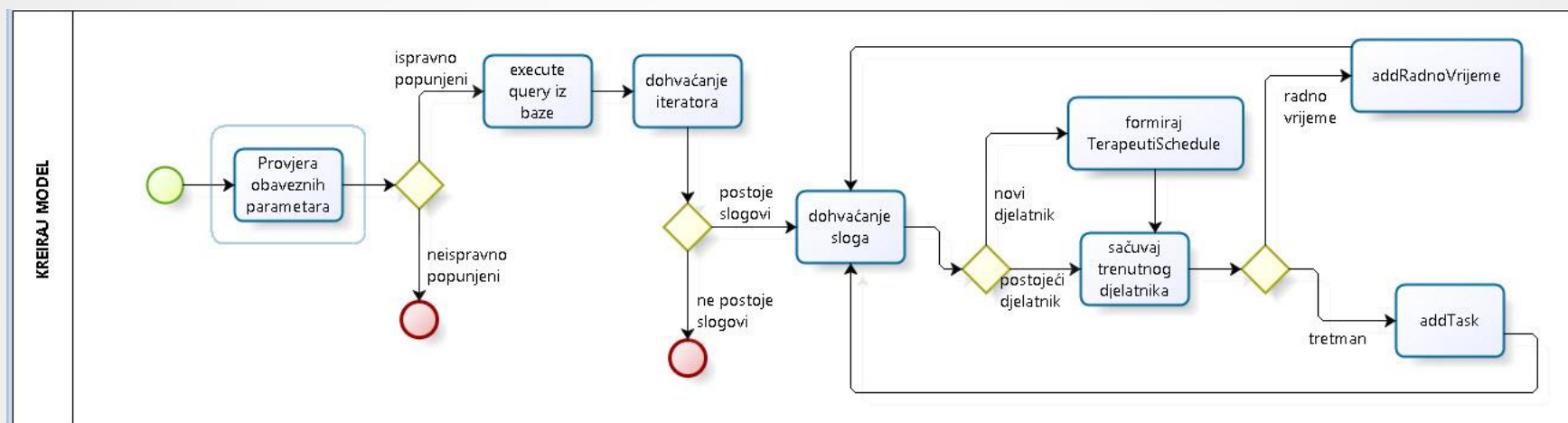


## \* Oracle ADF Bindings – Scheduling gantt





## \* Oracle ADF Bindings – Scheduling gantt



## \* Oracle ADF Bindings – Scheduling gantt

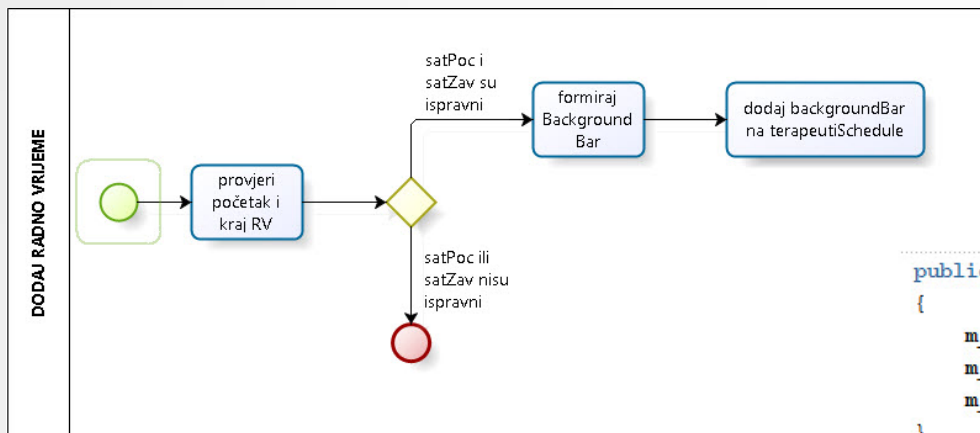
```
public static List<TerapeutiSchedule> radnoVrijemeListModel(BindingContainer bindings, String iterator,
    boolean radnoVrijeme, String tipPregleda) {
    List<TerapeutiSchedule> lista = new ArrayList<TerapeutiSchedule>();
    String oldId = null;
    String colour = null;
    TerapeutiSchedule _r1 = null;
    DCIteratorBinding refreshIter = (DCIteratorBinding) bindings.get(iterator);
    ViewObject vo = refreshIter.getViewObject();
    RowSetIterator rsi = vo.createRowSetIterator(null);
    while (rsi.hasNext()) {
        Row nextRow = rsi.next();
        String id = ((Integer) nextRow.getAttribute("Id")).toString();
        String nazivDjelatnika = (String) nextRow.getAttribute("NazivDjelatnika");
        String sifra = (String) nextRow.getAttribute("Sifra");
        Integer oznVanjski = new Integer(0);
        if (tipPregleda.equals("T")) {
            oznVanjski = (Integer) nextRow.getAttribute("OznVanjski");
        }
        if (oldId == null || !oldId.equals(id)) {
            _r1 = new TerapeutiSchedule(id, nazivDjelatnika, sifra, oznVanjski);
            if (oldId == null || (oldId != null && !oldId.equals(id))) {
                lista.add(_r1);
            }
            oldId = id;
        }
        if (radnoVrijeme && (((String) nextRow.getAttribute("TipPodatka")).equals("RV"))) {
            colour = "active";
            Integer cDostupan = (Integer) nextRow.getAttribute("Dostupan");
