infoart

19. hroug godišnja konferencija

ADF kao Lego

Vladimir Koren

Rovinj, 16.10.2014.

ADF - ukratko

- Model, View, Controller
- Puno više od MVC-a
- Čvrsta integracija različitih dijelova
- Brz razvoj aplikacija
- Licenciranje

ADF - “problemi”

- Magičan
- Nametnuta struktura aplikacije
- Deklarativan
- Težak za *customizaciju*
- Ogroman

Novi pristup

- Ideja
- Motivacija
- Koje funkcionalnosti su nam potrebne?
- Koje dijelove možemo sami implementirati, a za koje bi nam trebalo previše vremena?

Poželjne karakteristike novog modela

- Jednostavan - bez puno magije
- Single Source of Truth
- Što bliže direktnom SQL-u
- Imperativan
- “Posuđene” ideje od ADF-a

Mogućnosti novog modela

- SELECT, INSERT, UPDATE, DELETE
- Broj redaka bez dodatnog upita
- Sortiranje
- Sume po atributima
- Validacija

Implementacija

- Column
- AttributeValue
- ValueValidator
- Row
- Query - DbQuery, InMemoryQuery
- Model

Primjer sortiranja

```
public void sort(String property, boolean asc) {
    Long currentKey = getCurrentRowKey();
    Collections.sort(getResultSet(), new GenericComparator(property, asc));
    setCurrentRowKey(currentKey);
}
```

```
public void sort(String property, boolean asc) {
    ...
    if (property != null) {
        String newSql = "select * from (";
        newSql = newSql + super.buildSql() + ") order by ";
        newSql = newSql + property + " " + direction;
        execute(newSql);
    }
}
```

Sql2o

- Mali framework za izvršavanje SQL upita
- JDBC ~40 linija koda / Sql2o ~10 linija koda
- Nema generiranja SQLa - nije ORM
- Mapiranje rezultata na POJO objekte
- Imenovani parametri

Sql2o primjer

```
public class Artikl {
    private int id;
    private String naziv;
    private Date datumUnosa;

    // geteri i seteri
}
```

```
select
    id,
    naziv,
    datumunosa
from
    artikli
```

```
Sql2o sql2o = new Sql2o(dataSource);
String sql = dohvatiSqlIzDrugogBaloncica();

try(Connection con = sql2o.open()) {
    List<Artikl> artikli = con.createQuery(sql)
    .executeAndFetch(Artikl.class);
}
```

ADF Faces komponente

- Interesantne komponente:
  - *af:table*
  - *af:treeTable*
  - *af:query*
  - *af:inputText*
  - *ostale input komponente*

af:table

- `value="#{bindings.View.collectionModel}"`
- `CollectionModel extends DataModel`
- `FacesCtrlRangeBinding.FacesModel`
  - Ovo se isplati pogledati :)
- `IaCollectionModel`

IaCollectionModel

- `@override getRowKey, setRowKey`
- `@override getRowIndex, setRowIndex`
- `@override isRowAvailable, getRowData`
- `@override getRowCount`
- `@override getSortCriteria, setSortCriteria`
- `selectRow(SelectionEvent)`
- `getSelectedRowKeys`
- `getTotalRow, getCurrentRow`
- `getValidators, getColumns, getFilterModel, ...`

Primjer upotrebe

```
...
<c:set var="X" value="#{SB.model.nekiView.collectionModel}"/>
<af:table value="#{X}" var="row" id="t1" rowSelection="single"
    selectionListener="#{X.selectRow}"
    selectedRowKeys="#{X.selectedRowKeys}"
    displayRow="selected" filterVisible="true"
    filterModel="#{X.filterModel}"
    queryListener="#{X.filterModel.queryListener}">
...
```

af:inputText

- Edit u JSFu nije jednostavan kao onaj u modelu - kompleksan lifecycle
- `value="#{bindings.Attribute.inputValue}"`
- `converter="javax.faces.Double"`
  - alternativno: `<af:convertNumber/>`
- `IaLongConverter, IaDateConverter, …`
- `<f:validator binding="#{bindings.Attribute.validator}"/>`
- `IaFacesValidator`

Primjer upotrebe

```
...
<c:set var="X" value="#{SB.model.nekiView.collectionModel}"/>
<af:inputText value="#{X.currentRow.atribut}"
    label="#{X.columns.atribut.label}"
    converter="ia.LongConverter">
        <f:validator binding="#{X.validators.atribut}"/>
</af:inputText>
...
```

Povezivanje modela s Facesima

- Bindingsi su još jedan sloj koji hoćemo izbjeći
- Jedan dedicirani managed bean je dovoljan
- Koji scope odabrati?
- SessionBean
  - `model = new Model();`

Nedostaci novog pristupa

- Funkcionalnost frameworka neusporediva
- Vlastita implementacija svega što nedostaje
  - preduvjet je dobro poznavanje tehnologija
- Inicijalno potrebno puno više vremena za razvoj aplikacije

Zaključak

- Isplati li se ovakav hibrid?
- Naučeno puno
- Vremenski zahtjevno uz mogućnost lošeg ishoda
- Dobar osjećaj

Kraj

- Pitanja
- vkoren@infoart.hr