            if ((cDostupan != null) && cDostupan.compareTo(new Integer(0)) == 0) {
                colour = "not_active";
            }
            if (tipPregleda.equals("P")) {
                colour = "active";
            }
            addRadnoVrijeme(nextRow, _r1, colour);
        }
        if (((String) nextRow.getAttribute("TipPodatka")).equals("RZ")) {
            addTask(nextRow, _r1, tipPregleda);
        }
    }
}
```

```
public class TerapeutiSchedule {
    private String m_id;
    private String m_name;
    private String m_sifra;
    private Integer m_oznVanjski;

    private List<Task> m_tasks;
    private List<BackgroundBar> m_radnoVrijeme;
```

## \* Oracle ADF Bindings – Scheduling gantt

### Kreiraj radno vrijeme



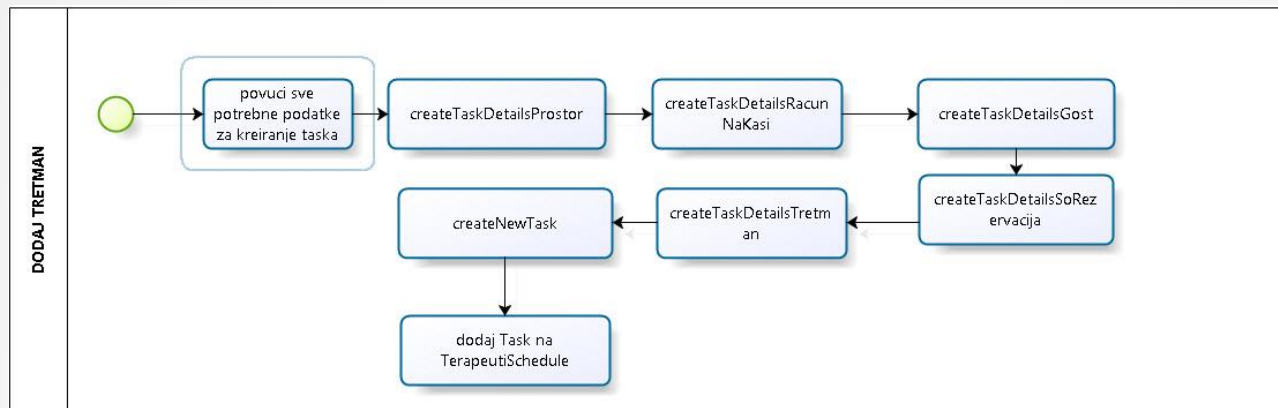
```
public class BackgroundBar {
    private Date m_startDate;
    private Date m_endDate;
    private String m_type;
}
```

```
public BackgroundBar(Date startDate, Date endDate, String type)
{
    m_startDate = startDate;
    m_endDate = endDate;
    m_type = type;
}
```

```
private static void addRadnoVrijeme(Row nextRow, TerapeutiSchedule terapeutiShedule, String pColour) {
    oracle.jbo.domain.Date satPoc = (oracle.jbo.domain.Date) nextRow.getAttribute("SatPoc");
    oracle.jbo.domain.Date satZav = (oracle.jbo.domain.Date) nextRow.getAttribute("SatZav");
    if (satPoc != null && satZav != null) {
        Date poc = satPoc.getValue();
        Date zav = satZav.getValue();
        BackgroundBar _b1 = new BackgroundBar(poc, zav, pColour);
        terapeutiShedule.addBackgroundBar(_b1);
    }
}
```

## \* Oracle ADF Bindings – Scheduling gantt

### Kreiraj tretman

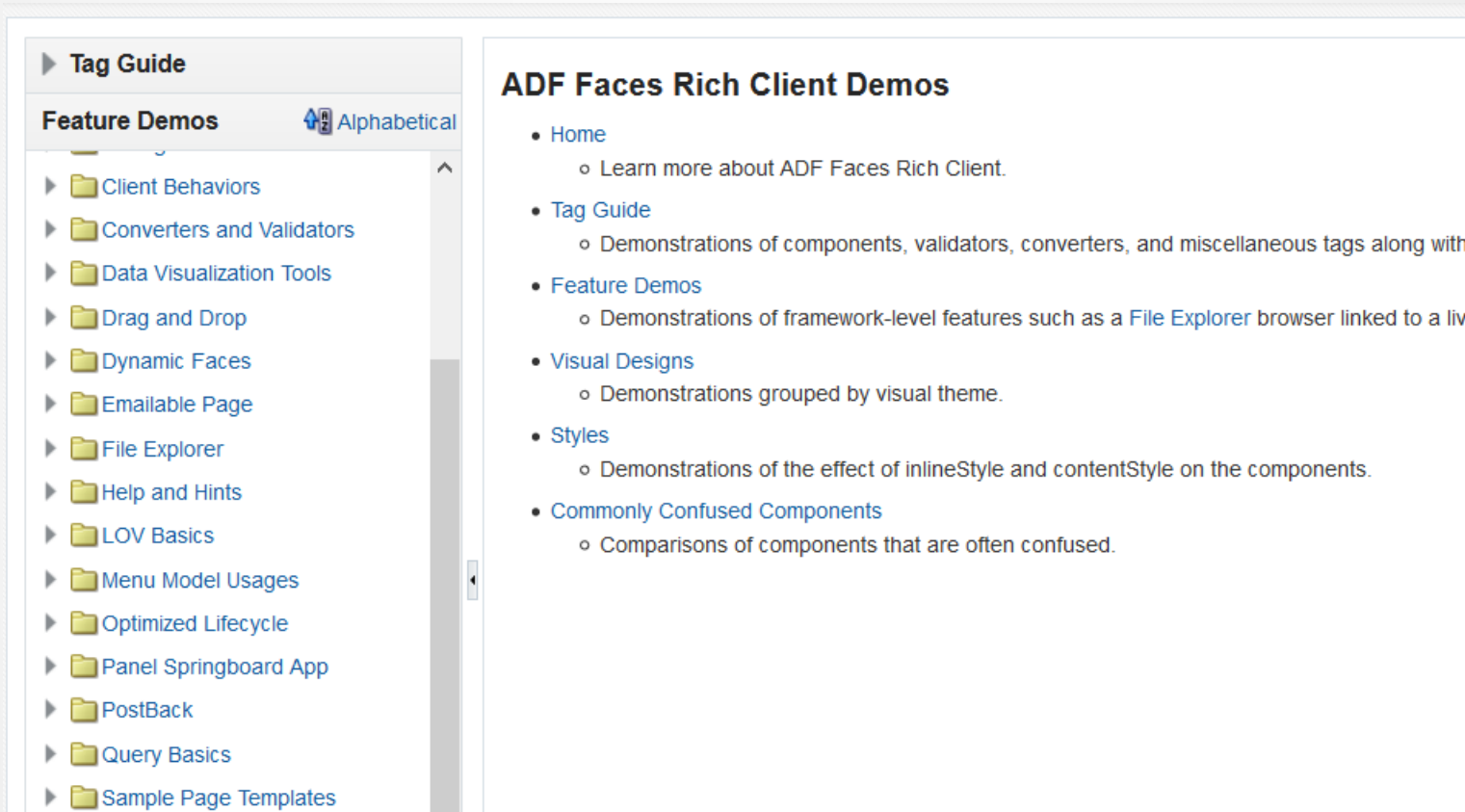


```

private static void addTask(Row nextRow, TerapeutiSchedule terapeutiSchedule, String pTipPregleda) {


    // RESOURCES - za potrebe scheduling gantta
    String idResource = ((Integer) nextRow.getAttribute("Id")).toString();
    // resource je uglavnom terapeut
    String nazivResource = (String) nextRow.getAttribute("Ime") + " " + (String) nextRow.getAttribute("Prezime");
    if (!pTipPregleda.equals("T")) {
        // resource je prostor
        nazivResource = (String) nextRow.getAttribute("Nazivprostora");
    }
    Task _t11 =
        new Task(idRezervacije, status, idResource, nazivResource, poc, zav, imeGosta, napomena, taskSt,
            terapeutName, dodTerapeutName, trezUnioDatum, trezIspravioDatum, unioKorisnik,
            ispravioKorisnik, rbrRezervacijeTretmana, obavljeno, fiksna_rez, detailsProstor,
            detailsRacunNaKasi, detailsGost, detailsSoRezervacija, detailsTretman);
    terapeutiSchedule.addTask(_t11);
}
  
```

## \* Dodatne informacije – ORACLE ADF Rich Client Demos



The screenshot shows a web application interface for the Oracle ADF Rich Client Demos. On the left, there is a navigation pane titled "Tag Guide" with a sub-section "Feature Demos" sorted "Alphabetical". The list includes folders for Client Behaviors, Converters and Validators, Data Visualization Tools, Drag and Drop, Dynamic Faces, E-mailable Page, File Explorer, Help and Hints, LOV Basics, Menu Model Usages, Optimized Lifecycle, Panel Springboard App, PostBack, Query Basics, and Sample Page Templates. The main content area is titled "ADF Faces Rich Client Demos" and contains a bulleted list of links and sub-links.

**Tag Guide**

**Feature Demos**  Alphabetical

- ▶ Client Behaviors
- ▶ Converters and Validators
- ▶ Data Visualization Tools
- ▶ Drag and Drop
- ▶ Dynamic Faces
- ▶ E-mailable Page
- ▶ File Explorer
- ▶ Help and Hints
- ▶ LOV Basics
- ▶ Menu Model Usages
- ▶ Optimized Lifecycle
- ▶ Panel Springboard App
- ▶ PostBack
- ▶ Query Basics
- ▶ Sample Page Templates

### ADF Faces Rich Client Demos

- [Home](#)
  - Learn more about ADF Faces Rich Client.
- [Tag Guide](#)
  - Demonstrations of components, validators, converters, and miscellaneous tags along with ;
- [Feature Demos](#)
  - Demonstrations of framework-level features such as a [File Explorer](#) browser linked to a live
- [Visual Designs](#)
  - Demonstrations grouped by visual theme.
- [Styles](#)
  - Demonstrations of the effect of `inlineStyle` and `contentStyle` on the components.
- [Commonly Confused Components](#)
  - Comparisons of components that are often confused.

## Oracle ADF Bindings

Hvala Vam na sudjelovanju!  
Za sva dodatna pitanja slobodno se  
obratite na e-mail :  
**[diana.mosnja@istrattech.hr](mailto:diana.mosnja@istrattech.hr)**